Universitat Autònoma de Barcelona

escola d'enginyeria

ANALYSIS OF AIRPORT TELEMATIC DATA USING DATA MINING AND MACHINE LEARNING

Memòria del Treball Fi de Grau
Gestió Aeronàutica
Realitzat per
Moisés Ortega Collado
i dirigit per
Liana Napalkova
Sabadell, 9 de juliol de 2015

UAB
Universitat Autònoma de Barcelona

escola
d'enginyeria

El sotasignat, *Liana Napalkova*

Professor/a de l'Escola d'Enginyeria de la UAB,

CERTIFICA:

Que el treball a què correspon aquesta memòria ha estat realitzat sota la seva direcció per en/na *Moisés Ortega Collado*

I per tal que consti firma la present.

Signat: ....

Sabadell, ...de...............................de 2015

FULL DE RESUM – TREBALL FI DE GRAU DE L'ESCOLA D'ENGINYERIA

Títol del Treball Fi de Grau: ANALYSIS OF AIRPORT TELEMATIC DATA USING DATA MINING AND MACHINE LEARNING

Autor: Moisés Ortega Collado

Data: Juliol de 2015

Tutora: Liana Napalkova

Titulació: Gestió Aeronàutica

Paraules clau

- Català: telemàtica, Machine Learning, Data science, regressió logística, transformació de dades,
- Castellà: ..
- Anglès: ...

Resum del Treball Fi de Grau

- Català: Els avenços tecnològics en diferents camps de la ciència provoquen que molts d'aquests acabin convergent en una de sola. Això succeeix en el cas de la telemàtica de dades, una combinació sinèrgica de la informàtica i les telecomunicacions.

  L'ús de la telemàtica no és comú dins les tecnologies emprades als aeroports. Tot i això, moltes de les gestions, dades i serveis són duts a terme de manera informatitzada i en molts casos transmesa a través de sistemes de telecomunicacions.

  En aquest treball es pretén estudiar les característiques que conformen i constitueixen la tecnologia telemàtica, així com analitzar els beneficis que se'n deriven del seu ús tot avaluant-los a través d'un cas real.

  Per tal d'assolir l'anàlisi es proposa un model basat en tècniques de Machine Learning i Data Science que permeti extreure la informació i el coneixement de les dades proveïdes per tecnologies de telemàtica i posteriorment analitzar els resultats obtinguts.

  En aquest model es duen a terme diferents mètodes i procediments propis de les tècniques emprades per a la predicció i classificació de les dades. Es per això mateix que el desenvolupament d'aquest model pot esser emprat en d'altres temàtiques i camps d'aplicació.

  No obstant, aquest model està adaptat a extreure coneixement de les característiques de l'entorn que conforma la telemàtica en l'ús d'automòbils, per tant si es volgués emprar en altres contextos, el model s'hauria d' adaptar en aquest sentit (mantenint però la base troncal del model proposat.

  Al llarg del treball es proposen diferents àmbits d'aplicació on, amb el proveïment de serveis i dades telemàtiques, el model predictiu proposat podria aportar valor i suport per a la gestió de les operacions terrestres dels aeroports.

- Castellà: El avance tecnológico en los distintos campos de la ciencia ha causado que muchos de estos acaben convergiendo en una misma. Esto sucede en el caso de la telemática, que es una combinación sinérgica de la informática y las telecomunicaciones.

  Hoy en día la telemática no es muy usada en las tecnologías de los aeropuertos, no obstante muchas de las gestiones, datos y servicios son llevados a cabo de forma informatizada y en muchas ocasiones son trasmitidas a sistemas de telecomunicaciones.

  En este trabajo se pretende estudiar las características que rodean y constituyen la tecnología telemática, así como analizar los beneficios que se derivan de su uso evaluándolos mediante un caso real.

  Para poder realizar el análisis se propone un modelo basado en técnicas de Machine Learning y Data Science que permitan extraer la información y el conocimiento de los datos obtenidos por las tecnologías de telemática.

  En este modelo se incluyen distintos métodos y procedimientos propios de las técnicas usadas para la predicción e interpretación de datos, obteniendo un modelo que podría ser usado en otros campos de aplicación u otras temáticas.

  No obstante, este modelo está adaptado a la extracción de información sobre las características del entorno que forman la telemática en el uso de Automóviles, por lo tanto si se quisiera aplicar el susodicho modelo en otros ámbitos, éste debería ser adaptado a los nuevos contextos (manteniendo por eso su base troncal).

  A lo largo del trabajo se proponen diferentes ámbitos de aplicación del modelo donde, juntamente con los datos y servicios telemáticos, podría aportar ventajas extras en la gestión de las operaciones terrestres en los aeropuertos.

- Anglès: The rapid development of technology in all the scientific fields has caused the convergence of them into the same science's field. An example of this is Telematics; a synergetic combination of informatics and telecommunications.

  Nowadays telematics is not very common in the context of airport technologies. However, many of the paperwork, data and services are computer-based and then retransmitted to telecommunications systems.

  In this thesis the characteristics that surround telematics technology are studied, and the benefits of its use are analyzed through case studies.

  To develop models, Machine Learning and Data Science techniques have been used. These techniques allow the extraction of proper information of the data obtained by telematics.

  The models include different procedures, including prediction and interpretation data techniques. As a result, the application of this model could be useful in different fields or environments.

In this thesis, the models are adapted to the extraction of the information about the characteristics of Telematics in Vehicles; therefore if this model is going to be used in other ambits, it should be adapted to the new context (keeping the main base).

During the thesis different application uses for the model are suggested. These applications, joining data and telematics services, might provide many advantages in the ground operations management at the airport such as the fuel consumption optimization across an accurate use of the concerning features affecting the fuel waste, reducing therefore emissions and avoiding monetary penalties for the accomplishment of certain environment protocols

ACKNOWLEDGEMENT

I would want to render my thanks to all those who helped me altruistically, making this work better and easier. Especially I express my gratitude to my project's tutor, Liana Napalkova. Without her attention and dedication this thesis would not be definitely the same and I would have been lost longer.

Table of Content

Introduction 9

State of the art 9

Objectives..11

Novelty 12

Methodology..13

Structure of the thesis..14

1. Overview of the Airport Telematics Data Analysis Task 15

2. Analysis of Data Science Methods..25

2.1 Data Analysis Methods..25

2.1.1 Data Cleaning 26

2.1.2 Data transformation 27

2.1.3 Center and Scaling..30

2.1.4 Data Smoothing..35

2.1.5 Feature Engineering 37

2.2 Machine Learning Methods..46

2.2.1 Prediction study design 46

2.2.2 The importance of cross validation 48

2.2.3 Prediction with logistic regression..49

2.2.4 Prediction with Random forest..54

2.3 Data Science using RStudio..61

2.3.1 Introduction to RStudio and R programming..61

2.3.1 Overview of the caret package 62

2.4 Machine Learning with Big Data 64

4. Case Studies 74

4.1 Case study 1: Recognition of driving styles for a simple data set 74

4.2 Case study 2: Recognition of driving styles with random Forest method..............................93

4.2 Case study 2: Recognition of driving styles for a realistic data set (Kaggle)............................95

Conclusions 98

List of Figures

Figure 1. Increasing gap between knowledge and data creation 10

***Figure** 2. A screen showing the position of vehicles by Pinnacles's software* 17

***Figure 3.** Different features of the performance for the vehicles given* 18

***Figure 4.** Skewed data and its transformation. On the left is shown a density plot histogram. On the right the normalization min - max is applied.* 32

***Figure 5.** Normal distribution which deviation is roughly null and the probability to fall in both sides of symmetry is the same.* 33

***Figure 6.** Distribution density showing skewness* 34

***Figure 7.** Smoothing different span values.* 36

***Figure 8.** Hypothetical intersections between two paths tracked by GPS position* 37

***Figure 9**. View of centripetal acceleration* 43

***Figure 10.**. Regular polygon proportions* 44

***Figure 11.** Example of data partition when the decision tree branches* 55

***Figure 12**. Entropy graph showing an unbiased coin toss case* 59

***Figure 13** R studio environment* 61

***Figure 13..** MapReduce execution process* 68

***Figure 14**. Random trip plotting of data drivers positions* 78

***Figure 15**. Random trip plotting of data drivers positions (different trip)* 78

***Figure 16.** Random trip plotting of data drivers positions (different trip) with data disturbance.*....78

***Figure 17.** Procedure followed to achieve a successful pre-processing in data* 83

Figure 18. ROC Curve generated from two classifier distributions 87

Figure 19. Distributions drawn overlapped 88

***Figure 20.** ROC Curves showing how the performance of it changes when distributions overlap* 89

***Figure 21.** Comparison between Rock cubes performance* 90

***Figure 22.** Different ROC Curves (trip 3 driver 1)* 91

***Figure 23.** Different ROC Curves (trip 4 driver 10)* 91

***Figure 24.** Different ROC Curves (trip 5 driver 2)* 91

***Figure 25**. Different ROC Curves (trip 6 driver 3)* 92

***Figure 26.** Example 1 of ROC Curves approximating log* 92

***Figure 27.** Example 2 of ROC Curves approximating log* 92

Introduction

State of the art

Taking a brief glimpse to our current environment, no one would attempt to find an argument in order to reject the fact that information, in first instance, and data in consequence, are stagger increasing with no end in sight.

Due to the converging between computing and communications, human habits and customs are changing faster than ever and taking a way never seen before. Fact that yield newer opportunities and challenges scenarios to deal with. As a matter of fact, these two main technologies unions, beside other sciences fields are working closely together, have allowed creating a new world with its own rules, limitations and chances to take profit.

The convergence of these two fields, telecommunications and informatics, allows increasing the amount of data gathering and capturing. This is known as telematics.

However, as this new data world increases, the need to use this information and the relationship between the physical world, the one where humans hang out, and the world provided by data are intensely rugged and drawn. Thus, the more data is obtained, the higher is the need of processing these data in order to extract information and make a profit of it.

The last instance, the trouble of massive data storage, is solved by the use of inexpensive multigigabyte disks and cloud computing which make easy to put off decisions and keep them idle waiting for being used sometime forward.

On the other hand, the process and enhancement of information is the question that leaks and needs a solution laid out. The trail across our decisions made through Internet or World Wide Web are recorded in databases; our preferences, our choices, our financial habits, our locations and definitely a wide range of possibilities feasible to record. A simple transaction as could be a flight booking involves a worthy amount of inputs and outputs going back and forth among different bunch of stakeholders and agents turning into, not only the material and touchable service ordered, in this case the people transportation to those who requested by whosoever of the regular air transport businesses, but also the knowledge improvement about customers as could be their preferences or profiles.

Unfortunately, as the data volume stored increases, the rate of knowledge of itself, or the proportion that it is understood, decreases. Consequently, a rudderless gap is growing between the data generation and the understanding of it.

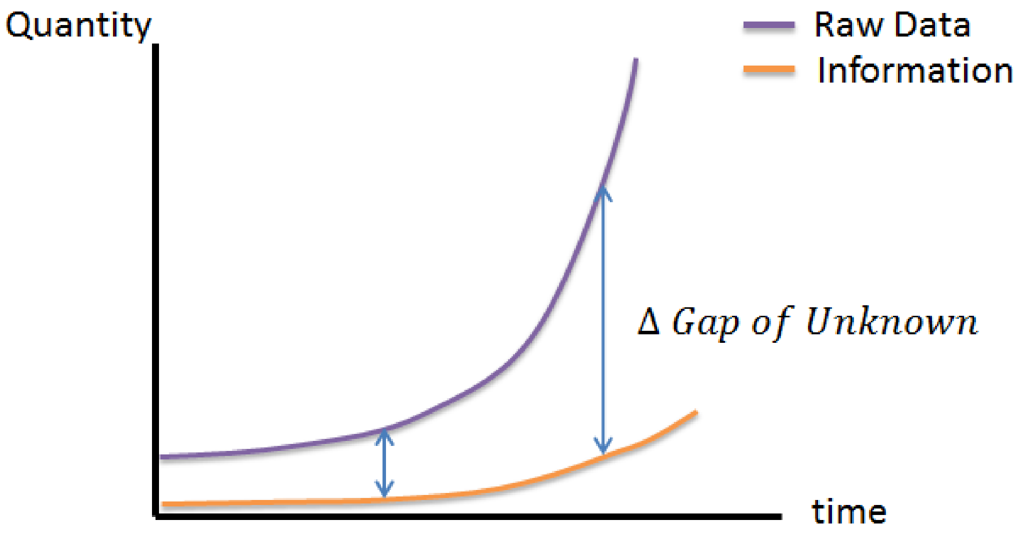


Figure 1. *Increasing gap between knowledge and data creation*

Otherwise, in that gap lie patterns in data which properly and intelligently analysed might be a source of competitive advantages and new useful information. Seeking and looking for those patterns and solving problems by analysing data are the Data Mining aims.

Data Mining is defined as the process of discovering patterns in data *(Stephen Marsland : Machine Learning: an algorithm approach)* . The process must be automatic or semiautomatic. The patterns discovered must lead to a meaningful leverage advantage, allowing making nontrivial predictions on new data for, in the future, help making better decisions. [39]

Classification and prediction amongst other fields where Data Mining is able to perform does not emerge from nowhere. It needs techniques to develop models in order to get valuable outcomes or solutions. Techniques for creating analytical models are studied and developed within a field known as machine learning.

Machine learning is the science that studies algorithms and structural methods in order to construct models that can *learn* from data and use it to make better predictions and decisions. Often due to the closeness machine learning is overlapped with statistical computing that also is developed for prediction-making. Even though, both disciplines must not be divided since both are used for data analysis, a lot of methods used in Machine Learning have been developed within statistics, others also have been adapted to improve performance and be more computationally efficient.

As described in the above paragraphs, nowadays telematics provides a huge amount of data that need the application of powerful analytics techniques to turn these data into winning results. Telematics providers chase to provide software as dashboard, graphics, monitoring, etc. That really might add value to the decision-making and business management by using data science methods to hold software. Thus, machine learning can exhaust all possible and credible scenarios and come up with the best answer analysing factors such as time of day, hard, braking, geographic location, driver age, vehicle profile, gender, claims and payment history, policy tenure, etc.

Objectives

The main objective of the project is to analyse how airport operations might be improved by the deployment of data science techniques applied to the analysis of on-ground telematics data.
To achieve this objective, the following tasks have been formulated:

1. To analyse the state-of-the-art of airport operations with the aim to identify weak points at which the intelligent usage of on-ground telematics data might improve operational efficiency and compliance.
2. To analyse the applicability of data science techniques needed to automate the extraction of useful information from telematics data with the final goal to solve various managerial tasks.
3. Based on real-world telematics data, to develop prediction models in which machine learning and data mining techniques are applied in order to identify driving habits of on-ground vehicle drivers and, thus, to identify gaps for improving the company reputation, reducing costs and exhaust emissions.

Novelty

The novelty of the thesis is the following:

- The analytical study of practical applicability of telematics analytics has been performed with the final goal to identify gaps for the improvement of operational efficiency and compliance in airports.
- The data science procedure that consists of data pre-processing, data transformation, feature selection and prediction of driving styles have been developed and tested in RStudio using real telematics data.
- The developed case studies had demonstrated the effectiveness of the data science procedure that enables the prediction of driving styles based on extracting the acceleration and deceleration profiles for each trip, the number of stops, as well as other useful indicators that impact on the company reputation, fuel consumption and exhaust emissions.

Methodology

The main tool used in this project to process data is RStudio Statistics. RStudio is an integrated development environment for R. It includes a console, syntax-highlighting editor, as well as tools for plotting, history, and debugging and workspace management. Summarizing, RStudio is a comfortable and improved interface to manage in a better way the R language; which might be considered as an aid for code's development.

Due to the closeness with Data Science and Statistics, is necessary a tool able to handle big amounts of data and furthermore integrate statistics tools for data manipulation. R provides widely this requirement and for that reason it fits in the undertaking purpose. On the one hand, R has a language integrated and well-focused to data management and wide range of functions in statistics. On the other hand, not just is able to operate above the data, it includes many data charts, graphs and pictures to illustrate and better understand the data dealing with.

At the same time, R includes the default setting and tools but also a large sorted packages developed by the users'. These packages contribute to enhance the existing tools and add new possibilities.[1]

Once this point is achieved and knowing the program used and some overall characteristics of itself, someone logically could wonder if this performance requirements and unavoidable features mentioned are not committed with other programs available. The answer cannot be other than affirmative; there are some other computing environments and programming languages that offer similar throughput and characteristics as R does. Matlab and Python could have been other choices that largely fit with objectives resolution.

Nevertheless, R was chosen among the other options because is a handle computer language and exist a widely open sources in Internet, as blogs, information forums and some other books where is easy to find answers in case of getting stuck along the code's development.

[1] More information about caret and general features and environment of R is stated in the section 3.3 Data Science using R.

Structure of the thesis

The thesis consists of introduction, 4 chapters, conclusions, bibliography and appendixes. The thesis contains 101 pages, 28 figures and 2 tables. The bibliography contains 45 entries. The thesis is structured as follows:

Introduction motivates the work formulates the aim and tasks, describes the methods used in the thesis, and explains novelty.

Chapter 1 "Overview of the Airport Telematics Data Analysis Task" discusses the characteristics of this technology and analyse its applicability in airports and transports. As well as proposed fields and circumstances where could be applied.

Chapter 2 "Analysis of Data Science" discussions about all the procedures related with Machine Learning methods applied on practice in the theory basis for the methods in which the model constructed were held on. In this section are include getting data and data pre-processing, predictive model construction either in logistic as in random forest approach.

Chapter 3 "Study cases" Composed by three different case:

In the first one, a detailed declaration of all functions and tools that were developed is stated. The algorithm includes the ROC Curve to assess its effectiveness and is applied only in 4 random drivers.

In the second one, a brief discussion of the code is texted; the algorithm is applied into a real case for big data in the telematics “footprints” detection.

Conclusions of the thesis where the assessment and analysis of the achieved results and the conclusion of it in accordance to this analysis is carried out.
Bibliography information sources and its quotation are listed consecutive with its respective number in the text.
Annex: The code source to run the algorithm implementation and probe the model construction.

1. Overview of the Airport Telematics Data Analysis Task

Before starting to tackle deeply the current telematics' state of play in airports and in general in the air transport industry, a definition of the telematics' term has to be figured out.

According to some definitions found through different websites, the concept of telematics refers to any device which merges or blends together telecommunications and informatics. However, Margaret Rouse in Techtarget website[2] analyses in depth the term and adds to the previous definition "telematics is the blending of computers and wireless telecommunications technologies, ostensibly with the goal of efficiently conveying information over vast networks to improve a host of business functions". Many others definitions can be located, but essentially the main idea is the applicability of the union of telecommunications and informatics.[1][34]

This section explains different applications of telematics and the reason why it can be applied in the field of transport and logistics, specifically including airports and ground handling services.

For example, telematics can be used to assess the driving style for different drivers. Driving style impacts directly either in fuel consumption as in safety. In the first case, with fuel consumption business might wonder how to pull down this cost by assessing an optimum driving style way, or a desirable one, and compare it with the present in the drivers and then, try to change it. This fact would have a direct impact into the economy of the business.

Furthermore, since fuel consumption affects the fuel emissions those can be reduced and improve the company image with environment and sustainability philosophies.

In the safety concept, by using telematics can be detected those patterns that hazard the operations and eventually are the cause of potential accidents.

Telematics, among other applications that will be explained later, is largely set in fleet tracking management. Tracking systems make use of location technologies and time information, which are further developed with Global Navigation Satellite System. But rather than just provide the

[2] *TechTarget, SearchNetworking.*

position and the expected direction, those integrate information for a better knowledge of the vehicles deployment and pinpoint opportunities to improve the management of them and reduce significant cost expenses.

This type of information includes the measure of different components and equipment of the vehicle such: amount of fuel, the temperature and the throughput of the engine, flap settings, until vortex generation and some others features that would help to understand and improve the decision-making. All this information transmitted is gathered and stored in the corresponding server, and using the software is displayed creating reports and other tools to improve the knowledge of the information provided.[35]

However, even being possible to measure a lot of information from the aircraft with different devices and sensors, tracking implies two types of operation:

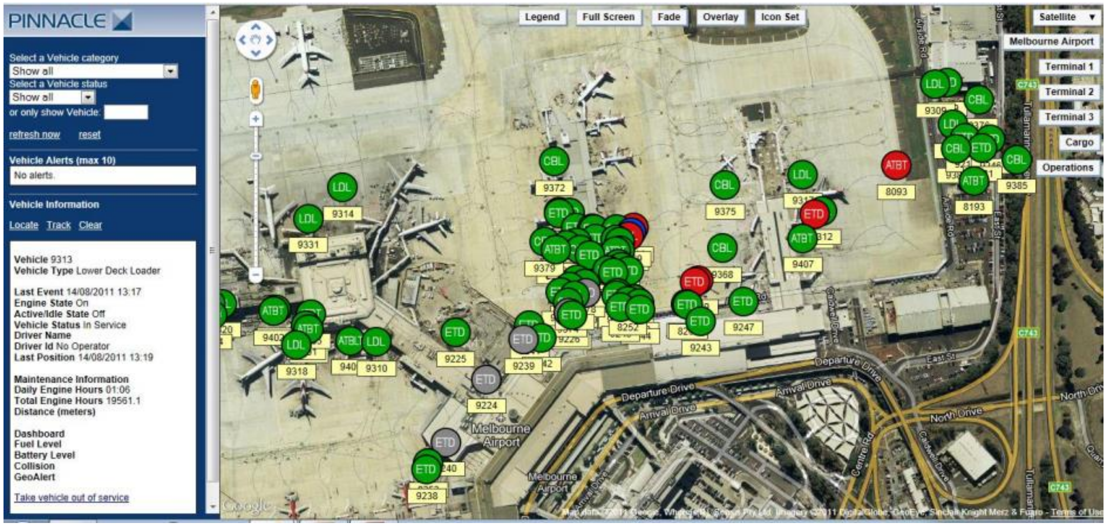
- On-real-time tracking, which yields a seamless distribution of the information and a big effort to hold the network, although the refreshing time is limited.
- Passive tracking, where the information is stored and later (when the operation is finished and data can be downloaded) it is uploaded in the system. Then it is processed or displayed, as the user prefers.[2]

All data captured from the devices and the technologies applied are sent via wireless network, or by uploading the data manually in the case of passive tracking, to a server or software where it is manage and stored. These servers act as a regulator sending the respective information according to the user profile and the information requested by users that demand it.[3]

In plain words, telematics provides a way to get different features of the vehicles or assets with which the company deals and operates, and based on analyzing these features to extract the valuable knowledge for making profit of cutting down the inefficiencies of the system and to conversely make the better use of it.

In figures 2 and 3 an example of the software developed by Pinnacle is shown. It supports tracking on-time position of all vehicles transmitting the features gauged by hardware sensors. These characteristics are displayed in a single screen to scope the overview and manage the system as better as it is necessary (see Figure 3).[4]

In order to clarify and illustrate the concept explained before, let's think about the daily environment of an airport. Airports are big areas of roughly congestions and continuous movement of different kind of vehicles with different purposes and characteristics. Not only the aircraft are in the front of line of the operations, those are supplied and assisted by several vehicles such as the puller or tug, baggage transporters, ramp-handling services, fuel suppliers, Ground Power Units, etc. Those vehicles do not work in an isolated way, rather they closely cooperate with harsh precision and with a continuous race of time.



Select a Vehicle category: Show all
Select a Vehicle status: Show all
or only show Vehicle:

Vehicle Alerts (max 10)
No alerts.

Vehicle Information

Vehicle 9313
Vehicle Type Lower Deck Loader

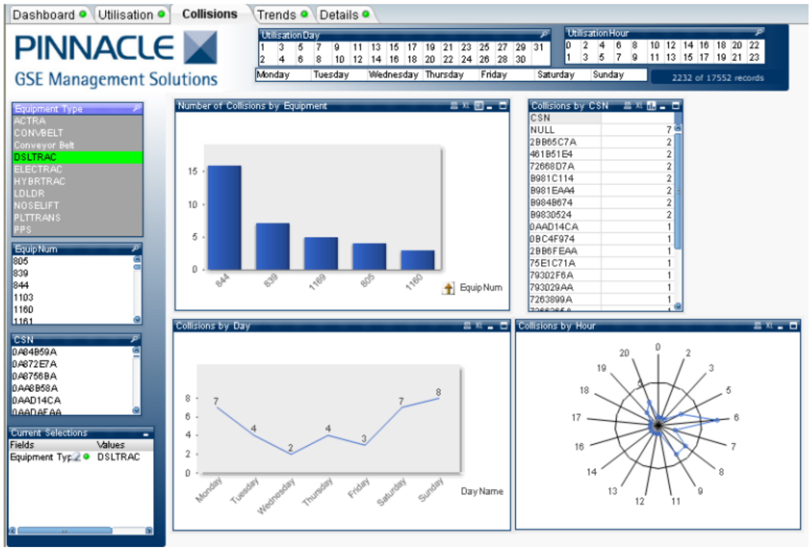
Last Event 14/08/2011 13:17
Engine State On
Active/Idle State Off
Vehicle Status In Service
Driver Name
Driver Id No Operator
Last Position 14/08/2011 13:19

Maintenance Information
Daily Engine Hours 01:06
Total Engine Hours 19561.1
Distance (meters)

Dashboard
Fuel Level
Battery Level
Collision
GeoAlert

Take vehicle out of service

***Figure** 2. A screen showing the position of vehicles by Pinnacles's software*



Collisions by CSN

CSN	
NULL	7
2BB65C7A	2
461B51E4	2
72668D7A	2
B981C114	2
B981EAA4	2
B984B674	2
B9830524	2
0AAD14CA	1
0BC4F974	1
2BB6FEAA	1
75E1C71A	1
79302F6A	1
793029AA	1
7263899A	1

Current Selections

Fields	Values
Equipment Typ●	DSLTRAC

__Figure 3.__ Different features of the performance for the vehicles given

The logistic challenge and the endless precision requested by the users turn out a real headache for operators, coordinator agents and managers. It will be unfeasible trying to coordinate and manage efficiently all these variables (including the restrictions and considerations that the aeronautical field has) without proper information on hand. For that reason, several companies are taking measures to leverage the current situation and enhance their production reducing the costs and improving the service offered to customers.

For instance, in 2010 Swissport started using telematics with different aims and purposes. There are several reasons why Swissport decided to change the way things were usually done in the company and trusted on telematics to transform them in something more sophisticated but eventually better. Swissport realized that the way ground operations were performed was a real waste of time and inefficiently in many aspects, i.e. the vehicles and operators were oversized and consequently most of the time idle and unfruitful.

An obstacle which Swissport came up with tugs vehicles management was that, as airfields and airports are huge extensive areas with several terminals plus the respective gates for each terminal, the time travel from point-to-point of the airport was so elevated. Additionally, tugs are not especially fast due to the speed limitations throughout the platform to 15 km/h in some points. Furthermore, in some areas there are restrictions because of the clearance and priority crossing for other vehicles or aircrafts. This turns into awaiting times in remote places of the airfield without doing anything else that remain idle.[5]

Taking into account this fact, handling service suppliers locate its vehicles before the flight arrival just in case the flight would have come earlier or perhaps on-time. However, the tug would not be able to reach its operation position due to the long distance to the end point or due to any of the reason exposed before.

Nevertheless, many times flights are delayed, and that modifies the original location of the tug. In other words, the tug is not in the right position and remains idle waiting for a flight that is supposed to arrive but do not. This delay implies an opportunity cost due to this tug could be used in other operations and finally less tug vehicles would be needed for carrying out the same service. Anyway, the worst point is that these operators in many cases are not able to know the current state of the flights and they have to be informed by others instead of being independent and autonomous.

One way to solve these disadvantages would be the use of telematics. Telematics can connect both airline servers with handling operators in both directions; either company with the operators as operators with the company.

Using telematics the handling services suppliers could visualize in real time where their operators are, the state of them, and on which tasks are they working on. For example, for the tugs or pullers, it would be useful to set a mechanism that permit to visualize and display the state of the task they are working on and the current state of them including position, fuel tank, and operability of the employee[3]. Concurrently, the workers could be provided with tools to visualize the state of those activities and circumstances that affect them directly as flight schedule on-time, working position assigned, etc.

In this way, the company could try to find out the better disposition of its vehicles and be more flexible against unpredictable circumstances and others unforeseen events. Therefore, instead of selecting the orders for every vehicle before the operations and keep those unmovable, these

[3] There are different shift of work and timetables.

could be assigned on-demand to those vehicles in better disposition of carrying out the task in the better circumstances. For instance, by closeness with the target or the idleness, thus saving time trips and movement operation for vehicle and improving the time performance of the assets.

Some telematics' companies realizing the potential of this technology and the possibilities it offers for the airport ground services, they have bring out further this term and they have been trying to encompass all the services and integrate those into the business. For example, the leasers of GPUs provide the service and send the bill automatically to their customers according to true usage and energy supplied. Thus, billing airlines is made through a software interface that transmits the usage data to the billing system, ensuring a higher efficiency when it comes to billing.

Summarizing, the background idea is to leave away the current ruthless schedules and inflexible activities planning that are susceptible of becoming interrupted and jumbled, hence allowing poor leeway to solve the troubles that came upon. Conversely, the idea is to provide tools for decision-making that permit to adapt the resources available in the best way to the present circumstances, turning out in a situation more flexible and manageable besides gaining efficiency and swiftness in the airport processes enabling stakeholders to coordinate their activities on ground much more better.

The challenge of improving turnarounds

Due to aircraft are intended to transport point-to-point passengers and cargo, plus, their fuel tank runs out promptly and needs to be full again, aircrafts require a series of operations to prepare the same aircraft for the next jump. The operations referring the preparation of inbound the aircraft and outbound the flight that is scheduled for the next jump with the same aircraft it is called turnaround. Respectively, turning around activities include inbound and outbound of passengers, crew, catering services, cargo and baggage services.

During turnaround, beyond the purpose of replacing and changing those things that shouldn't be in the aircraft such as; passengers, debris, cargo, etc. for those that must be in, there are technical activities necessary to perform the air transport activity safely and feasible from the operational point of view; e.g. fuelling, checking routine, cabin and aisle cleaning…

Turnaround process is a crucial step in the milestone of the airlines' schedule: At the end, air transport users principally sell *time* to their customers as aircrafts might be seen as a vehicle that permits going from a point-to-point. This time that customers *buy* when they choose traveling by air transport instead of other modes of transports, is paid by aircraft owners (airlines) during turnaround because instead of moving people and cargo, their production assets are stopped inevitably. When aircraft are in turning around procedures, the business has to pay for all the services it develops in this stage; slot in the platform, the gate of the terminal and the consequent services (jet bridge), fuel reload,… among other taxes and fees for airport services instead of developing its function that is transport things and people.

Some of these costs are unavoidable; however the consequence of the waste of time is not. For that reason, the process is set out the rules for pushing down the execution time. However, as the circumstances changes, and the number of passengers, cargo/baggage loads changes flight to flight turnaround is not equal every time rather it is stochastic.

Under the complexity of the mechanism that involves the process (all agents and variables affecting the whole process) the minimum error or disturbance, not only the "controlled" or related directly with the own actions but the unpredictable or outward streaky conditions such passenger check-in delays, sudden fails in the aircraft mechanism, weather conditions, networks congestion among others, may consequently cause delays to departure flights.

It is important to notice and realize the importance of the turnaround time and the effect it causes, because turnaround determines the schedule and the planning fulfillment of the following flights. The following flights are in close dependency of some others. Thus, a disruption in a single flight not only will affect in the current flight, but all the following of the same flight and those at the same might cause to spread this disruption throughout the whole network.

The efficient management of turnaround might not ensure a good business, but the wrong management might lead to closing the business. Because of this, it is desirable to not abandon the well success of these operations to good luck and other random events that can ruin the good work. On the contrary, a good management of turnarounds might improve the better use of assets and factors that are under the domain of the company turning this one more competitive and productive.

Telematics does not ensure unforeseen events to happen, rather it provides tools and information to handle and get along them.

Collaborative decision making

Doubtlessly, telematics opens up a new horizon. However this horizon it stretches out as long as the information domain is limited. No matter how many information one is able to manage if this information does not depend only on oneself but on any others.

Airports are the heart of several types of transport that involves road transport, in some of them railway, and it is a source of continuous flow of operations. Moreover, companies do not work in an isolated way, rather they use synergies and share resources to take profit of their strengths and abolish weaknesses by forming alliances or partner agreements. This allows entrepreneurs to focus on their fields and externalize and take profit of networks with partners.

Nevertheless, that fact causes businesses depend upon the others and loss certain control of the services. For instance, a logistics supply chain business focused on road transport perhaps requires transportation by plane. This one, as it cannot serve this kind of service, but neither wants to loss its customers, hires or contracts the service of other business that provides air transportation. This current situation often occurs in the transport of goods. On the one hand, the business which is in charge of the air freight depends on the delivery time of the road transporter. So it is sensitive to the delays. Therefore, despite planning which is the best combination of goods and doing all merchandise formalities, it has to wait till goods are in its warehouse, check if it is really what was supposed to be, the state of them and so on and once this is done, it can finally start its liabilities.

On the other hand, the road transporter supplier, once the goods are delivered, losses the perception and the state or the order, being exposed to partner affairs. For sure there are agreements, responsibilities and charges for violation of the contract or for loss of goods, but these are more punishment measures than reactive and operationally evidences.

Through telematics, the on-time order tracking is a feasible reality. By telematics partnerships are able to share not just time arrivals, or departures, but information about fuel consumption, state of the goods in case of special circumstances, walk-in freezer, and control of goods can be done by electronic signals upon labels etc. Even finance transactions can be done by automatics systems only supervised by on charge employees, thus saving time to them to be focused on other more priority tasks.

In addition, this might serve to generate a better statistics and other information about overview business and customer satisfaction.

Even though, not all the data need to be shared or exposed to partners, this data can be protected from unauthorized access by encrypting and by users' access limitations. Thus, keeping the information out of the sheer operation domain safe and not sharing risky information or compromising data to possible competitors.

In conclusion, the deployment of telematics' technology upon the airport facilities and environment can back up and improve some activities and operations management through the benefits this provides. In particular, the following benefits have been identified:

- Reduction of aircraft delays and operations time execution.
- Reduction of ground support equipment maintenance costs.
- Monitoring and reporting of the situation.
- Improvement of manpower efficiency.
- Support of procurement decisions.
- Permission of automatic and accurate billing.
- Environmental reporting.
- Upgrade of the partnership closeness.

However, telematics is not almighty; it requires a cost of maintenance and a substantial investment and absolutely do not ensure a full success on the campaign. Therefore, those businesses that could contemplate the possibility of adding telematics technologies in their business management should assess the benefits of it and keep in mind how this technology will affect.

Even though, telematics is not a mature technology, not because of the technology features and opportunities with which contribute, but the scarce cases and customers that are nowadays performing and developing its benefits.

Nevertheless, some businesses have already starting the deployment of telematics technologies in their daily operations as a support for a better decision-making.

Analyzing how telematics contributes to those businesses, it has been shown that in practice telematics truly adds value and improves decision-making, which later can be extrapolated to other businesses.

AENA, on its side, some years ago decided that a change in the optimization of the resources available in the airport facilities should be implemented in order to deal with the traffic in the airports. Thus, SADAMA[4] system was created and it's currently used in the vast majority of Spanish airports.[6]

SADAMA's goal is to provide in a single and standardized way the airport operational working environment and deploy an assignation tool which would work on real time, getting modifications upon the schedule and reacting properly against them. Moreover, this application has to get several characteristics from the different operations happening in the work environment and relate them with the regarding resource assignment.

These functionalities with the backup and use of telematics technologies could be widely improved and supported. Somewhat telematics could be deployed widely and faster with the addition in the AENA's overall system and be independently performed by the handling services providers through taxes and fees for the technology usage.

[4] SADAMA is linked with other systems as CONOPER in charge of refreshing the information automatically about the state of the flight for all flights with destination or departure at the concerning airport.

2. Analysis of Data Science Methods

2.1 Data Analysis Methods

Preparing the data can be seen as a process necessary to convert data into the same range of values and thus get a uniformly represented data set available to be assessed. Often, data samples are described with several features and thereby are represented with different types of data for every feature. There are considered from numeric values as real-value variables or integer to categorical value. For instance, seizures, temperatures, in the concerning case: positions, speeds, and so on. Examples of the categorical data are the following: sex, citizenship, etc.

Note that numerical values can be compared using logical operates, for example, $5 < 3$, but categorical data can only be compared as blue = blue and blue ≠ red. Because of this, sometimes categorical variables are converted into binary variables with two values: 0 or 1.

Another way of representing and classifying variables is to look at it as a continuous variable or a discrete variable.

The continuous variables (also known as quantitative) are measured using either an interval scale or a ratio scale. These variables are defined or measured with infinite precision. Nonetheless, in the interval scale the zero point is placed arbitrarily and does not indicate the complete absence of what it is actually being measured. Conversely, a ratio scale has an absolute zero point, so the ratio holds true for variables measured using this scale.

Meanwhile, discrete variables are also called qualitative variables. Such variables are simultaneously nominal and ordinal. On the one hand, nominal variables are those that use order-less scales, using different symbols such characters, numbers, etc. Usually they are utilized in order to classify and state different values.

When the modeler is dealing with data, an important thing to keep in mind is the kind of data he/she is handling. In data we can found characters, rates, integers… Often they cannot be blended or mixed up. Similarly, when we refer to mass, time, space etc. with its units and data, it happens the same way. Data must be in the same form to be transformed, changed or handled. For that reason, it is important to split the assortment of data, or otherwise, transform it in a suitable form capable of being treated.

Data is often stored in largely amount of unrelated and messy data sets, heaped in large warehouse provided by different sources. Due to data come from different devices and are provided for several purposes, there must be a differentiation in data features. As long as these different sources may use different representation style, different time of periods, these are restrained for different kind of errors and so others data's own characteristics.

For that reason, before starting the application of data mining methods, it is recommended to scope how available and valuable are data sets with the purpose to find missing values, distortions, misrecordings, inadequate samples and so other features which blur data flawless.

Otherwise, the lack of this process may lead to a looping or endless task because of inconsistency between data, missing data or others circumstances that affect data quality. Conversely, data preprocessing helps to save a lot of time obtaining trustworthy results and make easy the computer processing.

Data preprocessing as works with the set of data which will determine the quality sample in forward phases of the predictive model constructing, is one of the most critical steps to implement by the model programmer.

The affairs and main procedures committed in preprocessing phase are divided depending on their nature. The most important are described below:

2.1.1 Data Cleaning

No matter how much amounts of data available these data are stored in databases, the subset of cases with complete and trustful data are relatively small. Missing data occurs due to no information is provided for several items. Depending on the method applied to satisfactory process data to reach a reasonable conclusion, missing data will be accepted or not.

On the negative case, when the missing values are essential and subject to change the conclusion extracted from the assessment, a simple but fair enough accurate solution is the reduction of data set and eliminate missed values. The way to perform this simple method is cut back the data set in shorter amounts of data to avoid those values missed.

This ordinary method is only feasible when large data sets are available or in case the missing values are lost in a low rate of samples. Reason for what is no longer useful. If samples cannot be shrunk further more by any reason, e.g. there is no possible data assortment within any missing value found it into the sample chosen, in which there is no model construction able to get a reasonable outcome, then, the only choice remaining is to complete the missing values.

First choice available is examine gaps in data and manually add a reasonable, probable or expected value by experience judgment. The lack in this method is produced when any obvious or likely value can be introduced, thereupon the miner would be introducing noise into the data set.

Another usual case it happens when a programmer deals with data sets are the illogical –non-sense values in the data. Those can be detected by applying some out-layers detection methods, by applying some threshold restrictions to be fulfilled by data sample and some other technic to detect deviations and errors.[40]

2.1.2 Data transformation

Box and Cox (1964) published a procedure to transform data into a normal shape. The main goals of this transformation are disclosed below.

Box and Cox method aim's to ensure the usual assumption for Linear Model hold that is: $y \sim N(X\beta, \sigma^2 I)$. Not all data must be transformed to Normal. However, even in those cases that transformation cannot bring the distribution to exactly normal, the usual estimates of λ will lead to a distribution that satisfies certain restrictions, thus will be usually symmetric.

Box and Cox might be considered in two different approaches: the maximum likelihood method, which is easily computable and, on the other hand, the Bayesian method.

For transforming a set of values Y into Y_i^{λ} normalized values (transformed value), can be written as the next formula:[17]

$$Y_i^{\lambda} = \begin{cases} \frac{Y_i^{\lambda}-1}{\lambda} & if\ \lambda \neq 0 \\ log(Y_i) & if\ \lambda = 0 \end{cases} \tag{1}$$

The objective in Box and Cox transformation is to make inference on the transformation parameter λ in order to estimate a value for a given lambda, always in a fixed set of data that maximizes the likelihood function. In other words, the variable Y converted in a resulting Y_i^{λ} will have a distribution as close as it can to normality by finding the optimal value of λ that satisfies the normality assumption best. [8]

Assuming that transformed values responses $y \sim N(X\beta, \sigma^2 I)$, where X is the designed matrix and y is the raw data. The parameters in the equation are the mean β, and the variance σ^2, and parameter λ.

Then, the density for $y\ (\lambda)$ is described as:

$$f\big(y(\lambda)\big) = \frac{exp(-\frac{1}{2\sigma^2}(y(\lambda)-X\beta)^{'}(y(\lambda - X\beta)}{(2\pi\sigma^2)^{\frac{n}{2}}} \tag{2}$$

Finding optimal λ requires an iterative procedure usually performed between two boundaries e.g. [-2,2].[5] The algorithm followed to achieve a successful transformation is done by applying Box-Cox transformation with a fixed λ and later calculate the error function for λ tied. This process is repeated over and over till all the λ values previously selected are equalized. Once a suitable λ is found for which the error function is minimum and hence, the outcome λ approximate the transformation as close as possible.

Once data transformation methods, as scale, centering and Box & Cox transformation are applied data must be further suitable and ready for being used for modeling. At the beginning of this section was said that in data sets there is untrustworthy information regarding some trips which

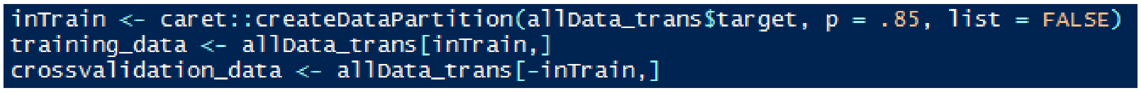
[5] Otherwise the values of lambda might be infinite.

do not belong to the driver to whom is appointed but another a priori unknown driver from the sample.

Predictive Modeling might help to identify those trips that are wrongly set up. To achieve this goal, transformed dataset must be divided into training data and cross-validation sets (or test set). Training data set is used to build and tune[6] the model and test set is used to estimate the model's predictive performance usually by the hand of cross-validation.

Cross-validation allows assessing how well the results of a statistical analysis will generalize to an independent new data set. Furthermore, Cross-Validation also is useful for overcoming the problem of over-fitting. Over-fitting is a term which refers to when the model requires more information than the data can provide. For example, over-fitting can occur when a model which was initially fit with the same data as was used to assess fit. Much like exploratory and confirmatory analysis should not be done on the same sample of data, fitting a model and then assessing how well that model performs on the same data should be avoided. When we speak of assessing how well a model performs or in other words how well the model predict new information. [40]

Caret package has several functions that attempt to streamline the model building and evaluation process, as well as feature selection and other techniques. Amongst these functions specifically *createDataPartition* can be used to create randomized samples of data into training and tests sets.



```
inTrain <- caret::createDataPartition(allData_trans$target, p = .85, list = FALSE)
training_data <- allData_trans[inTrain,]
crossvalidation_data <- allData_trans[-inTrain,]
```

In variable p is specified how this data is split up, in this case 85% and 15%, and list as FALSE avoids returning data as a list. The rate and the amount of data destined to training data or test is under criteria of modeler. Ideally, the model should be evaluated on samples that were not used to build or fine-tune the model, so that they provide an unbiased sense of model effectiveness. As said above, training data is the term to create the sheer model, while the test set or validation is used

[6] Train the model in order to make it a good learner of the inferring and predictive values.

to analyze the performance. The larger is the data set evaluated the more quantity to gauge the performance.[7] [9]

The process of splitting data in different data sets must be done, under all circumstances, once all data have been transformed and normalized under the same parameters. Otherwise transformation would bounded by different values according its data sets constraints. Thus, normalization and data distribution in each case would not be comparable and transformation would have been made useless.

2.1.3 Center and Scaling

Starting by the first step to address the proper process to transform data, before must be said that raw data transformation is not only a process that has to be done to organize data into a standard form ready for being processed by data mining methods and other computer domains.

Furthermore, data preparation may lead to the best-performance of algorithm applied as well as improve the predictive ability of some models. For that reason, a data transformation is required when working with datasets analysis, albeit this pre-processing procedure may vary depending on the samples characteristics and the desired objectives.

Data transformation varies depending the kind of errors data is subject to: missing data, out layers, nonsense values, and so on. But mainly data is transformed in relationship with the model approach chosen. Not all the models respond and perform with the same manner.[11]

Standardization of dataset is a common requirement for many machine learning estimators implemented as data transformation. For instance, RBF Kernel of Support Vector Machines[8] or the I1 and I2 regulators of linear models assume all features are centered around zero and have unique variance; otherwise they might behave badly if the individual features do not look likes standard normally distributed data: zero mean and unit variance [-1,1] or [0,1].

[7] Random sampling, dissimilarity sampling and other ways of splitting data has not been taking into consideration here.

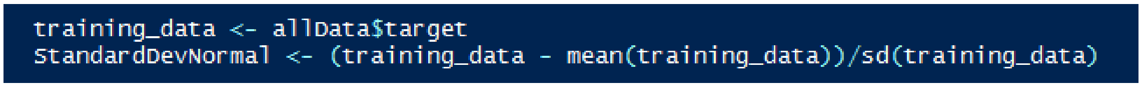
[8] Supervised learning models with associated learning algorithms that analyze data and recognize patterns, used for classification and regression.

When values are highly skewed, is likely to trick up machine learning methods since the variance is larger because of some extreme values. To solve this “outliers” hitches, a solution may be find out through standardize variables. There are several types of transformation of data. A usual form to do this is by removing the mean value and divides by standard deviation. This process, called commonly Standard Deviation Normalization, which takes data centered on 0 with standard deviation of 1. [12]

Standard deviation normalization transforms the data into unrecognizable form from the original data. For a feature x, the mean value and the standard deviation are computed for the entire dataset. Hence, for each (i), the feature value is transformed using the following equation: [40]

$$X(i) = \frac{(\ v(i) - mean(X))}{sd(X)} \tag{3}$$

Then, in RStudio code we will have something as:



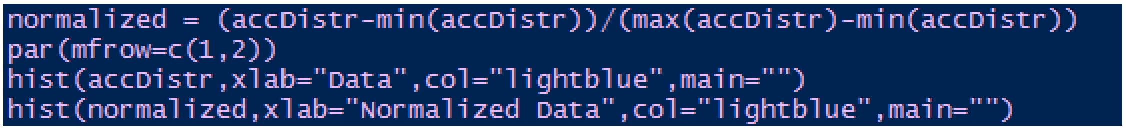
```
training_data <- allData$target
StandardDevNormal <- (training_data - mean(training_data))/sd(training_data)
```

However, caret package as will be probed in paragraphs above saves all this tedious work and uses internal functions of the package.

An alternative standardization is scaling features to lie between a given domain, between a minimum and maximum value, often between zero and one. To obtain distributed values on a whole, normalized interval previously specified, using min – max normalization can be equalized through a formula given as:

$$z(i) = \frac{(\ (x(i) - \min(x(i))\ )}{(\max(x(i)) - \min(x(i))\ )} \tag{4}$$

Below there is a code execution which applied the previous formula:



```
normalized = (accDistr-min(accDistr))/(max(accDistr)-min(accDistr))
par(mfrow=c(1,2))
hist(accDistr,xlab="Data",col="lightblue",main="")
hist(normalized,xlab="Normalized Data",col="lightblue",main="")
```

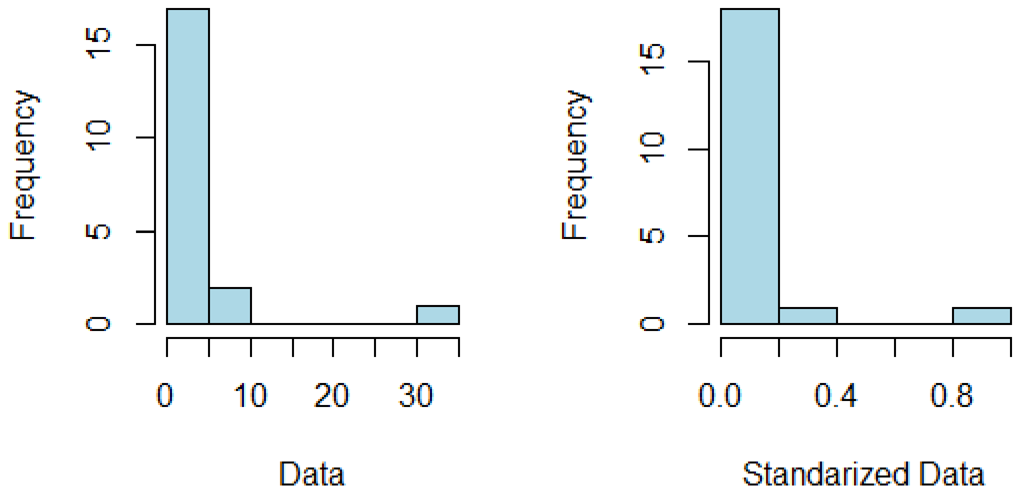


Figure 4. *Skewed data and its transformation. On the left is shown a density plot histogram. On the right the normalization min - max is applied.*

As specified before, often variables of a data set must be transformed in a suitable way before performing predictive modeling. Many statistics analyses techniques only work under the normality condition. Thus, the original variables transformations into a normal distribution allow the statistical techniques application that, otherwise, without normality assumption would not hold enough well.

When a distribution is seldom symmetric, that means that probability of falling on either side of the distribution's mean is not equal, the distribution is skewed. If distribution yields a large number of points on the left side (small values) than on the right side (higher values) we are talking about right-skewed. Conversely if values are larger, the distribution will be left-skewed. [13]

Formula to equalize skewness is given by:

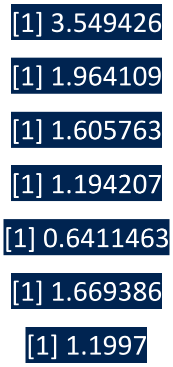
$$skewness = \frac{\sum(x_i - \overline{x})^3}{(n-1)v^{\frac{3}{2}}} \tag{5}$$

Where:

$$v = \frac{\sum(x_i - \bar{x})^2}{(n-1)} \tag{6}$$

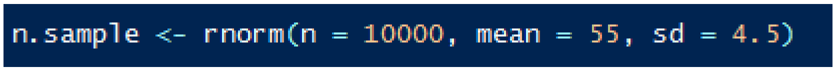
The number of values is n, x is i-nth variable of data set, and $\bar{x}$ the sample mean value.

Here an example of different skewness seizures obtained in the sample regarding acceleration distribution for the first trip is shown.[9]



```
[1] 3.549426
[1] 1.964109
[1] 1.605763
[1] 1.194207
[1] 0.6411463
[1] 1.669386
[1] 1.1997
```

These raw values may not mean too much by themselves. However if we generate random values in a normalized distribution, which is not deviated (the skewness can be negligible) we can notice:



```
n.sample <- rnorm(n = 10000, mean = 55, sd = 4.5)
```

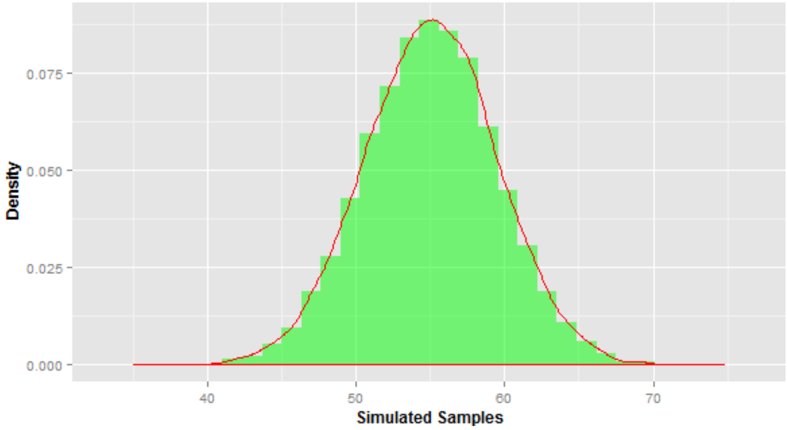


Figure 5. *Normal distribution which deviation is roughly null and the probability to fall in both sides of symmetry is the same.*

Applying function to get skewness in a normal distribution in several randomized distributions generated skewness with the following values of:

[9] The skewness calculation has being made through skewness() function from *moments* package with the purpose of save equalizations and be more straightforward.

```
[1] 0.04026633
[1] 0.01255082
[1] 0.0364343
```

Therefore can be concluded that sample distributions are left-skewed, not just by observing the distribution form but also with quantitative values.[12]

Distribution in data set skewness previously assessed would be viewed as:

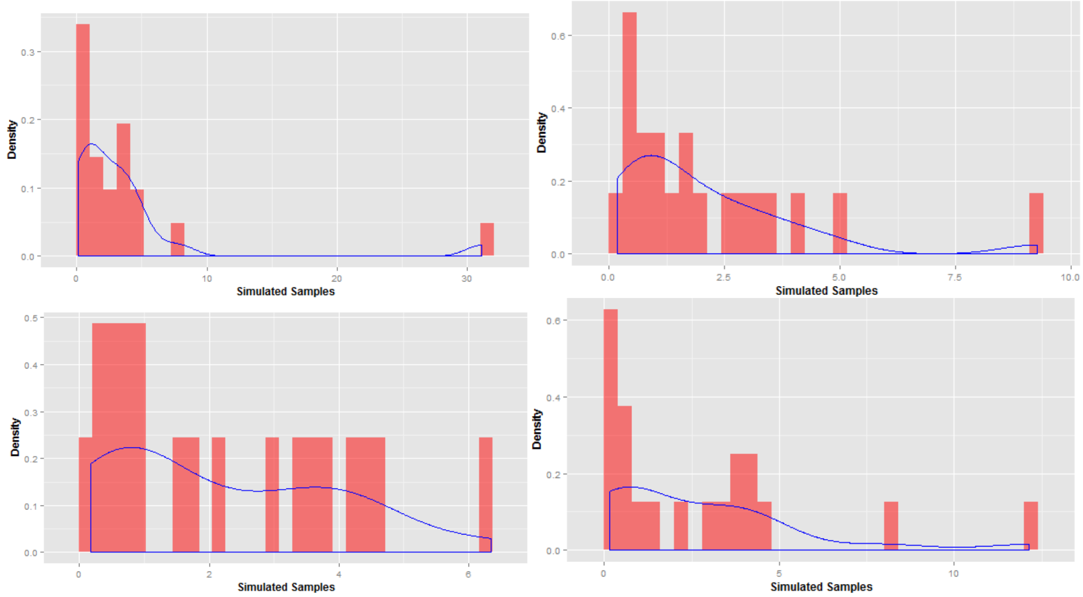


Figure 6. *Distribution density showing skewness*

The graphs, as the skewness rate tested, show a clear tendency to the left. This data bias must be removed, or smoothed in order to get a successfully cross-validation and testing result.

In plain words, data transformation can be seen as a daily life conversion as currency exchange rates, liters into gallons, or mass into energy. Whenever a transformation is performed, the same mathematical operation must hold the whole data sample in order to be sure the relations keep untouched. Despite the common transformations as those mentioned before, the transformation sought it pretends to change the shape of the distribution, fact that does not occur when a simple linear transformation is performed.

2.1.4 Data Smoothing

In many scientific or technological fields where data is provided by devices and technologies, for instance, wave modulations or in this case GPS location, data measuring is performed at discrete points. When data is taken over time may lead to a random variation of it and prone to error. This yields a mistaken use of data if the proper methods to cancel the random variation are not applied.

Data smoothing can be defined as; "the use of an algorithm to remove noisy data, allowing important patterns to stand out."[10] In other words, what smoothing procedure pretends to do is to erase those data values which are products of disturbance and defects in the data provider equipment[11], but keeping the essence of it.[7]

There are several methods to *smooth* data; such as differentiation, Fourier series, computational filters…,etc. The main concepts of two important techniques are explained below:

Moving average

The simpler technique for filtering signals, or data values, consisting of equidistant point is called the *moving average*.

The algorithm is simple. It consists in convert the raw noisy data $[y_1, y_2, \dots y_n]$ into a new data smoothed data set by applying the average of an odd number of consecutive 2n+1 points of raw data $[y_{k-n}, y_{k-n+1}, \dots y_k, y_{k+1,}, y_{k+n-1,}]$

$$(y_k) = \sum_{i=-n}^{i=n} \frac{y_{k+1}}{(2n+1)} \tag{7}$$

The odd number $2n + 1$ is usually named filter width. Intuitively, the grater the width of the filter the more intensified and deep smoothing effect.

The inconvenience of this method occurs when the filter passes through peaks which are narrow in relationship with the filter width. Then, information is last and distorted.

Savitzky-Golay algorithm

An improved and less damaging method for data smoothing if compared to averaging points is to perform a least squares fit of a small consecutive data point to a polynomial and take this new central point calculated from the fitted polynomial curve as the new smoothed data point.

[10] Definition by Investopedia.

[11] Referring to GPS or other any sensor or hardware capable of transmit whatsoever it gauges.

Savitzky and Golay (*Analytical Chemistry, 1964)* probed for the first time that a set of integers $(X_{-n}, X_{-(n-1)}, \ldots, X_{n-1}, X_n)$ can be derived and used as weighting coefficient to do the smoothing operation. The use of weighting coefficients yields exactly equivalent to fitting the data to a polynomial. This results more effective and much faster than averaging moving.

The filtered or smoothed data pointY_k is given by Savitzky and Golay by the equation:

$$Y_k = \frac{\sum_{i=-n}^{n} A_i y_{k+1}}{\sum_{i=-n}^{n} A_i} \tag{8}$$

In the following picture the difference between different span or width filter values can be seen. In this case, instead of using the previously mentioned methods, logistic regression smoothing is used which is much more effective but, at the end, based on the same intuitively idea. The values to plot the graph were 0.05, 0.5, 1, 10. Being the black line the more dispersed, followed by the red on that took 0.5 value and consecutively the green and blue which took 1and 10 respectively.[8]

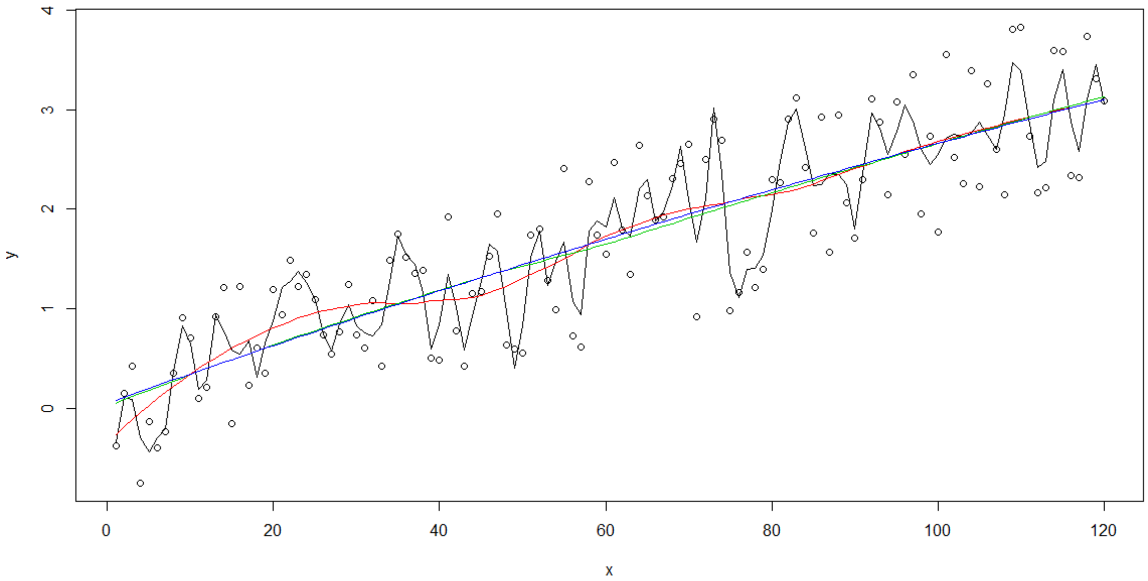


Figure 7. *Smoothing different span values.*

2.1.5 Feature Engineering

Predictive model can't be built up if, beforehand, there is no data to hold it. The first step in the model construction must be setting the parameters, data nature and the features that will constitute the input roots for the model.

Features are all the elements that describe all the model inputs. It might be seen as the creation of the space where the data supplied by external devices, as in this case GPS, moves on and is converted into a meaningful characteristics.

Building up the model implies to know the nature of the data and its limitations and restrictions. In this case, as said before, the data provided is just the positions for a corresponding time in two dimensions. Hence, there are no motivations to attempt equalize how much fuel the drivers burned because there is no relative data to get this kind of information. Or perhaps attempting to describe the slope or inclination of the route followed by drivers would be as well an utopia since data it has only two dimensions and hence are submitted in a plane.

Furthermore, in the information provided can be errors due to equipment's lack of precision, some deviations. This untruthfully data must be take it into account or eliminate it when the feature engineering creation.

In order to illustrate the aforementioned, figure out GPS data supplied: If this one would be entirely precise and with no turning in the signal, the route assessment could be done continuously, somewhat implies a better analysis of the path performed. However, this one is disturbed[12], and there is lack of precision that disrupts data. It would be sought to analyze the intersections of the path because they cannot be equalized by searching two coincident points in the space (because likely this will not ever happen).

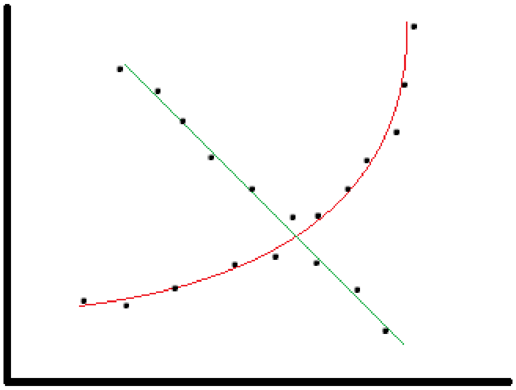


Figure 8. *Hypothetical intersections between two paths tracked by GPS position*

[12] Positions are taken between two different instants. Thereby there exists a time lapse in which information it get lost between the two stages.

As can be observed in the graph above, which illustrates hypothetical intersections between two paths, mathematically if the functions would be, continuous those would share a common point (the intersection point). Albeit, as the locations of the GPS are interrupted and it has some deviations errors due to the lack of precision, the programmer must take into account this considerations and do not assess, in this specifically case, the intersection through the chase of a common point but the intersection of two lines creating vectors with the points.

Once all restrains represented by the problem are taken into account and the assesment for which features would be profitable represent a gain for the model, the feature engineering can be started.

Basic Functions

This section states how to get the basic knowledge of features from raw data supplied by GPS. The more straightforward feature one may come up with the position in relationship with time is nothing else that velocity or *swiftness*. This feature tells how much distance it has been trailed between two points in a defined interval of time.

Once all distances are equalized for each interval, different features about speed can be extracted. For instance, minimum and maximum velocities, mean speed, speed distribution and even skewness of it.

The speed is equalized by the next reason:
We can define instantaneous velocity or just *speed* as the rate of change throughout its trajectory, in a lapse of rate or specific instant, and in a determined trajectory point. Thus, is also well-known as the position vector derivative regarding the time.

$$\vec{v} = \lim_{\Delta t \to 0} \vec{v}_m = \lim_{\Delta t \to 0} \frac{\Delta \vec{r}}{\Delta t} = \frac{d\vec{r}}{dt} \qquad (9)$$

Where:

$\vec{v}_m =$ Mean Speed Vector

$\Delta \vec{r} = Displacement\ Vector$

$\Delta t = Time\ interval\ drawn\ to\ 0.$[13]

The process to reckon the speed in a certain point consists in equalizing it between point A and any other point in the space as closer as possible to A. In that case as GPS do not provide position seamlessly, the time lapse will be the same as the refreshing location time lapse which is the smallest interval possible. Hence, the interval will be 1 second.

[13] Will not be feasible, and is not completely necessary, to drawn it in the concerning case to zero.

The closer is the point chosen to A, the more resemblance between the displacement vectors with trajectory's tangency in such point. Therefore, its module gets closer to the value of the space gone over the track followed.

As long as immediately speed is a vector magnitude, its module can be expressed based on the space covered like:

$$|\vec{v}| = \lim_{\Delta t \to 0} \overline{|v|}_m = \lim_{\Delta t \to 0} \frac{|\Delta \vec{r}|}{\Delta t} = \lim_{\Delta t \to 0} \frac{\Delta s}{\Delta t} \qquad (10)$$

Where the speed vector is expressed through Cartesian coordinates in 2 dimensions by:

$$|\vec{v}| = \sqrt{v_x^2 + v_y^2} \qquad (11)$$

Where v_x is X component of the speed vector previously equalized. Concurrently, v_y is the speed vector$y$ coordinate.

Acceleration

The speed calculation is a generic feature from which further knowledge can be derived. Speed leads to a better understanding of the track followed by the driver. However only with speed the knowledge is limited. Fortunately, from speed values more interesting features can be acquired.

As done in the previous chapter, in order to understand why and how acceleration is equalized a previous brief knowledge about what is acceleration must be fulfilled.

Often when the motion of an object is observed this one does not remain constant throughout the entire track followed. Rather, this suffers an increasing or decreasing in the speed. Depending on the magnitude of this speed rate of change will be denoted as acceleration (if this change in speed is positive) or deceleration (braking) if that is negative.[14]

The function's purpose is to study the instantaneous acceleration, which is nothing else than the speed variation taking place in specific instant of time.

$$\vec{a} = \lim_{\Delta t \to 0} \vec{a}_m = \lim_{\Delta t \to 0} \frac{\Delta \vec{v}}{\Delta t} = \frac{d\vec{v}}{dt} \qquad (12)$$

[14] In physics commonly it is stated that a body experience acceleration also when it changes its speed direction. However, this one is not contemplated in the project's domain.

Acceleration as can be probed is a vector magnitude so can be seen as:

$$\vec{a} = a_x\vec{\iota} + a_y\vec{\jmath} = \left(\lim_{\Delta t\to 0}\frac{\Delta v_x}{\Delta t}\right)\vec{\iota} + \left(\lim_{\Delta t\to 0}\frac{\Delta v_y}{\Delta t}\right)\vec{\jmath} \qquad (13)$$

Again the module in Cartesian coordinates can be expressed as the following:

$$|\vec{a}| = \sqrt{a_x^2 + a_y^2} \qquad (14)$$

Where a_x are the x components of acceleration previously equalized ($\frac{\Delta v_x}{\Delta t}$), and a_y are the y coordinates respectively.

Trip duration

The next feature is coded with the purpose of knowing the length of the trip by assessing the amount of data inputs for each trip. In other words, the time travelled along the path recorded by GPS. To obtain the desirable result, the trip variable is provided into the function and as this variable contains the number of positions collected by the GPS per second, we can know how long it was the trip knowing the length of the vector Trip (for each trip).

As well as in this occasion the variable selected was the time, the total distance trailed would be a possible choice.

Algorithm improvement

Aside the algorithm approach applied, a meaningful point to consider in the algorithm enhancement is the information available to cope. Not just in quantity but in quality.

From the beginning data provided by the GPS, the immediately information extracted is the position of the object in a ruthless sequence of time. At first sight can be seen as not that much information further than what straightly is: the path followed by an object in a specific time lapse. However, along the path done can be subtracted a lot of useful information that could be relevant and helps to clarify more details about what happened along the way and how was done. The more relevant information on hand, the more details and better knowledge of the path assessed. Thereby, even making the problem computationally and operationally more complex, the results expected, in the large majority of cases, will be more accurate and trustful.

Is not difficult to see that if it's compare the main velocities of two different drivers, and the result shows that one is largely higher than the other one, the reasons might be several but nothing assures that suddenly the “slower driver ” in one trip may be faster because is going by high-way or it has rush because an emergency. However, observing the acceleration we can get information about the driving style for each driver. For instance, knowing accelerations and decelerations can be seen how aggressive or moderate a driver is. So, even in some paths expecting to get a high velocity on average, the way the driver is changing speeds, would not be alternated and those won’t be mistaken classified.

More features aside those already calculated in the program that would support to understand the particularities for each driver could be the following:

- *Local Maximum and minimum velocities*

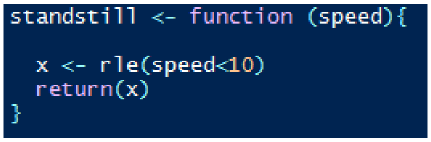
By mathematics is known that when second derivative (acceleration) takes values from negative to positive, or concurrently, from positive to negative, a local maximum or minimum is found[15]. Thereafter, might be equalized, among these two points (the period of time between local minima and local maxima), a measure of linearity connecting the difference between the linear function and the true speed values in the period of time. This feature helps to understand how is the driver behavior, if it’s continuously changing velocities, and the track nature it goes over.

Moreover, this feature can be mixed with speed distribution in those intervals close to Maximum and Minimum speeds values adding some information about the track. E.g. If the maximum velocities are 120 km/h and the speed average in those intervals is around 100 km/h – 120 km/h might be concluded the track is along a highway.

- *Standstill: Time spent in the stops*

Although in those cases when vehicle speed is close to be 0, is rather susceptible to leak some errors because of GPS’s lack of precision.

The process to follow for getting this feature is:



```
standstill <- function (speed){

  x <- rle(speed<10)
  return(x)
}
```

The outcome of this function returns a vector with the values repeat it that accomplishes the restriction imposed. So for instance, if the vehicle stopped 3 times, this function returns a vector of three positions with the number of seconds the vehicle had stopped for each stop.

- *Measure of how many loops, self-intersections… the path had.*

This feature is interesting because it brings out two principles it must be taken into account in the election of the best features engineering.

[15] In those cases where function is continuous.

The first one is the importance or the gain by performing the so-called feature. Sometimes it might happen that even calculating a new feature do not provide a considerable accuracy addition in the model, what yields an unnecessary waste of time processing. Hence, in order to avoid dragging unnecessary loads of operations it might consider the addition of time to carry out the feature in the algorithm and the gain by performing it. For instance, if the addition of a new feature implies 10 extra hours in an algorithm execution when it took 4 hours without the new feature, and the addition turns out to be almost "insignificant", should be reconsidered the addition in the model of this new feature.

However, there are also situations where this a priori "insignificant" addition in the accuracy of the model is imperative necessary or, what's more, this features aids to classify certain drivers and the features turns to be unavoidable. Fact that leads into the second matter to take into account in the engineering feature formulation.

Being aware of the nature of data at hand, and thereby the restrictions of itself, the same feature may be approached in many different ways.

Often the same feature goal can be calculated from different points of view and procedures. One oversized way to perform the intersection study, would be for instance having a list of vectors (direction of the vehicle) and distances between consecutive points (though with the same points is feasible to achieve it) and formulate through parametric equations of each vector and probe if those ones are intersected somehow in any point.

Having the two direction vectors such for instance:

$$\vec{a} = \begin{bmatrix} 2 \\ 1 \end{bmatrix} \tag{15}$$

$$\vec{b} = \begin{bmatrix} 5 \\ -2 \end{bmatrix} \tag{16}$$

Being: $L = \{\vec{a} \vee \vec{b} + \lambda(\vec{b} - \vec{a}\} \mid \lambda \in \mathcal{R}$

$$L = \left\{\begin{bmatrix} 2 \\ 1 \end{bmatrix} + \lambda \begin{bmatrix} 3 \\ -3 \end{bmatrix}\right\} \mid \lambda \in \mathcal{R} \tag{17}$$

The next parametric equations are come out from it:

$$x = 3\,\lambda + 2$$
$$y = -3\lambda + 1$$

Finally only remain to find out if there is any λ value for what the equation is fulfilled.

Although these operations can be restricted only for those points that are closer in distance since in a loop always must be points in the distance are closer than others (taking a point as a reference of the distance). The total amounts of operations are so heavy and sometimes even boundless.

- *Centripetal or Normal accelerations*

Before, when the acceleration was equalized, the direction of the trajectory was entirely neglected: The modulus was the only element considered.

However, every time a particle is drawn into a curve or it moves in a circular movement, this one experiences a force toward the center of the circle and perpendicular to the speed of the particle that is rotating.

This acceleration is known as centripetal, and appears always as the direction of speed vector changes. So, in a curved movement, since the vector of tangential speed is continuously varying, the centripetal acceleration comes out.

The formula to equalize the centripetal or normal acceleration is given by:

$$a_N = \frac{v^2}{R} = \omega^2 \cdot R \tag{18}$$

Where:

v: Tangential speed.

R: Radius of the circumference

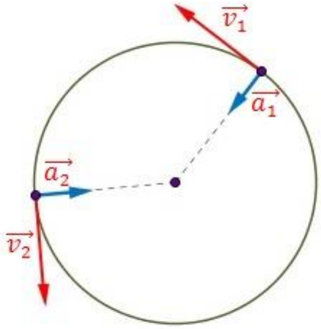


***Figure 9**. View of centripetal acceleration*

Notice in a straight movement (rectilinear), normal acceleration is null because radius is infinite. Notice in the formula it is need the radius of the circumference, but positions in GPS there is no radius to any circumference. So it must be created in order to apply the formula.

The radius, or the distance toward the center of the curvature can be obtained considering some polygonal rules and semblances explained below.

Since, as mentioned before, centripetal acceleration only occurs whenever the body, or the particle, experiences a change in the direction of the trajectory, the first step should be to find those points where the trajectory changes. This can be done through the equalization of the angle between consecutive vectors or just starting a chase for those vectors which change in both coordinates (x,y). In this case it has taken the first choice because in further calculations will be the angle between consecutive vector so can be taken into profit and just compute them once.

The reason for what the angle of consecutive vectors are needed it is because the radius that appear in centripetal acceleration formula's it is calculated through the knowledge of this angle. Regular polygon, which is nothing else that a polygon that keeps all its angles and sides equal, enclosed in a circumference that share all vertex points.

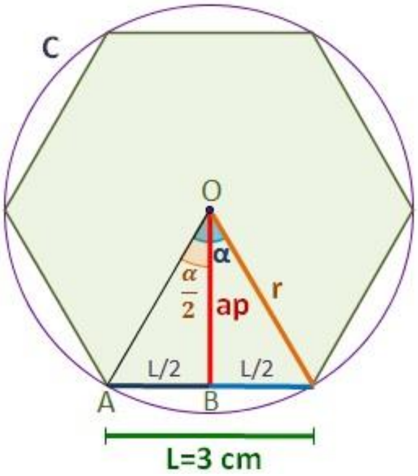


***Figure 10.**. Regular polygon proportions*

In the picture above can be observed that the distance from any point of polygon's vertex toward the center, is the same as the circumference radius.

In regular polygon figures also the angle between all sides of the figure, the outline planes, are kept among them constant. Hence, as long as the figure must be enclosed, the total amount of angle is given by:

$$Sum\ of\ inner\ angles\ = (n-2)\ x\ 180$$

However, as the given information is the angle between consecutive vectors, in this case, the angle formed by each side of the polygon. The formula to know the total amount of angle is given as:

$$Each\ of\ inner\ angles = \frac{(n-2)\ x\ 180}{n}$$

As n, is the number of sides that forms the figure, solve for the n variable gives the number of sides.

Hence, knowing the polygon total sides number and dividing 360º by the so-called number it is gotten the $\propto$ for each side of the polygon.

Thereafter, only remain to apply trigonometry in order to know the radius of the circumference. First, $\propto$ is divided by 2 to form a triangle rectangle. Secondly the distance between consecutive points is equalized. This can be done by applying Euclidean distance:

$$Euclidean\ distance = \sqrt{(x_2 - x_1) + (y_2 - y_1)}$$

Once the distance between consecutive points it is known, the next step is divide this one by 2 to get the opposite leg and apply the formula:

$$\cos \propto = \frac{opposite\ leg}{hypotenuse}$$

Thereby, the circumference's radius is already known and only it left applying the radius into the formula for the discrete centripetal acceleration and once, for the given point, it is finally known the unit of the acceleration:

$$a_N = \frac{v^2}{R} = \omega^2 \cdot R \tag{19}$$

2.2 Machine Learning Methods

All sections above state those procedures concerning the mainstay of the model. However, any new knowledge it has been extracted from it yet. Since the dawn of Machine Learning, and the statistics modeling in general, different methods and approaches to achieve the purpose of construct a successful classification model have been aroused.

In the following sections, the theory basis, in which some models such they performed in the study cases, is detailed.

2.2.1 Prediction study design

Often it happens that one problem in real life has different interpretations of it, it might be tackle from different angles and be resolved by several forms and methods.

In the case of predictive models is not an exception. One problem can lay out in some many ways sometimes achieving similar resources, and others with more or less satisfaction.

The point key here is to know the problem characteristics, limitations and particularities that shape and define the problem to, once they are known, look for the model that better fits the characteristics. For example there are models that need numerous data to really get a successful result, others are outbound to perform, others waste more time, and each model, as a main rule, has it owns advantages but as well disadvantages.

Moreover, even using the same techniques the resolution and the approach can be utterly different. E.g. One might decide to use a random forest technics, or Bayesian networks which both are highly used in Machine Learning but however, construct different trees, or built up different networks. Actually, the desirable performance would be reached testing the model with different approaches and later compare the results and the outcome gotten and later choose the best ones. Before starting to dick deep in the matter, it might be said that in Machine learning and in data science in general, there exist two points of view about how to model the algorithm to extract the information, either in classifications problems or recognition of patterns and some others. These two main approaches are named as supervised and non-supervised learning.

Machine Learning is an algorithm type which is data-driven, i.e unlike the usual algorithms that transform data and extract results from the operands set in the code, but in this case is the data that tells what the good answer is. To illustrate this, in the following paragraph there is an example.

Suppose it is wanted to know the driving style of different drivers and eventually classify those drivers into different clusters: In the ordinary algorithm type, classifications parameters must be set out before the code processing is run. So that's means it is need a previous knowledge of what is going to classify or identify. For example, depending in the fuel consumption classifications drivers in eco-friendly, but at the same time a definition of eco-friendly must be set out, for what the value is determined and irremovable.

Summarizing, the classifications are made by the previously definition of what is going to expect about the algorithm.

However, machine learning algorithms would learn those classifications by examples. In the supervised recognition, in the algorithm's code would be said what is and what is not eco-friendly driver. For what in the data set must be set out whether or not the data train (or data example) is eco-friendly or not. Then based on this previous assumption, a good algorithm will eventually learn and be able to predict whether drivers are eco-friendly or not in the rest of cases. In this type of machine learning algorithm the examples, with which data is trained, must be labeled or somehow explicitly pointed out what are and what aren't in the classification.

On the other hand, in unsupervised algorithm the examples, with which data is trained, are not labeled for what it is not said a priori any classification. These algorithms attempt clustering data into different groups according merely on data features and characteristics.

Yet there is another middle term that is between these two types. This is known as semi-supervised and active learning. Technically, this a supervised method in which there is a way to avoid a large number of labels and the own algorithm itself tells what should be labeled. Since there are classifications that are clear, this previous assumptions might help to the algorithm improve its results. This method is widely used in Artificial Neural Networks which are composed of nodes that are activated (fired) or not depending on conditions keys. These firing rules are set out previously in those cases that are clear and well-known. Regarding those that are not clear or unknown the own algorithm will found itself. Thus, these methods allow finding new patterns and trends in data.

The approach chosen in this study case was the supervised algorithm. Below it is shown how the labeling was carried out and the procedure followed.

Labeling is a simple process; the idea is to choose randomly a number of trips and choose them as the selected to label with a particular classification e.g. 0 or 1 depending if the trip is chosen or not. The rest of samples unlabeled are chosen as the contrary classifications i.e. if in the first step trips for a given driver were selected to be the candidates to be the "hypothetical", the others are taken as those that are not trailed by the driver.

The algorithm with this procedure attempts to find the relation between trips that are chosen to be those trailed by the given driver and to construct

2.2.2 The importance of cross validation

The Overfitting problem

The overfitting problem is important and highly present in machine learning. As said previously in the section 3.1, Machine Learning algorithms use examples, for which the desired output is known, to train the algorithm that will predict the rest data. The training data set is supposed to be useful when will be able to predict the correct outcome for the rest samples of data. Therefore, the training data set has to be tested and generalized in those situations not presented during training. Until here there is nothing new.

However, in those cases where training data sets are too long or where training is scarce and rare, the learner or the predictive function may adjust to very specific random features of the training data that, as a matter of fact have no causal relation to the target function. In those causes, when overfitting occurs, the performance accuracy on the training examples increases while the performance in the rest of data, in the testing data set, becomes worse. This matter was discovered for the first time by Larson (1931) who stated that training an algorithm and evaluating its statistical performance on the same data yields an overoptimistic result.

Technically, a learning algorithm is said to be overfitted when it is more accurate in fitting the known data but less accurate in predicting foresight. The overfit of the learning algorithm occurs due to in data samples there is information relevant for the future predictions and irrelevant information also known as *noise.* The more noise existing in the information domain, the more uncertainty and thus more noise data need to be ignored. [19]

The model trained is typically trained by maximizing its performance on the training data but its efficacy is determined not by its performance on the training data rather to perform well on the unseen data. Overfitting might be understood, in plain words, as if the model would perform as a student that instead of the learning and understanding the subjects just memorize the information. This student will get a successful mark as long the examination is the same as he/she studied. However, the more differences between the subject studied by this student and the exam, the worse mark will be gotten by the student. Similarly, the same happens with the overfitting in algorithms. Generally, the more unseen or unknown data in which the model is applied, the worse performance of it because the model only compares but is not able to generalize and thus gets biased.

Cross-validation

Cross-validation is a method to overcome the overfitting problem. Broadly, the method is based onto not just use the entire data set when training. The whole data set is split, depending on the total amount of data this can be done several times, and once data sets are trained, the performance is tested those parts that were not used to train. Thus, there is one part to train, and the rest is the validation data set which will estimate the risk of each algorithm.

Cross-validation avoids overfitting because both validation and training data sets are independent between each other. Nonetheless, the question that comes up when applying cross-validation is in how many splits should the data set be split or, if this is going to be split just once[16]: How much data is dedicated to train the model and how many to test it.[20]

In real problems data is no longer enough to split it and tested properly without assessing how should be do it. A usual way to carry out the split of data is K-fold cross validation:

In *K-fold cross validation* data set is divided into k subsets, and the holdout estimator of the risk is repeated k times.[17] For each iteration one of the k subsets is used as the test set and the other k-1 subsets form a training set. Then all k trials average of the risk estimator is computed.
Performing this procedure every data point has to be by definition in a test once, and in the training data k-1 times. The more k splits the variance of the resulting estimation is reduced. Conversely, one disadvantage of this methods is that the training algorithm has to be run k times leading a considerable time of execution since the method applied to make the prediction will be executed k times.[18]

Often instead of doing it in the sheer of k-fold, data is randomly divided into a test and training test k different times. The advantage of doing it like this is that data can independently choose how large each test set is and how many trials the data is tested.

2.2.3 Prediction with logistic regression

There are several possible circumstances where a process for estimating the relationships among variables is required. For instance, prediction whether a voter in an election process will choose one or another party in accordance with sex, age, social conditions etc… or perhaps predict likelihood of a company defaulting to its mortgage. In statistics, when the focus is on the

[16] If data is splitted once, is not technically cross-validation but holdout method. However, these are related and based on the same concept.

[17] The holdout method split data into training data and test data as cross-validation. However, with the inconvenience of splitting data uniformly and holding just a single train-and-test experiment that might mislead in case of getting an “unfortunate” split

relationship between a dependent variable an one or more independent variables is called regression analysis.

Specifically, regression analysis might be considered as a process to understand how a dependent variable[18] evolves or changes when independent variables are varied, while the other independent variables are held on fixed. A task where regression may fit would be estimation of the average value of the dependent variable when the independents are fixed.

The variation of the dependent variable around the regression function is known as probability distribution. However, there are cases in which is interesting the input-output relationship, but nevertheless, the output is discrete not continuous. For instance binary outcomes as, flight delayed, navigation certificate expired, pending overhaul. To answer this, a classification process is needed. However, with a Boolean "yes" or "not" answer, maybe is not enough, especially if there is no perfect rule and might be some errors in the predicted outcome. For that reason is deduced that we need probabilities or a probability distribution or a stochastic model.[38]

In that way, when a given input variables Pr (Y|X) in response to Y, this probability will tell us how our predictive model precision is. So, for instance in any sample is gotten a result of 51%, will be less reliable than another one with 99%, even though 99% chance is not a sure thing.

In order to understand this better let's pick up one class and call it "1" and the other "0". Hence, Y becomes an indicator variable[19], and thus we can conclude Pr(Y = 1) = E[Y]. However, this can be expressed similarly by Pr(Y = 1 X = x) = E[Y/X = x]. In other words, condition probability is the conditional expectation of the indicator.

Summarizing, the aim for regression analysis is to model the conditional probability Pr(Y = 1 / X = x) as a function of x, which the output must be binary and, where all unknown parameters must be estimated by *maximum likelihood*.

[18] Also known as "criterion" variable.

[19] In regression analysis a indicator variable is one that takes values 0 or 1 to indicate the absence or presence of some categorical effect that may be expected to shift the outcome. Can be thought of as a truth value.

Logistic regression

The main ideas to approach logistic regression are the following:

1. Let *p(x)* be a linear function of x. Every increment of a component of x would add or subtract so much to the probability. However, p must be a value between 0 and 1, fact unaccomplished because linear functions are unbounded.
2. The next idea is let *log(x)* be a linear function of x, so that changing an input variable multiplies the probability by a fixed amount. The problems turns out because logarithms are unbounded in only one direction and linear functions are not.
3. Eventually, the modification of log p into a logistic transformation $log\frac{p}{1-p}$, permits to handle unbounded range in both directions without run the risk of getting nonsensical results.

Hence, the logistic regression model is formally described by:

$$log\frac{p(x)}{1-p(x)} = \beta_0 + x \cdot \beta \tag{20}$$

Which solving p, turns out into:

$$\frac{e^{(\beta_0 + x \cdot \beta)}}{1+ e^{(\beta_0 + x \cdot \beta)}} = \frac{1}{1+e^{-(\beta_0 + x \cdot \beta)}} \tag{21}$$

Notice logistic regression gives as a result a linear classifier. The decision boundary separating both predicted classes is the solution of $\beta_0 + x \cdot \beta = 0$, which is a point if x is one dimension (1 solution), a line if it's two dimension, a plan whether it's three dimensions and so on.[38]

Likelihood function

Likelihood function is a function of the parameters of statistical model. Likelihood functions play a role in methods of estimating parameters from a set of statistics and how well they are estimated. Due to logistics regression estimates probabilities not just classifiers, can be fitted using likelihood. Can be observed as well, that for each training data sample, a vector of features, X_i for instance all tracks for a given driver, and observed class, Y_i, the i-th track belongs to the driver or not. The probability of the class is either p, whether $y_i = 1$, or on the contrary $1 - p$ if $y_i = 0$. The likelihood for a binomial (or logistic function) is given then by:

$$L(\beta_{o,}\beta) = \prod_{i=1}^{n} p(x_i)^{y_i}(1 - p(x_i)^{1-y_i} \tag{22}$$

Nevertheless, the log-likelihood is defined to be the natural logarithm of the likelihood function:

$$l(\theta; x) = log\ L(x; \theta) \tag{23}$$

Where if the parameter θ of $f(x;\theta)$ is a variable that is characteristic of $f(x;\theta)$, and let $x_1, x_2, \dots, x_n$ be a random sample from a probability distribution $f(x;\theta)$.

Then, the probability distribution for an overall sample is often equalized by the product of the likelihoods for the individuals x_i. Being x_i a vector of features and a observed class y_i. Where the expected probability for each class p for $y_i = 1, or\ 1 - p$, whenever $y_i = 0$.

$$L(\theta; x) = f(x;\theta) \tag{24}$$

$$L(\theta; x) = \prod_{i=1}^{n} f(x_i;\theta) \tag{25}$$

$$L(\theta; x) = \prod_{i=1}^{n} L(\theta; x_i) \tag{26}$$

However, when logs are taken, products are changed to sums and the operations are simplified permitting an easily understanding. Applying these concepts into the binomial likelihood function, the following formula is obtained:

$$l(\theta; x) = \log \prod_{i=1}^{n} f(x_i;\theta) \tag{27}$$

$$l(\theta; x) = \sum_{i=1}^{n} log\, f(x_i;\theta) \tag{28}$$

$$l(\theta; x) = \sum_{i=1}^{n} l\,(\theta; x_i) \tag{29}$$

$$L(\beta_{o,}\beta) = \sum_{i=1}^{n} y_i \log p(x_i) + (1 - y_i) \log 1 - p(x_i) \tag{30}$$

Where applying some mathematic summation properties as well as logarithmic and other. We can simplify into:

$$= \sum_{i=1}^{n} \log 1 - p(x_i) + \sum_{i=1}^{n} y_i\ log \frac{p(x_i)}{1-p(x_i)} \tag{31}$$

Where according to formulas (1) and (2) and substituting into the formula above, the following result is obtained:

$$= \sum_{i=1}^{n} \log 1 - p(x_i) + \sum_{i=1}^{n} y_i\ \beta_0 + \tag{32}$$

$$= \sum_{i=1}^{n} -\log 1 + e^{(\beta_0 + x\cdot\beta)} + \sum_{i=1}^{n} y_i\ \beta_0 + x \cdot \beta \tag{33}$$

Once this step is achieved, typically finding the maximum likelihood estimation is carry out by setting the derivatives and equalize to zero and then solving the resulting equation. Nevertheless, the result of derivation process is a transcendental equation and therefore is no closed-form solution. Hopefully, can be approximated and solve it numerically.

There are several popular methods to come up with the optimal solution of the derivation function. One of the most ancient and yet important is Newton's method. Even though is not going to be explained due to it shifts away from the main core of this work-issue.

GLM function

Logistics regression, as mentioned before, is really useful and more suitable than other linear functions because has less assumptions and restrictions. This can be achieved successfully through *glm* function which allows the performance of generalized linear models, usually known as regressions on binary outcome, count data, probability data, proportion data and so other data types.

Generalized linear models are extensions of traditional regression models that allow the mean to depend on the explanatory variables through a link function, and the response variable to be any member of a set of distributions called the exponential family (e.g., Normal, Poisson, Binomial).

Generalized linear models can be understood as a generalized flexible linear regression that allows for a response variables that have an error distribution other than normal distribution.

The idea behind ordinary linear regression is the expected values prediction of a given unknown quantity (random variable) as a linear combination of a set of observed values called predictors. Hence, a constant change in predictors yields a constant variation in the response variable. This assumption is appropriate when a response variable has a normal distribution.

However, there are several circumstances which normal distribution is not largely effective and do not fit properly the response variables. For instance, in those cases where the response variable is expected to be always positive and varying over a wide range, a constant input changes lead to geometrically varying, rather than a constantly varying. Therefore, in those cases a log-model would help to solve the lack of linearity.

In summary generalized linear models are useful in all these situations by allowing for a response variables that have ba simple normal distributions, and for an arbitrary function of the response variable (the link function) to vary linearly with the predicted values.

In the following paragraph is illustrated a brief explanation about the overview and the model components which underlies GLM.

The generalized linear model pretends through a particular probability distribution as can be; normal, binomial, Poisson among others. To generate a *dependent variable* [20] known as Y, where the mean, μ, of the distribution depends on the independent variables, X, within the equation given as:

$$E(Y) = \mu = g^{-1}(X\beta) \tag{34}$$

Additionally:
$E(Y)$ is the expected value of the dependent variable Y.
g Is the link function
$X\beta$ Is the linear predictor; a linear combination of unknown parameter β.

Parameter β is typically estimated with techniques like maximum likelihood or Bayesian techniques with the purpose of minimizing the cost of $E(Y)$.

2.2.4 Prediction with Random forest

Decision trees is a technique used widely in Data Mining, Machine Learning and other computational science in order to create a model with the goal of make a prediction of a target variable that is trying to be understood or classified based on several input variables of different natures: categorical, Boolean, discrete or continuous values within a specifically dataset.

Input variables constitute the features that lead a domain within the classification target. Thus, if it's wanted to classify for instance, if a patient suffers a particular disease, the input variable will be the symptoms of the patience. This symptoms (input variables), will be compared with the features that constitute the decision tree for the particular disease in each node, and depending on the input values, the tree will follow a sequence of branches leading to a result; in this case whether or not the patience has the disease evaluated in the tree.

In the concerning case inputs would be all the engineering features previously presented.
Internal nodes constituting the tree, test the value of particular features X_j, and branch according the result of the test. Every arch coming from the nodes is the possible choice emerging from the features.
Besides, the logical operands and results can be set out into a truth table that makes easier the understanding issue and the simplification of the tree.[21]

[20] A dependent variable is that one which its values are dependent on the values of the set of values that take the variables in which depends on.
[21] An example of truth table for the support of decision tree (Boolean) construction.

X_1	X_2	X_N	y_i
0	0	1	1
0	0	0	0
1	0	0	1[22]

Summarizing, each internal node analyze the attribute arising from features. Branches emerging to each node correspond to those input variables that fit with the attribute value in each case. Thereby each leaf node (nodes' sons) assigns a classification in the dataset.

Decision trees are based on Boolean operands. Every time a node executes its corresponding operation evaluating whether or not the input data set satisfies the restriction set by the node. Continuous features might be test using a threshold. It can't be used an equation to equalize each features. However, these one can be approximate by a threshold, not learning exactly how they behave but proximately. In practice is difficult to approach because of the complexity of the resulting tree necessary. [21]

It must be considered too, that every time a node bifurcates into different branches, dataset has being splitting as well in different subsets as shown for instance in the next graph.

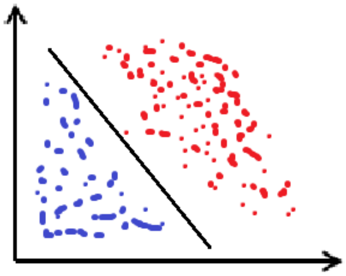


Figure 11. *Example of data partition when the decision tree branches*

Decision trees construction

In order to understand how decision trees work and how they are build up, imagine there is a dataset which contain several features for each driver.

X_1: Time trip in time

X_2: Number of stops.

X_3: Day of week

Y_i: Drivers

For any Boolean expression, there is a truth table that expresses each possible class. The combination of attributes $X_1 \ldots X_N$ gives the corresponding Y_i. So, in the worst case, constructing the tree to sort all classes will have to be equal to all the features included in the data set.

[22] Boolean truth table example showing the dependent variable Y in relation the features of variables X.

Whenever a decision tree is wanted to be laid-out, it must be considered that as the number of nodes (or depth) of the tree increases, the hypothesis space grows too. Thus, more iterations and operations must be done to carry out the classification goal. The goal should be to construct a tree with the less complexity as possible.

Decision trees can represent any Boolean function since:

Depth 1: “decision stump” can represent any Boolean function of one feature
Depth 2: Any Boolean function of two features can be represented; some Boolean functions involving three features e.g. $((X_1 \wedge X_2) \vee (\neg X_1 \wedge \neg X_3)$
Depth 3: Any Boolean function until seven features.

And as general rule, the features of the last level plus the double of the possible splitting nodes in the i-th level of the tree, are the total amount of feasible features that can be represented by a Boolean function in a decision tree.

Having said that, one may wonder if there exist any others feasible possibilities to construct the tree in other way and yet get the same result. Furthermore, may even questionable if the tree built is the optimum and desirable to fulfill the suitability requirements.

X_1	X_2	X_3	Y_i
2.15	1	Friday	1
1	1	Thursday	2
0.5	2	Wednesday	3
4	10	Friday	4
1	4	Friday	5

The answer to that question was founded by several independent celebrities of different kind of fields such psychologist or Machine Learning. They all happen to meet the following algorithm to run:

Grow tree algorithm

Being *S* the training data inputs:

```
If ( y=0 for all (x,y) ∈ S) return new leaf (0)
Else if (y =1 for all (x,y) ∈ S) return new leaf (1)
Else choose the best attribute $X_j$:[23]
            $S_0 = all(x,y) \in S\ with\ X_j = 0$
            $S_1 = all(x,y) \in S\ with\ X_j = 1$
Return new node ($X_j$ , GROWTREE ($S_0$), GROWTREE ($S_1$)) [21]
```

Generally, in order to accomplish the cost minimization, what must be done is, every time datasets are split, the tree has to grow in the most relevant features in dataset since branching the tree choosing the most relevant features ensures the optimization of the tree.

So, for instance, if it is obtained two possible branching solutions, the first one with 90% true positives and 10% of error, and another one with 60 and 40 respectively. Clearly, choosing the first option is better because it has more correct outcomes predicted.
Choosing the best attribute (S) is done by:

```
Choose j to minimize Jj
                    $S_0 = all(x,y)\ with\ x_j = 0$
                    $S_1 = all(x,y)\ with\ x_j = 1$
                    $y_0 = the\ most\ common\ value\ of\ x\ in\ S_0$
                    $y_1 = the\ most\ common\ value\ of\ y\ in\ S_1$
```

However, this easy way to choose in which the currently option available is evaluated to branch for each iteration is quite flaw and often turns to be mistaken.

For what is inferred it should be a form to calculate the cost of processing each tree.

This cost is known as *Entropy (S),* and it refers to the number of bits needed to classify the whole training data under optimal length code. Entropy measures the surprise of getting a particular result.
According to information theory, entropy may be seen as the expected information contained in each message transmission submitted into the system. Entropy describes the uncertainty about the information source, and increases as long as the sources are greater unpredictable.

[23] Branch by those attributes that give the lowest error rate. Error rate means if the tree stops right there, do not continue growing, will be the lowest mistaken.

The key point is that, the more likely is an event, the less information it provides when it comes up.

In countless books, paperwork's and other broadcast sources to back up understanding of entropy concept usually a coin toss example is stated.

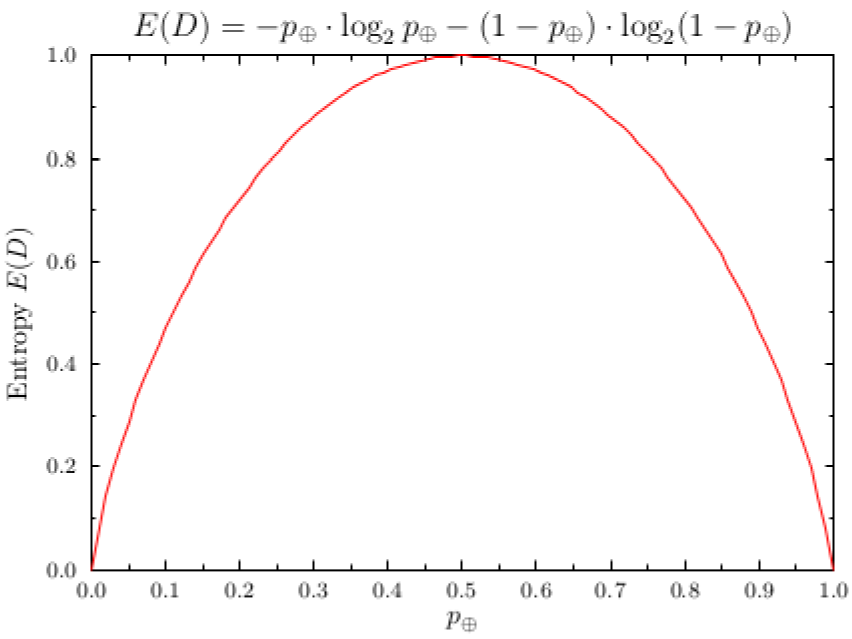
When a coin is tossed, if this one is unbiased and completely fair, the probability of heads and tails is theoretically the same[24]: 0.5. So, as the graph shown in the next figure, the surprise is minimum because the expected value is either face or tail as equal. Thereby, the entropy is as high as can be since there is no way to predict the final position of the coin and the prediction has a dim percentage of been right.

In the hypothetical case the coin would be entirely biased to get always one of the faces, so every time the coin would be tossed, the same result again and again is gotten, the entropy would be 0, because there is no need to send transmission and no bits are sent to.

Such an unbiased coin toss has one bit of entropy since there are two possible outcomes that occur with equal probability, learning the outcome contains one bit of information. Contrarily, a coin toss with a coin that has two heads and no tails has zero entropy since the coin will always come up heads, and the outcome can be predicted perfectly.

In order to explain in a simplified way what actually *Entropy* is, imagine for a moment it is sought to explain the expected random classification of a coin launched to the air.

[24] Neglecting conditional probabilities or assuming the coin is tossed just one time.



[34]

***Figure 12**. Entropy graph showing an unbiased coin toss case*

Nevertheless, if this coin would be biased leading a Bernoulli distribution to predict the probability to get the position of the coin once this one is thrown up, such as:

$$P(x = 0) = 0.7$$
$$P(x = 1) = 0.3$$

Clearly, it is easier in this case to know the expected result because the uncertainty has been reduced as the probability of one is greater than the other.

Until now, problems with trees discussed have been Boolean features i.e. datasets were split by branching through Boolean operands e.g. 0 or 1 if the condition is fulfilled.
However, there are situations, as in the concerning project, that features are not Boolean or there are multiple variables to script e.g. to determine if a car is going by highway or secondary roads or city, speed average can be separated into

In those cases, what is carried out is to set threshold to group data together and split regarding the threshold.

The explained above about decision trees is the base of different ensemble models based on, precisely decision trees.

In the following paragraphs, random forest method derived from decision trees principles is explained:

Random Forest is a unified method, for combining trees with the notion of an ensemble.

The general terms that Random Forest develops are:
1. Sample N cases at random to create subsets (these subsets can vary depending on the approach)
2. At each node:
- For some m number of predictor variables, m predictor variables are selected randomly from all the predictor variables available.
- Among all possible splits, the best predictor variable according to a target function is split.
- At the next node, choose other m variables at random from all predictor variables and proceed as before.[43]

Every time a new input is put into the system, it is run down all of the trees. As this inputs goesdown in the individuals trees more likely to find a optimum m

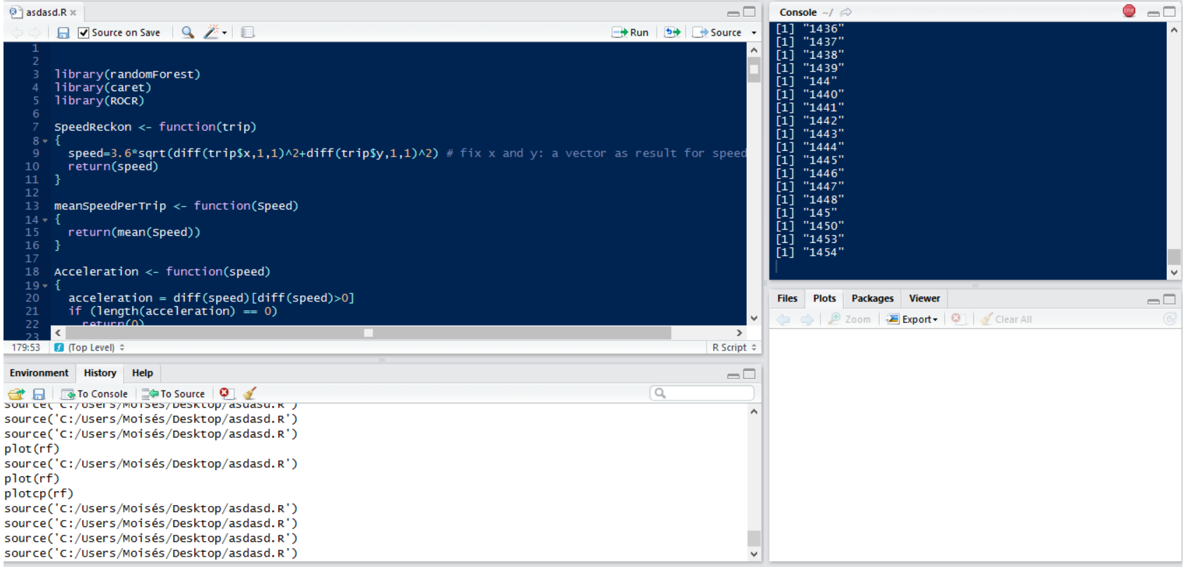
2.3 Data Science using RStudio

2.3.1 Introduction to RStudio and R programming.

“R is a free software environment for statistical computing and graphics.”[25] Its platform runs on a wide variety of OS such as MAC, Window, UNIX.

“The programming language and environment it is similar to S which was developed at Bell Laboratories by John Chambers and others but there are some important differences.” (CRAN)

R provides a wide variety of statistical tools; since linear and non-linear modeling, classical statistical tests, time-series analysis, clustering) besides graphical techniques. All of this developed and performed with the sheer of S language but with a touch of its own characteristics.
As a consequence of its S language heritage, R has an extensive high object-oriented scope than other statistical code source what makes R more versatile and competitive in different circumstances and with the environment it offers.



```
library(randomForest)
library(caret)
library(ROCR)

SpeedReckon <- function(trip)
{
  speed=3.6*sqrt(diff(trip$x,1,1)^2+diff(trip$y,1,1)^2) # fix x and y: a vector as result for speed
  return(speed)
}

meanSpeedPerTrip <- function(Speed)
{
  return(mean(Speed))
}

Acceleration <- function(speed)
{
  acceleration = diff(speed)[diff(speed)>0]
  if (length(acceleration) == 0)
    return(0)
```

```
source('C:/Users/Moisés/Desktop/asdasd.R')
source('C:/Users/Moisés/Desktop/asdasd.R')
source('C:/Users/Moisés/Desktop/asdasd.R')
plot(rf)
source('C:/Users/Moisés/Desktop/asdasd.R')
plot(rf)
plotcp(rf)
source('C:/Users/Moisés/Desktop/asdasd.R')
source('C:/Users/Moisés/Desktop/asdasd.R')
source('C:/Users/Moisés/Desktop/asdasd.R')
source('C:/Users/Moisés/Desktop/asdasd.R')
```

```
[1] "1436"
[1] "1437"
[1] "1438"
[1] "1439"
[1] "144"
[1] "1440"
[1] "1441"
[1] "1442"
[1] "1443"
[1] "1444"
[1] "1445"
[1] "1446"
[1] "1447"
[1] "1448"
[1] "145"
[1] "1450"
[1] "1453"
[1] "1454"
```

***Figure 13** R studio environment*

R might be seen as an integrated software facility for data manipulation, calculation and graphical display. In the figure 13 can be seen the environment with the different parts of the software such as; console, code blackboard, history, display dashboard, etc.

The main strengths of R among others are:

- An effective data handling and storage.

[25] CRAN: R support website[23]

- Calculations on arrays and particular matrices.
- An integrated collection of tools for data analysis.
- Graphical displays for data analysis.
- A simple and well developed programming language call, which includes logical operands as conditionals, plus optimized loops definer recursive functions and input and output facilities.

What really make R leverages in comparison with other languages are the newly developing methods of interactive data analysis, which has developed rapidly and has been extended by a large collection of packages.

R packages the use of user-submitted packages for specific functions or specific areas of study. Inside them can be founded tools to develop certain functions, methods for specifically kind of task, cases examples, even the whole resolution for a long list of problems laid out.

Furthermore, packages contribute to spread the applicability program as well as its broadcasting through different fields allowing R to grow either in functionality as in usage.

2.3.1 Overview of the caret package

The use of complex classification and regression models is growing up as the most common tool in science, finance and innumerable domains. The R language has an extend set of modeling functions for both classification and regression methods, that are becoming more and more difficult to maintain record of the syntactical shade of each function. The caret package (issued for classification and regression training) was developed by Max Kuhn and a large list of collaborators for different objectives:[11]

- To eliminate syntactical differences between the functions for predicting and building models.
- To develop a set of semi-automated, reasonable approaches for optimizing the values of the tuning parameters for many of these models and
- Create a package that can be used in parallel processing systems.

In summary, the caret package contains many tools for developing predictive models using the rich set of models available in R. The package focuses on simplifying model training and tuning across a wide variety of modeling techniques. It also includes methods for pre-processing training data, calculating variable importance, and model visualizations.

The main benefit caret package is that creates, in a single environment and in a unique interface, a set of functions for modeling and prediction, streamline model tuning using resampling and provide a wide variety of helps on day-to-day building tasks. *(Max Kuhn, Ph.D, 2013)*

Caret package functionalities

One of the primary and trunk functions of the caret package is the *train* function which evaluates, using resampling the effect of the model tuning performance, estimate model performance of the training set and, furthermore, choose the optimal model through the parameters gotten in the model tuning.

From this function derive other functions to split data into training and validation data set as *createDataPartition* which can be used to create a random split in the data sample into training and test or *trainControl which is used to specify the type of resampling;* By default uses a simple bootstrap resampling but can be customized using repeated k-fold cross-validation, leave-one-out among others. Besides, as practically in all the functions, there is option to customize the default setting and for instance, with *Repeatedcv* the k-fold cross-validations can change the number of repetitions, although by default is 10.

Caret package also includes gradient boosting machine (GDM) model with three main tuning parameters:

- Number of iterations
- Complexity of the tree
- Learning rate which means how quickly the algorithm adapts
- The minimum number of training set samples in a node to start splitting.

No matter which tuning model is used, caret package chose an optimum performance of the model. However, to measure an estimate the model performance there are several functions to compute measures about two-class problems such as the area under the curve (ROC), the sensitivity and the specify.[24]

Finally caret package as well includes plotting tools in order to show the relationship between the model tune and the ROC curve or any other feature about performance to be sure the model works properly and in accordance with the expectations.

2.4 Machine Learning with Big Data

It was aforementioned the existence of uncountable sources of information that are continuously flowing up and down, back and forth, information along the networks. Therefore, this huge volume of data are stored into separately warehouses looking forward being process and handled to be converted in useful information and inferred in new knowledge.

The occurrence of having available the astonishing amount of data, nonetheless, is not a guarantee for getting better results by itself. Doubtlessly, the possibility of handle such big data warehouses allow to deal with more realistic problems and get more accurate predictions or simply tackle until now unexpected fields. However, as well as tools and technologies are needed in order to collect information and store it, technologies and techniques are required to process the data.

Likewise, the X can be disclosed into the algorithm efficiency, basically described as the strategies and techniques to manage the data to get the sought information and, on the other hand, the resources and technologies implemented to run the algorithm. The more sophisticated and tedious is the algorithm structure, the more time is required to operate it, (reaching surprisingly, in a lot of cases, time lapses of millions of years). Clearly, in a circumstance where the cost, name it on time, is higher than the benefit of the processing result, the running procedure can be dismissed. Figure an application for GPS to find out the optimum route and the algorithm search would take for instance 10 days. Likely would be possible to find out circumstances where the GPS processing would be profitable yet. However, these are constrained by predictable events further than 10 days, and moreover process result benefit should be higher than the cost of keep running the algorithm the time enough to achieve the desirable result. Otherwise these 10 days running an algorithm to find the better way to arrive somewhere would be omitted.

It has been already discussed that depending how a problem is approached, this one performs different and yields a different accuracy rate resolution and specific resource consumption.

On the other hand, figure out the problem tackled needs a fixed accuracy rate outcome and the algorithm approach is considered the optimum. On this situation presented, if would be necessary to cut down the time processing (even sometimes the resource efficiency) the only way to do it would be through the physic technology used and performed to deal with the program at issue.

This chapter is about the emerging technologies and the technology scope range available currently on the market to enhance the execution of programs laid-out to deal with considerable complexity and tough algorithms as well as large datasets.

From Big Computers to Super computers

Taking a look around to our environment is easy to observe that we are surrounded of digital devices which have really different purposes; from the day-to-day machines as laptops, tablets, smartphones, wash machines, intelligent televisions to smart streets and cities. However not always has been like this and likely will not be anymore. There is no easy answer to date which was the first computer, because there were different concepts of what a computer is depending on the architecture and the elements based on. However, in this paperwork the reference takes into account on this computers able to apply iterative operations in a determined way to find a solution for a problem.

In order to determine the birthday of digital computer must move the time until December 1947 when, in BELL Laboratories, John Bardeen, Walter Houser Brattain and Willliam Bradford Shockley invented the bipolar transfer resistor for what they received in 1956 the Physics Nobel award.

This discovery will lead a breakthrough in the computer and digital history. Even though for those times the transistors dimensions were excessively large. However, being aware of the benefits and consequences which will have brought the technology revolution several in the upcoming days, companies invested in the transistor development; specifically in its reduction on size and capabilities or structure laid out.

Early the next successful inventions were come out: Intel on its behalf in 1977 launched the 8085 processor, with 4500 transistor within the device. On 1978 and 1979 the series 8086 and 8088 respectively with its own improvements had followed integrating progressively more transistors and capabilities into the microchips. Meanwhile, other companies as Motorola or AMD realizing the emerging power behind this technology had joined the research for the improvement of transistors and microchips. [28]

New improvements were achieved during the following years, doubling year by year the number of transistors per area unit. But specifically in 1986 an unperceiving outstanding achievement was fulfilled by Intel.

Since then, the microchips have been startling increasing the total amount of transistors hold inside it achieving the outstanding and shocking figure of 1400 million of transistors in barely 22 nanometers. This fact had led to come up with a boundary hard to cross since the space reduction between transistors is so tiny and thereby the electromagnetic signal propagation increases the energy loss and the conductor temperature. This rising temperature diminishes the transistor velocity. On the contrary, the rising temperature increases as well the transmission leaks turning up instability in the circuits.

Gordon E. Moore Intel cofounder observed the computer progress and realized that the number of transistors per square inch on integrated circuits had doubled every year since the invention of it. Moore predicted that this trend will continue in the future. However, it seems as if this trend will not be fulfilled anymore. Perhaps the increase in power of calculus has ended it up. For that reason, AMD and Intel have been testing new materials as grapheme instead of using the common silicic or even trying new ambitious solutions as optic propagation signals or using quantic computing or molecular transistors.

While there is no clear alternative or a solution to this inconvenient yielded by the temperature tied to the circuit, the use of multi core or the settlement of different kind of treatment of memory such as SRAMs and DRAM, implemented to improve the power without need to increase the number of transistors.

Multi core carries out the read and execution of program instructions at the same time by using two or more processing units increasing overall speed of programs amenable to parallel computing.

Until now it has been explain the aim to get better hardware or physical features to get high-throughput computing. However, with the current technology available must be tackle problems with high complexity to solve unknown of different nature. Having said that, attempt to scope a real problem of high complexity with an ordinary computer probably would take up too much time to get satisfactory results. Thus, in light of that situations what is used usually are computers with high-level computational capacity also known as supercomputers.

A supercomputer may be understood as a system capable of processing really different instructions, usually high mathematical complexity, with the less time possible in the currently highest operational rate of computers. Supercomputers utilize multiprocessing system, so each part processes is connected among the others and they share messages broadcasted to announce that particular operands have been assigned but, on the contrary to sequence multiprocessing systems (SMP), massive parallel processing in order to get around the propagation messages time and share the optimum resources, do not share all the information with all the parts forming the system, but rather just send the operands to those need it. In flat words, the system is subject to a network in charge of transmitting the instructions and commands.

Current supercomputers can be working in different chores hundreds of thousands processors running vast quantities of data simultaneously. Systems with massive processors generally are split into those that use large number of discrete independent computers or terminals, linked across a network as e.g. Internet, involved in a common problem. In that way, the problem tackled is discomposed into different parts. All that parts are thereafter designated to all agents involved in the problem and each of them perform a certain piece of the whole problem which, with the aggregation of all the rest outputs gathered in a central and mighty server, constitutes the overall solution for a case.

However, the input data is usually large and the computations have to be distributed across hundreds or thousands of machines in order to finish in a reasonable amount of time. The issues of how to parallelize the computation, distribute the data, and handle failures conspire to obscure the original simple computation with large amounts of complex code to deal with these issues.

Map-reduce

Over the time many different authors from different companies had implemented several platforms to process large amounts of raw data. Unfortunately most of them were conceptually straightforward and they spin out too much execution time.

Whenever input data is large, in order to finish in a reasonable time, data usually is distributed across hundreds of thousands of devices. In that way, the computation is parallelized, data distributed and those problems drawn by the intrinsic computing complexity of massive parallelization are managed together.

Google inspired by the map and reduce primitives functions present in several functional languages, created simple and powerful interfaces that enable automatic parallelization and distribution of large-scale computations, turning in result the high-performance on large cluster of commodity PCs.[27]

MapReduce is a framework for processing parallelizable problems across huge datasets, using a large number of computers, collectively referred as cluster if all nodes are on the same local network and use similar hardware, or otherwise *grid*, if the nodes are shared across geographically and administratively distributed systems and using heterogeneous hardware.

The program is composed of two basic functions: Map() and Reduce(), functions which each node is committed to perform it.

Both, Map and Reduce functions, are defined with accordance to data structured in (key, value) pairs. Map procedure is a list of pairs for each call and, after that, the MapReduce framework collect all pairs with the same key from lists and gather them together, creating one group for each key.

$$Map\ (k1, v2)\ --> \ list\ (k2, v2)$$

The Reduce function is applied in parallel to each group, which in turn produces a sort of values in the same domain:

$$Reduce\ (k2, list\ (v2)) -> \ list\ (v3)$$

The reduce function takes the input values, operates upon them and generates a single output and the final result.

- *__Map Reduce execution__*

The functions Map invocations are distributed through several machines by automatically partitioning the input data into a set of M splits. M splits can be processed in parallel by different machines. Reduce () invocations are distributed by partitioning the intermediate key space into R pieces using partition functions.

When a program calls Map-Reduce function the sequence that occurs is illustrated in the figure 12.

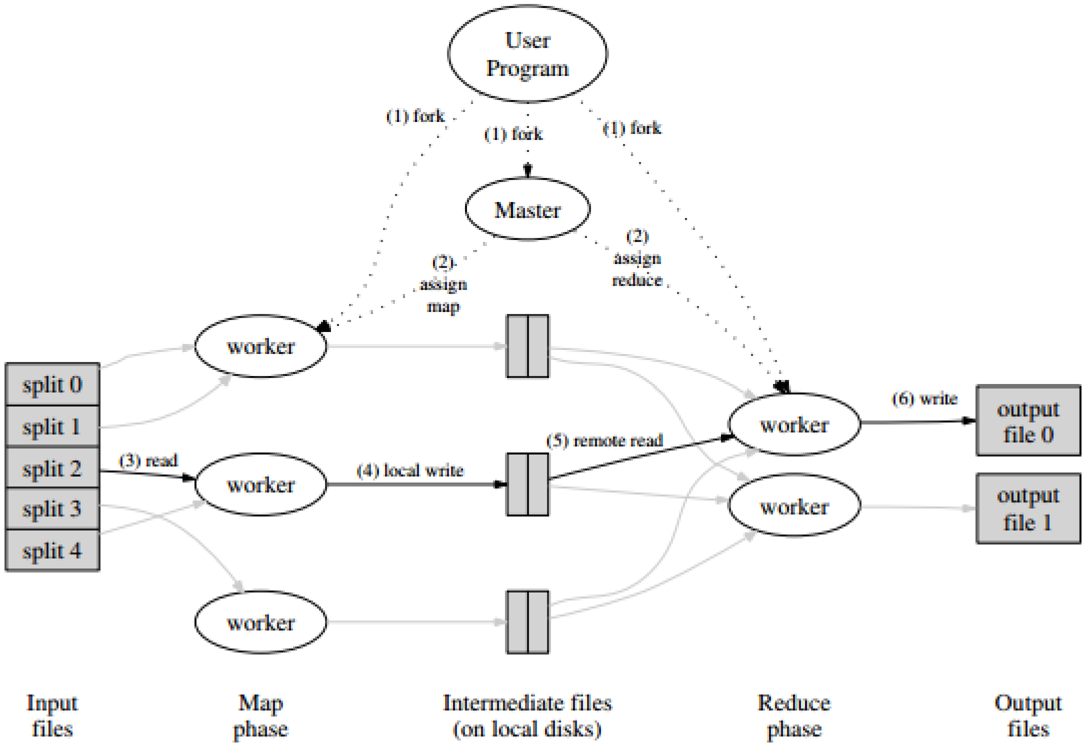


***Figure 14.*.** *MapReduce execution process*

1. First, input files are split into M pieces of typically from 16 megabytes to 64 megabytes per piece. Many copies as long as the specified by the user are created on a cluster of machines.
2. One of the copies determines a special – the master. The rest are workers assigned for work by the master. The master picks idle workers and assigns each one a map task or a reduce task.
3. A worker who is assigned a map task reads the contents of the corresponding input split. It parses key/value pairs out of the input data and passes each pair to the user-defined Map

function. The intermediate key/value produced by the Map function are buffered in memory.

4. Periodically, the buffered pairs are written to local disk, partitioned into R regions by the partitioning function. Master forwards buffered pairs locations to the reduce workers.
5. If a reduce worker is notified by the master about these locations, it uses remote procedure calls to read the buffered data from the local disks of the map workers. When a reduce worker has read all intermediate data, it sorts it by the intermediate keys so that all occurrences of the same key are grouped together.
6. The reduce worker iterates over the sorted intermediate data and for each unique intermediate key encountered, it passes the key and the corresponding set of intermediate values to the user's Reduce function. The output of the Reduce function is concatenated to a final output file for this reduces partition.
7. When all map tasks and reduce tasks have been completed, the master calls the user's program. At this point, the MapReduce call in the user program returns back to the user code. After successful accomplishment, the output of the execution is available in the R partitioned output files (one per reduce task). Typically, users do not need to combine these R output files into one file – they often pass these files as input to another MapReduce call, or use them from another distributed application that deals with partitioned input into multiple files.[28]

The master is in charge of saving several data structures. For each map and reduce task, it stores the state where is (idle, in-progress, or completed), and the identity of the worker machine (for non-idle tasks). The master is the channel through which the locations of intermediate file regions are propagated from map to reduce tasks. Updates to this location and size information are received as map tasks are completed. Then, information is pushed incrementally to workers that have in-progress reduce tasks keeping them with desirable throughput.

- *<u>Clusters</u>*

On the other distributed approach, separating the problem into different parts and send them to the corresponding agent to be processed, and once the assigned task is finished, send the output process back again, turns out to be a communication spending of time. Because of that, clusters save time being together and, though tasks carried out are split as well into divisions, the agents work closely unlike as happens in grid structure, computer cluster have each node set to perform the same task, controlled and scheduled by software.

Components of a cluster are connected between each other through Local Area Networks (LAN), with each node running its own instance of operating system. Even being possible to use different hardware and operating systems, hardly ever are different amongst each other.

Clusters are deployed to enhance performance and availability over having single computers working independently, while typically being much more cost-effective than single computers of comparable speed.

Clusters are closely related with supercomputers, since a supercomputer is a large cluster of computers.

The most powerfully and faster supercomputers nowadays are presented in the following table[31]:

RANK	SITE	SYSTEM	CORES	RMAX(TFLOP/S)	RPEAK TFLOP/S)	POWER(KW)
1	China	Tianhe -2	3,120,000	33,8862.7	54,902.4	17,808
2	United States	Titan	560,640	17,590.0	27,112.5	8,209
3	United States	Sequoia	1,572,864	17,173.2	20,132.7	7,890
4	Japan	K computer	705,024	10,510	11,280.4	12,660

[26]Each approach, cluster or distributed architectures as grip, has its own advantages and disadvantages. Since they provide different intrinsic features, every concerning case shall choose one or another depending in the needs found.

Somehow, MSP are always operative, they do not stop running because of some computational reasons and energy productiveness. Thus, adding up the forthright fact that having thousands of processors which in turn are composed of thousands of elements that need energy to work, eventually the energy need to operate MSP is out of proportion and that cause users must pay this excessive energy consumption raising the prices considerately. [32]

In addition, the amount of supercomputers around the world is limited, what makes its availability restrained raising even more the high-prices.

Because of this, all those pretending to use supercomputers, before running their programs and codes, they have to be sure the algorithm provides a reliable and a sought result as well as ensures there is no other approach, at least known, with which the running time translate it into algorithm efficiency is lower than the chosen.

- *Cloud Computing*

It has being mentioned in the previous section than there are two main models: Clustering and distribution computing through a network. Nevertheless, with the improvement of facilities; optic

[26] This table shows the current most powerful cluster of supercomputers with some of their throughput capabilities as energy consumption or processing capacity.

wires capable to transmit massive amount of data with very few marginal loses, and telematics widely-spread over the territory, a new trend have being emerging in the new days.

Many on-line storage platforms as Mega, OneDrive, Google, DropBox and a wide variety of servers are well-known because of the non-physical but on-line data storage service they provide to the users. Basically, their service is based on taking profit of the good transmission and communication facilities to receive and send the requested data in a shared warehouse but divided among all the users, where every user has its own little fraction of a the huge warehouse without permission to access the rest user's information. This storage has the particularity of consume compute resources as a utility rather than having to maintain and build up a computing infrastructure which is not completely used.

In Cloud computing the resources needed are just the used. No more, not less. Depending on the type of service offered, e.g. processing, storage, or software, resources can be controlled and optimized by monitoring, controlling, and reporting. For that reason a big elasticity and efficiency thanks dynamically reallocation per demand.

Cloud computing acts as a way of sharing resources to take profit if scale economies as well as utility throughput maximization by converging service into a single one, and share across the network the services on-demand. In this manner, cloud computing allows the users avoid sunk costs as facilities and maintenance costs, and focus just in the outcome they seek.

Likewise outsourced storage, in the recent days some companies offer the chance to use, through the cloud (Internet) their powerful facilities providing not just storage, but processing services, information management and so on across the network such as Ethernet.

Cloud Computing summarizing is a way to outsource some resources avoiding the high expenses for purchasing computing facilities but taking advantage of the benefits of high computational performance in those sudden occasions whenever would be profitable to process swiftly certain programs that require plus processing power. Moreover, since there is no physical facility, at least in the user or customer case, files and data are in the cloud. Thus, these are accessible from every place with Internet access.

A real case study

Massive processors have a wide-range of applicability and fields where can contribute to add value. Applications include since capital trend market studies, weather forecast, nuclear decay and energy research, molecular and drugs simulations, new visual-graphical development among a large list of possibilities.

The case brings out is from astronomy field, specifically GAIA's satellite data process. Astronomy's field usually, due to the high amount of data it deals and the high level of operations and mathematics involved, requires high performance processors to handle the data and transform it into useful information for science.

GAIA's aim is literally take a photo of Milky Way Galaxy, precisely is going to survey the most precise three-dimensional map by observing 1 per cent of its population of 100 billion stars.

In shortage, the picture of the galaxy is done by scanning the light striking into few Nano metrical observation angles meanwhile the satellite is rotating constantly at an angular rate of 1° per minute around an axis perpendicular to two field of view that describe a circle in the sky in 6 hours. The orbit even being in a very stable thermal and radiation environment, and allowing a uninterrupted mapping of the sky, sometimes suffers some deviations in the seizures and the orbit craft must be align again in the orbit. [25]

The information resulted from the mapping is not only just a three-dimensional map of the galaxy, but also information about substances and components which form every star, angular and propagation velocity of them, discover of new kind of starts, and so other features that describe and improve the knowledge of the obscure Universe.

The complexity of the project is maximum; every single error produces in the process an addition of hundreds of thousands of new variables to solve and can cause a big delay in the progress. The project itself deals with millions of variables and unknown incognita but, as long as the project goes by, these figures are boosting up day-to-day because of new unexpected things as problems and drawbacks came out. The flow chart of data is seamlessly rushing information up and down. The distributed centers must process the raw data get that comes from the satellite. However, this one first has to be modeled by mathematical and statistical procedures and once this is done, the raw data can be processed.

By this reason, the data is not send only in a single hub where is handled. Rather, data is interconnected among different institutions and research facilities spread over the whole world to address, in each case, the corresponding work and again distribute the outcome to the rest network to hold the network synergies to ensure each node or agent, deals with its concerning issue and thereby maximize the efficiency based on the specialization of each node.

For example, immediately data coming from the satellite scanning is not send straight to the science modelers that have to understand and wonder what is going on out. But before, this data is treated eliminating redundancy and possible failures, verified the trustfulness, applied some reduction to the problem and then is distributed specifically only each part that is interesting for each node. In some there will go the satellite function, in others some sort of data relevant to determine some specifically structure about the universe, in some others information classified in levels depending in each case. So each node being part of the program receives the specifically information useful for itself. No more, no less.

In a big scale these might see as the nodes of the neural network working parallel and sharing information between each other to accomplish a whole main objective.

4. Case Studies

In the following section an example of GPS data is given. Through the use of the Machine Learning methods, it is sought to apply the methods explained in the previous sections in order to construct a model able to recognize a personal driving style of each driver.

This model permits to have knowledge about the driving habits of the analyzed drivers. With this knowledge, the vehicles owning company may make decisions to identify gaps for improving the company reputation, reducing costs and exhaust emissions.

Moreover, insurance companies might take profit of it to tender its offers according the real behavior of the driver, being more permissive with those with good attitudes and punitive with those with sharp or strange patterns in the car conduction.

Finally, this drivers' classification may be analyzed in order to improve drivers' safety and remove some hazardous driving attitude saving their live and pull down the index accident.

This study case is an example for a specific driving study in cars. However, this can be extrapolated and used as an example for any other applications fields that requires human interaction with the handling of devices or vehicles; pilot driving, Formula 1 competitions. Logically, in each application case, data have to be transformed according to its features and engineering requirements. So, it must be adapted regarding the desirable result expected.

4.1 Case study 1: Recognition of driving styles for a simple data set

In case study 1 the basis of the predictive model is performed. In this chapter it is sought to lay out the roots of the model and the structure will be composed of for later, once will be estimate the truthfulness of it and the computational viability perfectly assessed, perform the same model in a real case with big amount of data involved.

In this case only 4 drivers are taken in the sample. However, the principle and the process to carry out is exactly the same. The model is not train directly in the real case to save time and reduce the complexity.

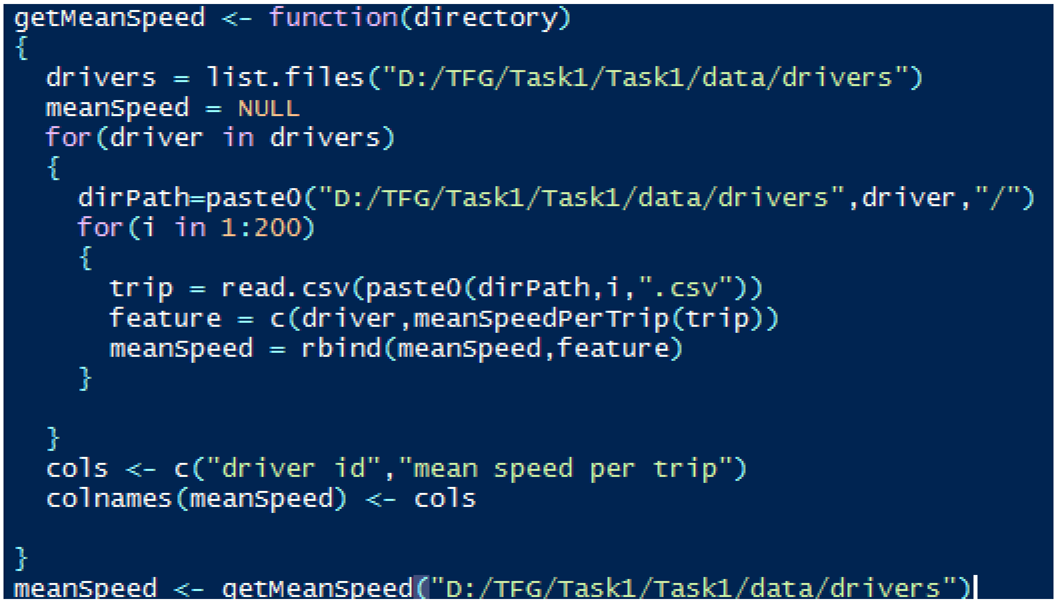
The main code tasks in sequence are the following:

1. Get raw data

2. Transform raw data in useful information
3. Clean data and transform it
4. Divide data set into test and training data
5. Apply the prediction method approach in the training data (train data for learning)
6. Evaluate data trained in validation test
7. Analyze the results from ROC Curves.

Below is detailed how was constructed the code to achieve the prediction model with logistic regression.

The code structure proposed is divided into different functions that are called from a big main function called *getMeanSpeed.* Likewise, this is divided into different parts:
The first one is on charge of extract the information from a file where data is stored. To perform this, the function *list.files* produces a character vector on the list of files named on the directory. Henceforth, the new variable drivers will contain all lists per driver where trips are allocated.
The directory does not have to contain any especial character otherwise may cause different problems of recognition.



```
getMeanSpeed <- function(directory)
{
  drivers = list.files("D:/TFG/Task1/Task1/data/drivers")
  meanSpeed = NULL
  for(driver in drivers)
  {
    dirPath=paste0("D:/TFG/Task1/Task1/data/drivers",driver,"/")
    for(i in 1:200)
    {
      trip = read.csv(paste0(dirPath,i,".csv"))
      feature = c(driver,meanSpeedPerTrip(trip))
      meanSpeed = rbind(meanSpeed,feature)
    }

  }
  cols <- c("driver id","mean speed per trip")
  colnames(meanSpeed) <- cols

}
meanSpeed <- getMeanSpeed("D:/TFG/Task1/Task1/data/drivers")
```

On the next loop, per each trip is *read*[27] the .csv file and is stored in trip. When using of .csv file is assumed beforehand that the data file is in format called "comma separated values". That is, each line contains a row of values which can be either numbers or letters, and each value is separated by a comma. Moreover, is assumed that the very first row contains a list of labels.

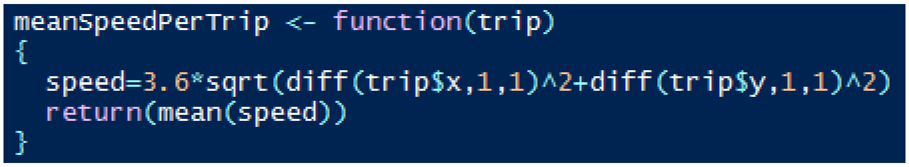
[27] Read.csv it has three different arguments. On the first is indicated the file. In this case as is not just a file must be used dirPath variable. The second argument, indicates iteration (in this case the trip number) and finally the last indicates how the file is structured (".csv), also will be allowed using sep = ",".

Those labels in the top row are used to refer the different columns of values, so will be the variable identification. By default if it is not changed by the use of: head = FALSE. The function will take the true value so, it will subset each variable by *$VariableNoun.*

Notice that the number of trips will be as much as trips per driver there are in the directory specified; in this particular case 200. Assume that we know how many trips we will assess beforehand and they are uniformly per each driver. Otherwise this way will not be feasible to carry out on this way.

In the code illustrated before, the main aim is to equalize the mean speed per trip. However, in data available there is no speed anywhere. Fortunately, as long as position is extracted in a time lapse speed, speed of motion or in other words “swift of object” is easily calculable.

Now that it is known how speed is equalized can be compute it as:

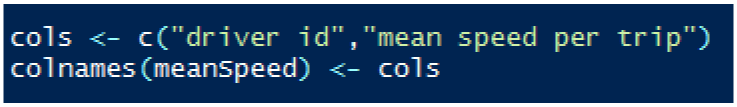


```
meanSpeedPerTrip <- function(trip)
{
  speed=3.6*sqrt(diff(trip$x,1,1)^2+diff(trip$y,1,1)^2)
  return(mean(speed))
}
```

This *meanSpeedPerTrip* function equalizes the speed and then the mean Speed per Trip. Using function *diff,* R perform a subtraction between the values within the selected vector. The output from this function is a vector with the results from this subtraction between the n-value and the next value: n+1. This function also needs to specify the lag used; which in this case is 1.
Constant 3,6 multiplying the whole function is just done to convert the seconds into hour and hence, get speed in km/h magnitude:

$$X\frac{m}{s}=\frac{1km}{1000m}\cdot\frac{60\ s}{1\ min}\cdot\frac{60min}{1h}=X\ km/h$$

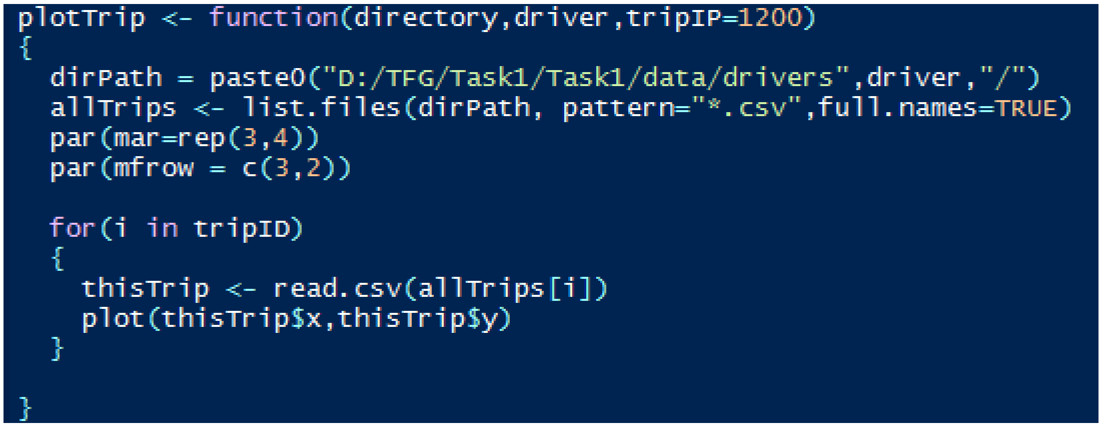
Once the vector *speed* is calculated, the function returns the mean speed equalized thanks to the R function *mean*.
After this, *feature* variable is used to save the meanSpeed for each different feature equalized on this way: *meanSpeed = rbind(meanSpeed,feature). Rbind* is a R simplief function in R which is used to add rows in accordance with the variables specified between (). So in this way, all results equalized are stored regardless of the previous iterations. Concurrently, exists a *Cbind*, function that do the same task as *Rbind* but adding in columns.
In order to visualize better the work done, the names of columns are changed by the use of:



```
cols <- c("driver id","mean speed per trip")
colnames(meanSpeed) <- cols
```

Where *cols* is a variable declared to assign the string names per column contained in *meanSpeed variable*. In this code shown before there are 2 columns.

Immediately, once *meanSpeed* is correctly equalized and safely save in a data structure, is available to be plotted or graphed; besides operate with other features to obtain other new results. In the code is called a function *plotTrip* to visualize how data is drawn upon graph visualization.



```
plotTrip <- function(directory,driver,tripIP=1200)
{
  dirPath = paste0("D:/TFG/Task1/Task1/data/drivers",driver,"/")
  allTrips <- list.files(dirPath, pattern="*.csv",full.names=TRUE)
  par(mar=rep(3,4))
  par(mfrow = c(3,2))

  for(i in tripID)
  {
    thisTrip <- read.csv(allTrips[i])
    plot(thisTrip$x,thisTrip$y)
  }

}
```

In this function are provided directory of files[28], driver and the *tripID* (identification of those trips to plot). For reading and enlisting the object's position into a vector, is created a variable *dirPath* used for pasting the directory into the next code's line, but this time in dependence with the driver. Therefore all trips are stored for each driver separately.

Below are displayed the outcome *plotTrip* function for some random trips of drivers. Something as the picture might be seen for every trip trailed by each driver. [36]

[28] Exactly the same Data as used before to reckon meanSpeed.

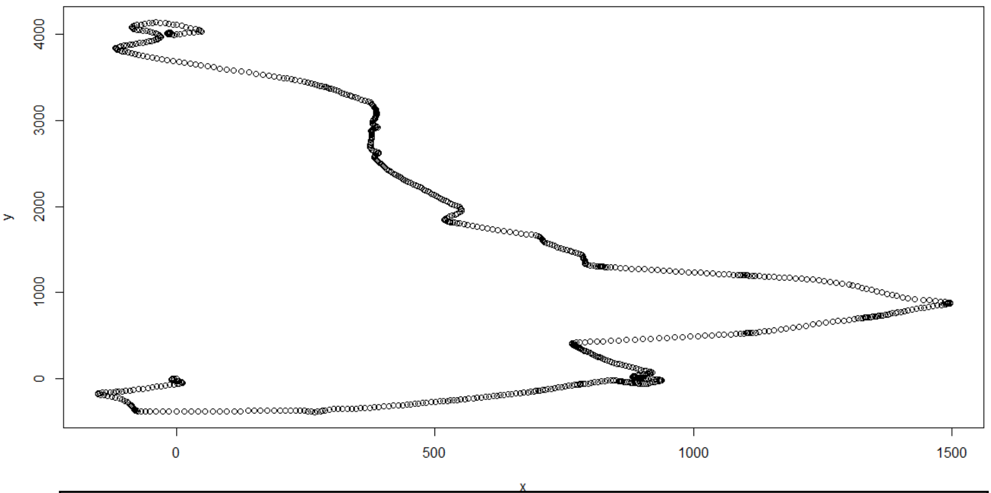


Figure 15*. Random trip plotting of data drivers positions*

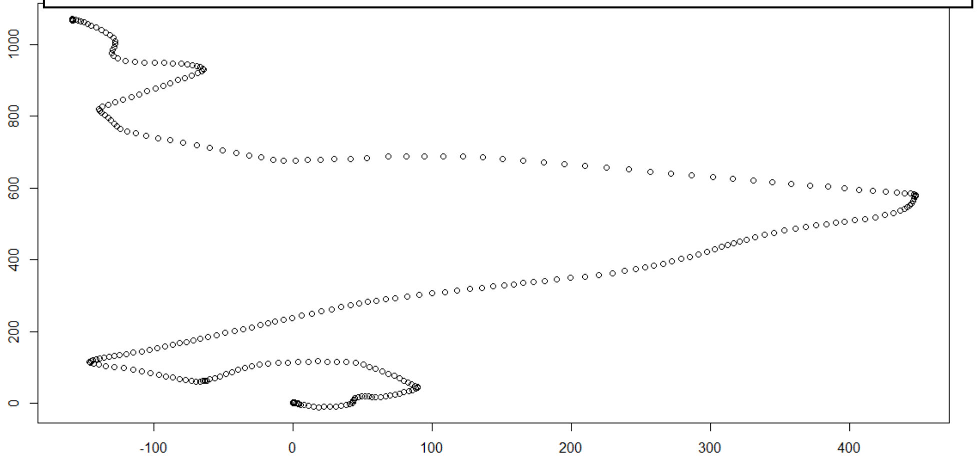


Figure 16*. Random trip plotting of data drivers positions (different trip)*

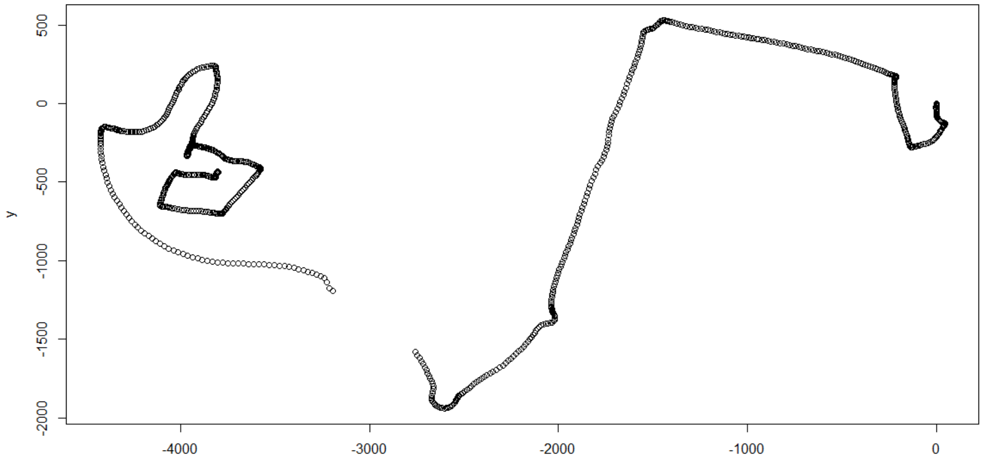


Figure 17. *Random trip plotting of data drivers positions (different trip) with data disturbance.*

How it may be noticed in the last graph some data is disturbed and it is not found so, as in forward chapters is explained, data must be transformed and clean.

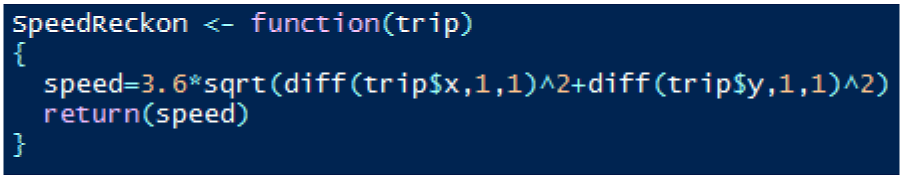
The use of *allTrips* variable allows enlisting *allTrips* in order to, afterwards, when the loop *for (i in tripID)* will turn on, read each Excel file previously listed. For each trip in the drivers file, is stored in a new variable that forward will call a function *plot* to draw the path described in each case.

R allows to combine multiple plots into one, using the function *par()*[29]*.* The option *mfrow* create a matrix of nrows x ncols that are fill by row. Conversely, *mfcol* do the same but in the opposite side. In addition, even not done it here, if it is wanted to, with the same functions two different graphs can be add to provide in a single view a whole overview of the enhanced graph.[30]

Acceleration feature

This time the code's aim is to call different functions that provide again, for each driver and per trip, different sought features that tell more characteristics and properties about the trips. Nevertheless, this time the speed has been split in a function where is equalized, and will provide the input for the functions that will needed. On the last task was explained how data files were read and match into variables. So this time this part is skipped, and proceeds straight onto those new code's functionalities.

The new functions are the following.
Acceleration: The input in this function is the vector *speed* which is the aforesaid vector where is stored in the equalization done through the use of the renowned speed function:

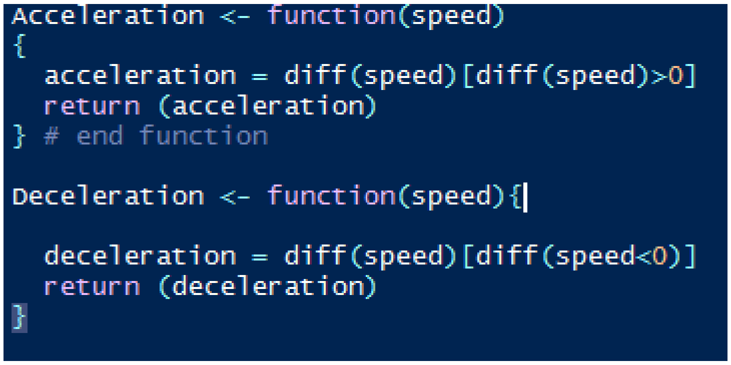


```
SpeedReckon <- function(trip)
{
  speed=3.6*sqrt(diff(trip$x,1,1)^2+diff(trip$y,1,1)^2)
  return(speed)
}
```

:

What it means proceed as done before with the speed calculation but this time changing position by speed vector. Resulting into the following code:

[29] Blog The function of the Day
[30] Quick R: Combining plots http://www.statmethods.net/advgraphs/layout.html



```
Acceleration <- function(speed)
{
  acceleration = diff(speed)[diff(speed)>0]
  return (acceleration)
} # end function

Deceleration <- function(speed){

  deceleration = diff(speed)[diff(speed<0)]
  return (deceleration)
}
```

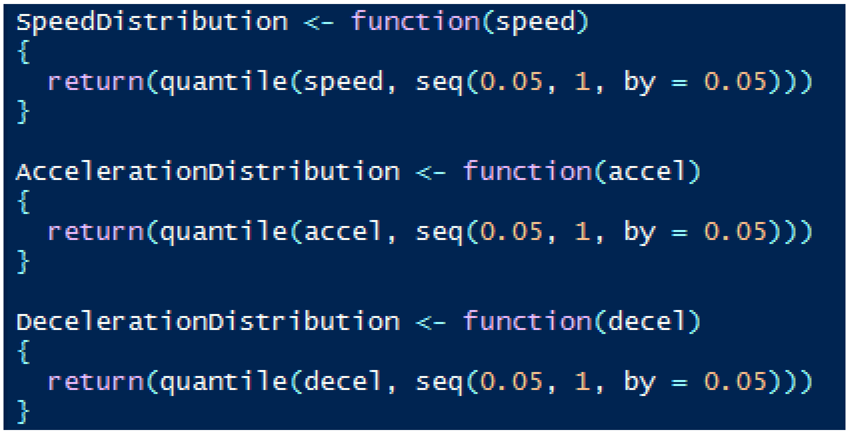
In this case acceleration has been split in the positive output from the resulting execution of *diff* stored in acceleration. On the other hand, when the resulting value is negative, this one is stored in deceleration.

The causes of splitting acceleration into positive and negative are merely practicability and because might be seen as two different features as long as a driver can accelerate brusquely but however decelerate smoothly by custom.

Distributions

In the R code proposed, it has been used *SpeedDistribution* and *accelerationDistbution* variables to assess both distribution with the only difference in the input (which will be speed and acceleration respectively).

R has an internal function *quantile* which produces sample quantiles corresponding to the given probabilities. Furthermore, in this is given example this function is called in a sequence from 0.05 to 0.95 like the way below:
In this way, among all possible values these are gather in those intervals where are more likely to be founded. Hence, for each possible value, this one is known the expected probability to find it in relationship the sample.



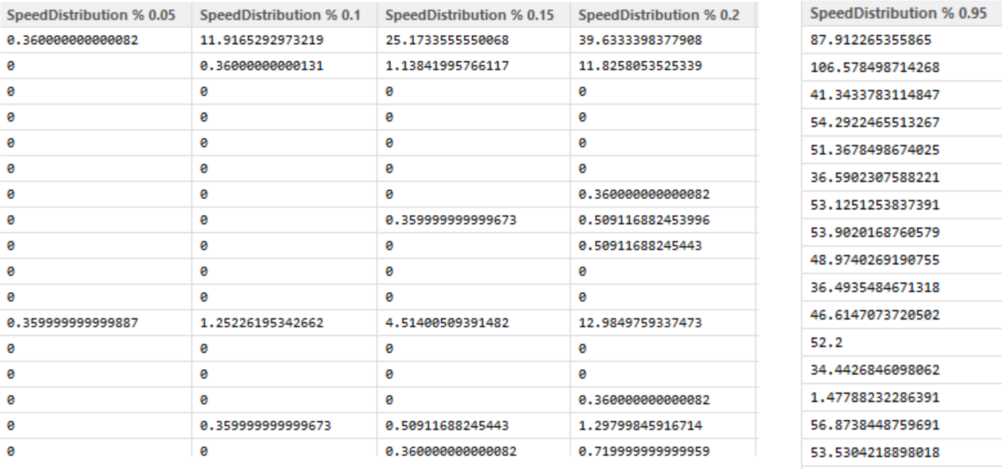
```
SpeedDistribution <- function(speed)
{
  return(quantile(speed, seq(0.05, 1, by = 0.05)))
}

AccelerationDistribution <- function(accel)
{
  return(quantile(accel, seq(0.05, 1, by = 0.05)))
}

DecelerationDistribution <- function(decel)
{
  return(quantile(decel, seq(0.05, 1, by = 0.05)))
}
```

What is important about *quantile* function is that provides a vision about how the features are distributed. For instance, can be observed if drivers go usually at high velocities, the main trends of accelerations and so many others driving style characteristics.

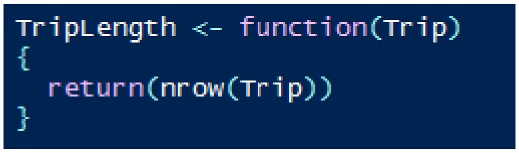
The outputs generated by these functions are columns with the *quantile* equalization for each probability. Thus, a column per probability assessed is added to the matrix. On the picture below we can observe different percentages of speed distribution.



SpeedDistribution % 0.05	SpeedDistribution % 0.1	SpeedDistribution % 0.15	SpeedDistribution % 0.2	SpeedDistribution % 0.95
0.36000000000082	11.9165292973219	25.1733555550068	39.6333398377908	87.912265355865
0	0.36000000000131	1.13841995766117	11.8258053525339	106.578498714268
0	0	0	0	41.3433783114847
0	0	0	0	54.2922465513267
0	0	0	0	51.3678498674025
0	0	0	0	36.5902307588221
0	0	0	0.36000000000082	53.1251253837391
0	0	0.359999999999673	0.509116882453996	53.9020168760579
0	0	0	0.5091168245443	48.9740269190755
0	0	0	0	36.4935484671318
0	0	0	0	46.6147073720502
0.359999999999887	1.25226195342662	4.51400509391482	12.9849759337473	52.2
0	0	0	0	34.4426846098062
0	0	0	0	1.47788232286391
0	0	0	0.36000000000082	56.8738448759691
0	0.359999999999673	0.5091168245443	1.29799845916714	53.5304218898018
0	0	0.36000000000082	0.719999999999959	

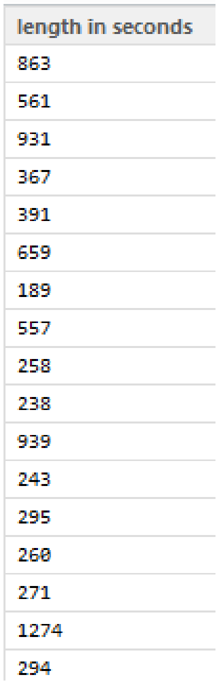
Path Duration

A function that helps to know the length for a given vector is length**().** Then according in which units (seconds or min) this figure will need to be multiplied. However, in this case is used a data.frame structure so instead of length will be used *nrows*, which returns the dimension for a given matrix or data.frame more suitable in this case because the nature of the variables are in data frame and do not new to be converted into different ones.



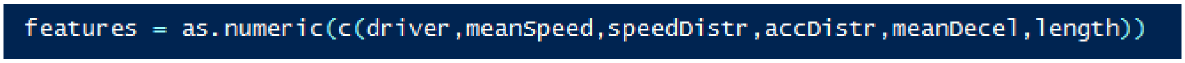
```
TripLength <- function(Trip)
{
  return(nrow(Trip))
}
```

The output of this function is a column with the length in seconds for each trip performed. This value is measured in seconds. Therefore if it is wanted to get the values in hour should be divided by 3600 seconds that constitute a whole hour.



length in seconds
863
561
931
367
391
659
189
557
258
238
939
243
295
260
271
1274
294

Once all data are gotten and save in a variable; in this case features:



```
features = as.numeric(c(driver,meanSpeed,speedDistr,accDistr,meanDecel,length))
```

The next step to develop is the data preparation.

Data transformation

Along this section, will be briefly discussed and carried out the proper data transformation and preparation of the data features extracted from the previous chapter to build up a predictive model able to identify each specifically driver style.

Data preprocessing converts raw data and signals into data representation suitable for application through a sequence of operations. The objectives of data preprocessing include size reduction of the input space, smoother relationships, data normalization, noise reduction, and feature extraction. Several data preprocessing algorithms, such as data values averaging, input space reduction, and data normalization, will be briefly discussed in this chapter.

This preprocess procedures are done by using *caret* package, which has within its functions; data cleaning, data splitting, testing/training data, model comparisons amongst others. These functions are not only fulfilled by *caret*. Other functions in RStudio and several scattered packages also do these functionalities. However, caret is a unified framework for applying different machine learning algorithms from different developers. That allow to use the same commands no matter for which methods is used, otherwise would be needed to learn the code type of each package individually.

The procedure followed to achieve a successful pre-processing is featured in the next figure:

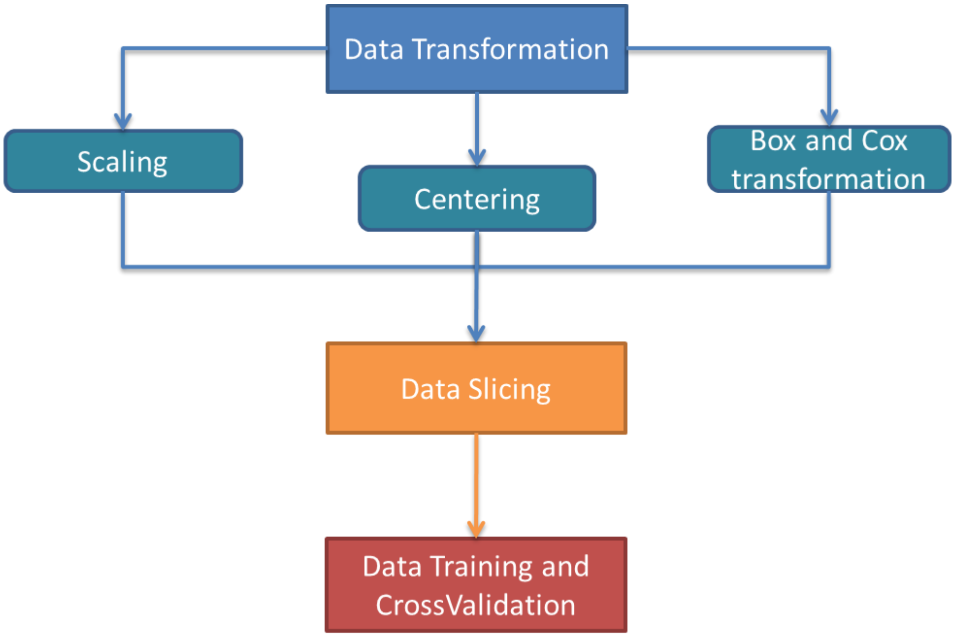
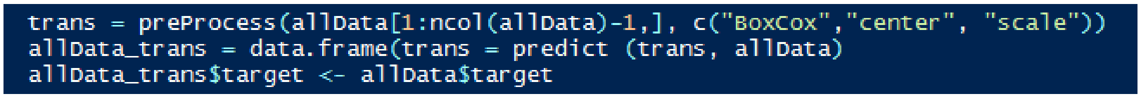


Figure 18. *Procedure followed to achieve a successful pre-processing in data*

With the caret package installed, preProcess function gather in a single function all procedures and step to achieve a successful data transformation:



```
trans = preProcess(allData[1:ncol(allData)-1,], c("BoxCox","center", "scale"))
allData_trans = data.frame(trans = predict (trans, allData)
allData_trans$target <- allData$target
```

Once data transformation methods, as scale, centering and Box & Cox transformation are applied data must be further suitable and ready for being used for modeling. At the beginning of this section was said that in data sets there is untrustworthy information regarding some trips which do not belong to the driver to whom is appointed but another a priori unknown driver from the sample.

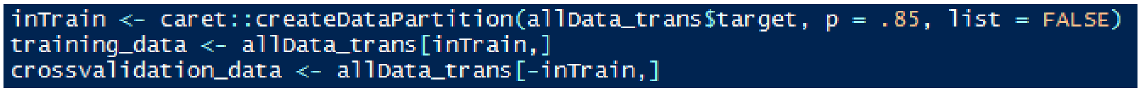
Predictive Modeling might help to identify those trips that are wrongly set up. To achieve this goal, transformed dataset must be divided into training data and cross-validation sets (or test set). Training data set is used to build and tune[31] the model, and test set is used to estimate the model's predictive performance usually by the hand of cross-validation.

[31] Make the algorithm learn from data in accordance to the predictive model set.

Cross-validation allows assessing how well the results of a statistical analysis will generalize to an independent new data set. Furthermore, Cross-Validation also is useful for overcoming the problem of over-fitting which it has been talked in 3.2.2.

Much like exploratory and confirmatory analysis should not be done on the same sample of data, fitting a model and then assessing how well that model performs on the same data should be avoided. When we speak of assessing how well a model performs or in other words how well the model predict new information.

Caret package has several functions that attempt to streamline the model building and evaluation process, as well as feature selection and other techniques. Amongst these functions specifically *createDataPartition* can be used to create randomized samples of data into training and tests sets.[16]



```
inTrain <- caret::createDataPartition(allData_trans$target, p = .85, list = FALSE)
training_data <- allData_trans[inTrain,]
crossvalidation_data <- allData_trans[-inTrain,]
```

In p is specified how this data is split up, in this case 85% and 15%, and list as FALSE avoids returning data as a list. The rate and the amount of data destined to training data or test is under criteria of modeler. Ideally, the model should be evaluated on samples that were not used to build or fine-tune the model, so that they provide an unbiased sense of model effectiveness. As said above, training data is the term to create the sheer model, while the test set or validation is used to analyze the performance. The larger is the data set evaluated the more quantity to gauge the performance.[32]

The process of splitting data in different data sets must be, under all circumstances, once all data have been transformed and normalized under the same parameters. Otherwise, in case would be carried out before transformation each data sets would bounded by different values according its data sets are constrained and the normalization and distribution in each case would not be comparable. Therefore, the transformation would have been made useless.

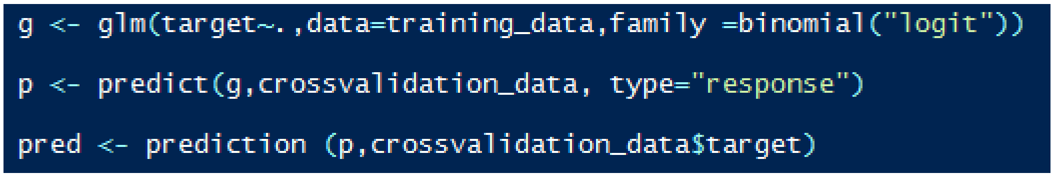
[32] Random sampling, dissimilarity sampling and other ways of splitting data has not been taking into consideration here.

The process of training data to make it learn

Every classifier evaluation using ROC method starts with creating a *prediction* object. This function is used to transform the input data (which can be in vector, matrix, data frame, or list form) into a standardized format.

Predict functions permits predicting values based on linear model object. These values are obtained by evaluating the regression function in the frame. Different arguments can be set up, for instance when se.fit argument is TRUE standards errors of the predictions are calculated. If Scale is set (with optimal *df* which calculates the degrees of freedom) standard deviation of the standards errors are computed, otherwise values are extracted from the model fit. Type can be used in order to choose between response and model term.[33]

If the fit is rank-deficient, some of the columns of the design matrix will have been dropped. Prediction from such a fit only makes sense if new data is contained in the same subspace as the original data.



```
g <- glm(target~.,data=training_data,family =binomial("logit"))

p <- predict(g,crossvalidation_data, type="response")

pred <- prediction (p,crossvalidation_data$target)
```

As can be noticed, predict function, which calculates the predicted probability based on the generalized linear model, is applied within the cross validation data set not in the whole data samples. This is just done in this way because the purpose is to know how well the performance of the model is. And, as it was specified in some chapters before, cross validation data samples are created specifically to chase this commitment.

Creation of a logistic regression model in R environment

The basic function fitting generalized linear models is the function glm, which its structure is the next one:

glm (formula, family, data, weights, subset,…)

Formula: An object of class “formula” or a class capable of been coerced to.
The ~ operator is basic in the formation of such models. An expression of the form `y ~ model` is interpreted as a specification that the response `y` is modeled by a linear predictor specified symbolically by `model`.

Family: Description of the error distribution and link function chosen to be used in the predictive model.

Data: Data frame or list or any object which is feasible to be coerced by *as.data.frame*, containing the variables in the model.

Weights: Often is used an optional vector with a "prior weights" to be equalized in the fitting process in those cases in which information about data is known beforehand. However in these cases it was not performed.

Ordinarily, the relationship or the reason for using glm within the logistic regression may be understood as though the glm would be a form or a tool which permits distinguishing every single sample according the distribution that follows the whole data set and fit the function a logistic function according the characteristic of data.

As long as logistic regression is appropriate when the outcome variable is categorical with two possible outcomes (i.e., binary outcomes), binary variables can be represented using an indicator variableY_i, taking on values 0 or 1. Hence, applying glm the indicator variable can be classified and, in the log it function case, predict how this classification accuracy is.[40]

At this point achieved, data training is supposed to be trained, but before applying the whole model to the main case study first must be known how well training data predicts its outcomes and functions.

For what is then applied the Receiver Operating Characteristics graphs. Below is explained its theory basis and the application in the algorithm:

Receiver Operating Characteristics (ROC)

Receiver operating characteristics (ROC), or ROC curve, is a graphical plot that allows illustrating and visualizing the performance of a binary classifier system as long as its discriminations threshold is varied. A binary classifier can be understood as an output with two different possibilities. In the concerning case, for instance evaluating if every single path or route committed by each driver really corresponds to the driver assessed or otherwise another one.

The curve is created by plotting the true positive rate against the false positive rate at various threshold setting. Concurrently ROC curve is used sensitivity and specificity.

The idea in the background of ROC curve is to judge how well the validation sets adjust to the real or true sample. Since Logistic Regression is a classification method that not only can make a prediction but also a distribution probability of the admission, comparing these values with the true ones is possible to determine how these predicted probabilities are performing. To get a main idea about what ROC curve is, take a look to the following graph.

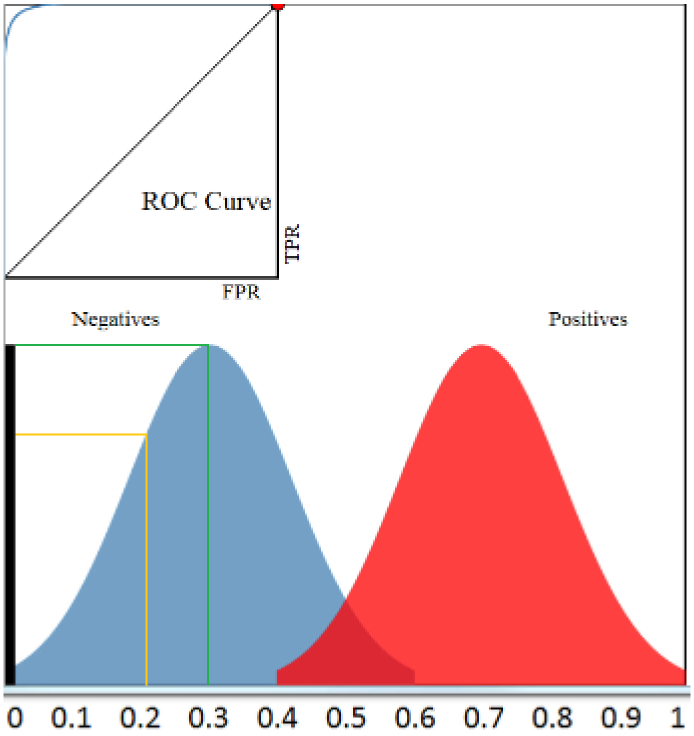


Figure 19. *ROC Curve generated from two classifier distributions*

In the graph showed, may be observed two distributions that correspond to the admission or acceptance status and the negative acceptance output from validation set. This example chosen is not the most representative of the real problem tackled, the amount of data in each distribution are equal (fact that is not regular in reliable samples) . So it means that if within positive classifier there are X samples, in the negative classification will be X as well. Besides, in the real-world problems the predicted probabilities are unlikely to be “smooth” distribution approaching the normal as actually it happens in this example. Even though, this simplified example helps to understand the background of RUC curve.

The x-axis represents the predicted probabilities equalized in the model predicted. On the other hand y-axis represents the amount of observations. So for example, in the picture illustrated, there is an amount Z for which the model predicted an admission probability of 0.2. However, in probability 0.3, the values predicted are in a larger amount of sets than in 0.2.

Based on this imaginative plot illustrated, if the threshold would be situated in 0.5, classifying everything above 0.5 as admitted, and appositively as not admitted, the accuracy of model would be 90% which is a considerable satisfying figure.

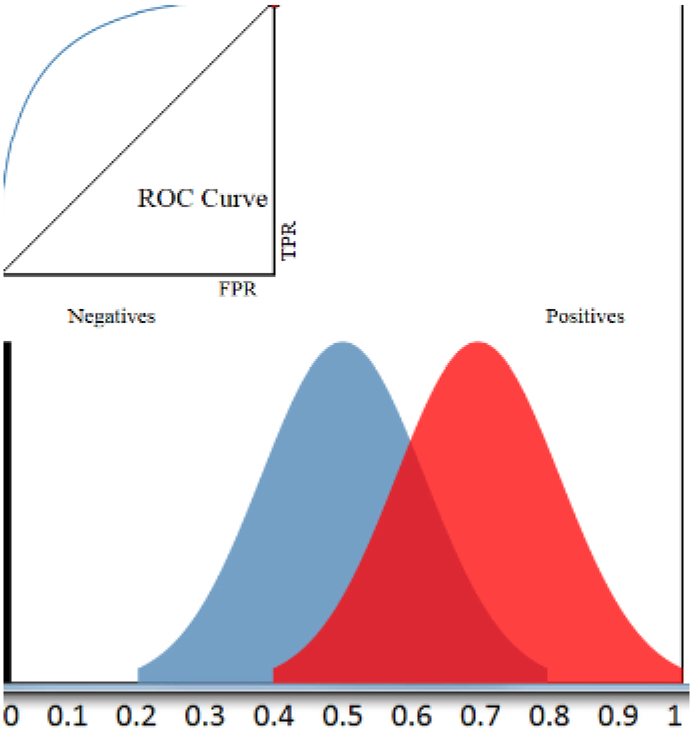


Figure 20. *Distributions drawn overlapped*

Pretending to spoil a little the classifier causing an overlapping on both distributions, now may be noticed that regardless where the threshold is situated, the accuracy of the predictive model is less than before.

In the upper left is represented what ROC curve is. As said before, ROC curve is a plot of the True positive versus the false Positive rate for every classification threshold. Thus, as long as the threshold is moved back and forward, the point in the ROC curve will be displaced as well.
True Positive measures the proportion of actual positives which are correctly identified as such in some field is known as sensitivity.

True Negative however, measures the proportion of negatives which are correctly identified as such. It’s known as well as specificity.

The trade-off between both measures is the result of ROC curve. Nevertheless, ROC curve also response questions as: When the actual classification is positive how often does the classifier predict positive? Or in other words, when the classifier predicts something admitted (positive) but is not, this fact is called false positive. Otherwise, when the classifier incorrectly predicts negative, so is incorrectly rejected, is called false negative.

Assuming positive being admitted and negative rejected, the next table is obtained:

True positive	Correctly admitted
False positive	Incorrectly admitted
True negative	Correctly rejected
False negative	Incorrectly rejected

In order to understand how ROC curve is represented and how is drawn, figure out this graph hold 500 samples which do not correspond to the set, and as well 500 samples that correspond. Thus, summarizing, one thousand samples to classify.

Setting the threshold of 0.8 would classify 100 samples as admitted, and 900 as rejected. The true Positive rate would be then 100 divided by 500[33], resulting 0.2. Moreover, the false positive rate would be the total amount of samples remaining in the right side divided by the all samples. In this case would be 0. Therefore, to represent the ROC curve, would be plotted a point at 0 on the x-axis, and 0.2 on the y-axis. Proceeding with the same logic for all the point from 0 to 1, would be described the ROC curve, which can be noticed is a trade-off between sensitivity and specificity.

Sensitivity can be defined as: $Sensitivity\ = \frac{true\ positive}{true\ positive+false\ negative}$

Whereas specificity is: $Specificity = \frac{true\ negative}{true\ negative+false\ positive}$

Using ROC curve represents a huge benefit evaluating all possible classification thresholds, whereas misclassification rate only represents the error rate for a single threshold.

The better performance of the classifier, the RC curve would tend to be more skewed to the upper left corner of the plot. Conversely, the worst performance in the output turned out by the classification model, (that's normally when the distributions of both TPR and FPR are overlapped[34]) the closer is the plot described by the ROC curve to linearity.

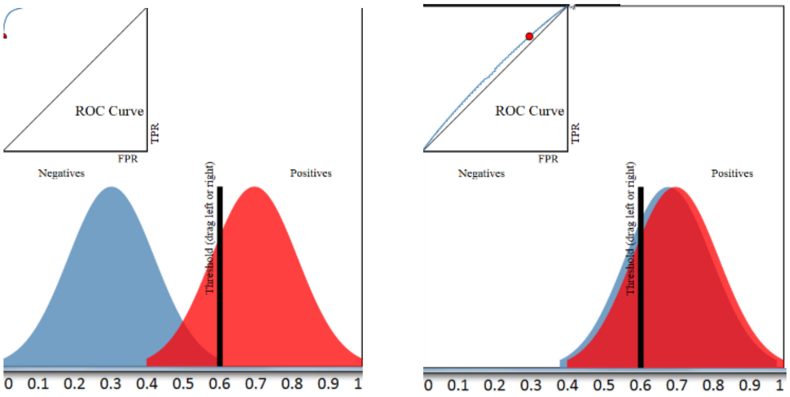


Figure 21. *ROC Curves showing how the performance of it changes when distributions overlap*

[33] This operation is easily equalized considering the amount of samples admitted by this specifically threshold setting, and divided by the true known simple which corresponds to positive, in this example 500.

[34] The more overlap between both distributions, the worthless is the model prediction. Conversely, the more overlap, the closer to linear function is the ROC curve described.

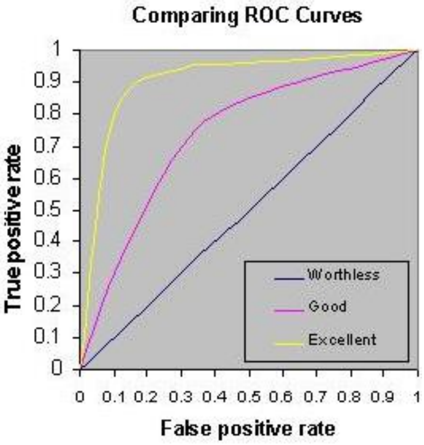


Figure 22. *Comparison between Rock cubes performance*

AUC Analysis

Once ROC Curve is represented in all its range, so a two-dimensional depiction of classifier performance is plotted, in order to compare classifiers is desirable to reduce ROC performance into a single scalar value representing expected performance. A common method is carry out by calculating the area under the ROC curve, abbreviated as AUC.

The area under the curve, measures the ability of the test to correctly classify the subsets into its corresponding group. In the concerning case, for instance if the path 56 really corresponds to the driver assigned. AUC allows to randomly pick different subsets, for instance one track which corresponds to the driver chosen and another which do not correspond. The area under the curve attempts to illustrate the percentage of random drawn pairs for which the prediction is correctly fulfilled (that is how well the test classifies the two options in the random pair).

In conclusion, the purpose or aim of AUC is to equalize the amount area under the curve. As long as this curve is measured in percentages, the outcome of AUC is equally the percentage of the performance. TPR and FPR are either measured from 0 to 1. Since ROC Curve is a trade-off between TPR and FPR, the lower area under the curve ever equalized could be 0.5, figure which shows a poor performance of predictive model. However, even not being possible to achieve it, a total amount of area reckoned equal to 1 would come to mean a really good prediction.

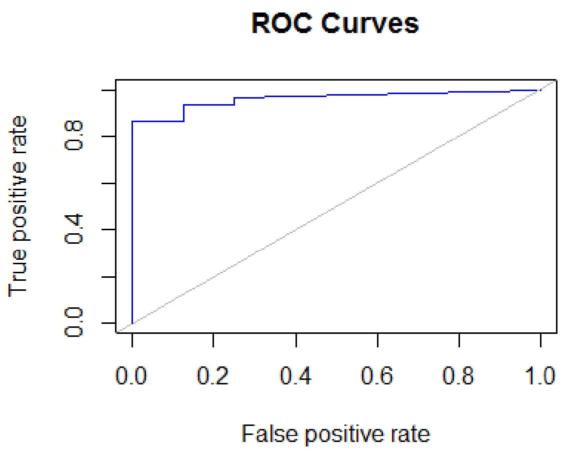


Figure 23 *Different ROC Curves (trip 3 driver 1)*

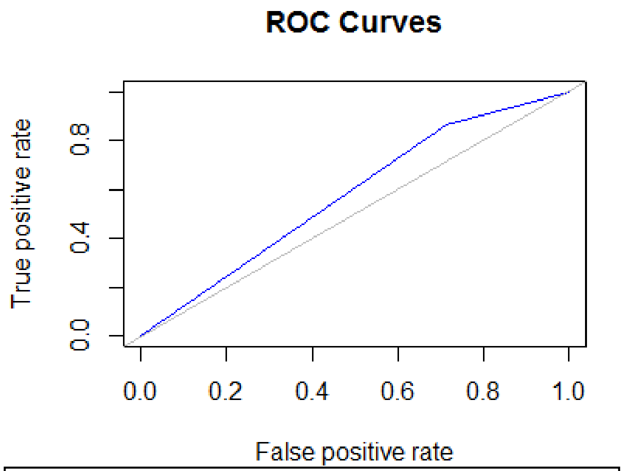


Figure 24. *Different ROC Curves (trip 4 driver 10)*

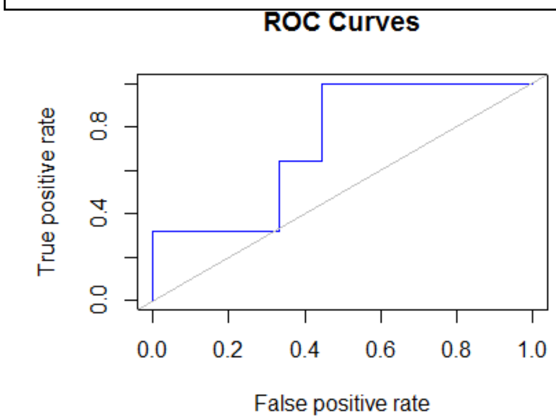


Figure 25. *Different ROC Curves (trip 5 driver 2)*

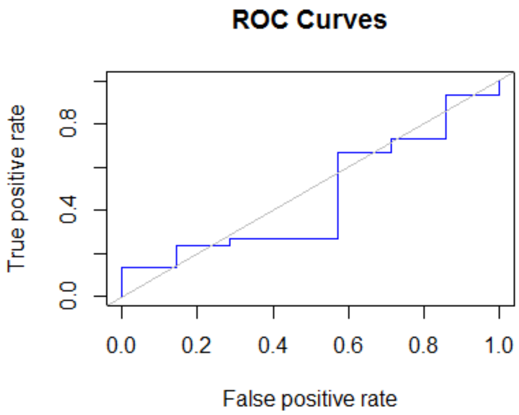


Figure 26*. Different ROC Curves (trip 6 driver 3)*

Despite the common way to visualize the ROC curve, in the line which denotes the trade-off between the TPR and FPR, also a common way to represent ROC curve is an approximation with log function.

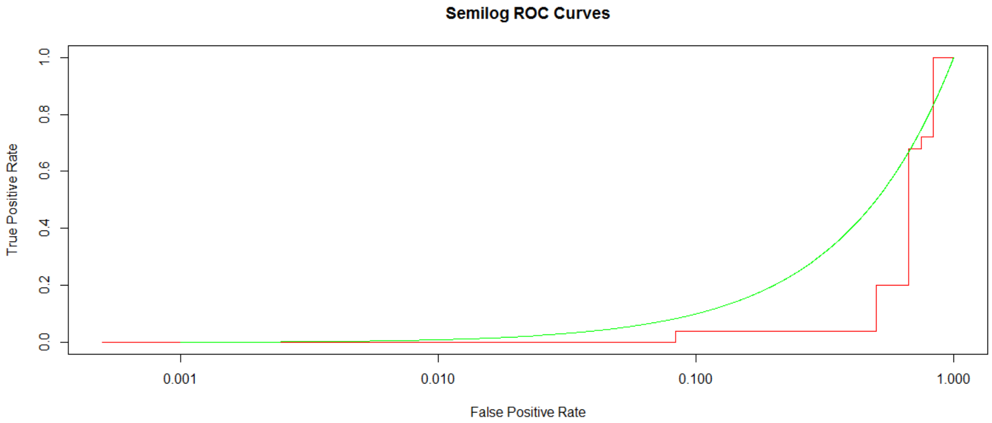


Figure 27. *Example 1 of ROC Curves approximating log*

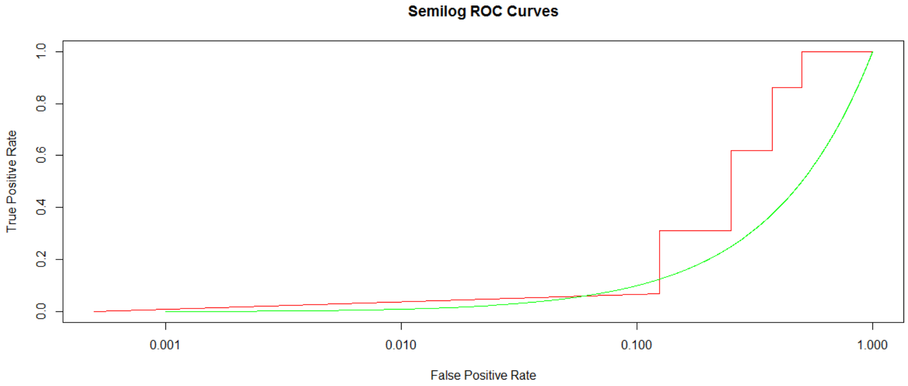
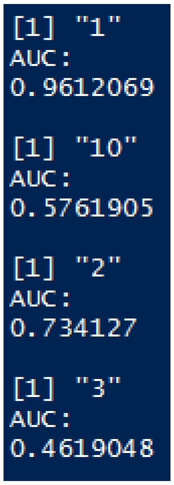


Figure 28*Example 2 of ROC Curves approximating log*

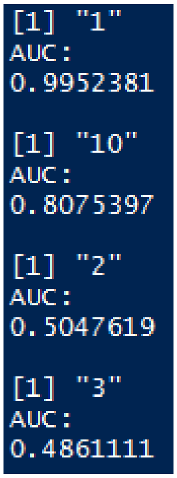


```
[1] "1"
AUC:
0.9612069

[1] "10"
AUC:
0.5761905

[1] "2"
AUC:
0.734127

[1] "3"
AUC:
0.4619048
```



```
[1] "1"
AUC:
0.9952381

[1] "10"
AUC:
0.8075397

[1] "2"
AUC:
0.5047619

[1] "3"
AUC:
0.4861111
```

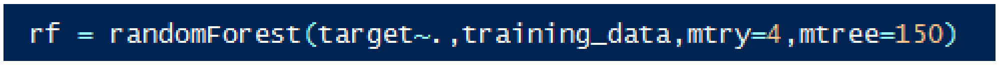
4.2 Case study 2: Recognition of driving styles with random Forest method

Once logistic is entirely performed and assessed its classification accuracy, another model can be applied in order to compare which fits better data and gets a better result.

As models are similar, they only change in the predictive training data, models are so similar only changing in the predict function. In this case instead of using a glm function that approach data, the data partition with decision trees is applied.

The prediction in this case is done through the ensemble model included in the package randomForest.

This package include a function described below that permits create random trees with the optimum selection of predictive variables

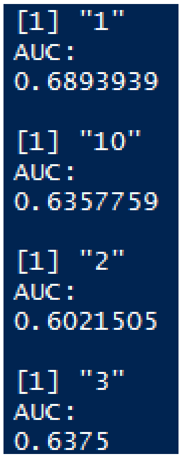


```
rf = randomForest(target~.,training_data,mtry=4,mtree=150)
```

In this function *mtry* refers to the maximum number of split in each leaf and $mtree$ the random forest in each iteration. The higher this value, the greater will be the depth and therefore the computation time of the algorithm. However, applying recursively higher values but realistic taking into account the time cost expense the output do not show a reasonable change in its performance.

As did with logistic regression this predictive model has to be applied in training data and validate through cross-validation.

The AUC values using $randomForest$ are the next one:



```
[1] "1"
AUC:
0.6893939

[1] "10"
AUC:
0.6357759

[1] "2"
AUC:
0.6021505

[1] "3"
AUC:
0.6375
```

That shows both methods are not that much far away in accuracy. However, in the computational cost they are quite different. Logistic saves time in comparison with randomForest.

Nevertheless, this does not mean logistic is better than randomForest. In this example perhaps it might be seen as it is, but in any other models maybe not. Models perform better or worst depending in the nature of the data and the function target, for that reason in order to get the most fancy accuracy different methods need to be approached.

Furthermore, notice logistic regression in driver 3 do not get a good result. In this case, decision trees fit better this variable. For that reason, in real models is not applied one chosen approach but different are. In that way, can be construct a model that is able to get the potentials of several of them, choosing those that are higher. In this case, the average of AUC for 4 drivers is: 0,6212. However, in logistic regression was 0,6814. Here in this small examples data comes out to be higher in logistic, but in reality are those so close randomForest performing slightly better.

But the valuable point here is that both models perform, for the different drivers, different outcome values. Hence, the best model it is sough one might choose the best result in both models. Applying ensemble models permit choosing those drivers, or classification for what each method perform better. In the example from above the AUC mixing randomForest and logistic regression would be such: 0,7467.

What is, as might be noticed a considerable improvement in the accuracy rate.

4.2 Case study 2: Recognition of driving styles for a realistic data set (Kaggle)

Kaggle is a website community of data scientists. This website publishes different business problems from some of the world's biggest companies that are outsourcing their data science tasks.

In that way, Kaggle provides cutting-edge data science results to companies from very different fields, starting from science and information technologies to retail, energy, and financial services.

Scientists and all the users forming platform come from quantitative fields such as computer science, statistics, econometrics, maths, etc… However, they compete to get the best solution and therefore win the prize money and data, besides meet, learn and collaborate with experts from related fields.

Likewise, Kaggle is a platform for hosting public data science challenges and competitions in which sponsors post their troubles and problems and then data scientists from all over the world compete to create the best solution.

The AXA insurance company provided, in Monday 15 of December 2014, a dataset of 547,200 anonymized drivers' trips of 2736 different drivers, each driver having 200 trips. The competition's goal was to develop an algorithmic signature of driving style in which with the use of it some questions whether driver drive for a long trip without stopping, if this one take highway trips or back roads as well as the attitude and behavior according whether they take turns at high speed or accelerate sharp or smooth.

All this features must be extracted from the data GPS data provided across telematics. Telematics might represent a big step in the evaluation insurance risk for a particular driver since not only the ability to pay or just the driver features (age, gender, accident history,…) can be taken into account but with the use of telematics a new horizon it comes out.

Thus, for this competition Kaggle participants through the raw data provided from GPS, which is the location (position coordinates) for a given time, had to come up with a fingerprint capable of distinguish what trips were driven by a driver, or in other words, taking 1 random trip from the sample, predict who was the driver it performed.

In the case study assessed, for each driver, there is an assumption that some drivers contain some trips that were not driven for that particular driver but belongs to any other a priori unknown. Since data had been mixed, all those trips wrongly classified into files that do not correspond.

In other words, there are some trips that do not belong to a particular driver who is specified but any other else, not knowing which these trips are neither to whom it belongs. Because of that, it is needed to construct a model that permits, knowing quantitatively, what is the probability for every single path to really fit in the specified file.

Applying the algorithm for a Big data set

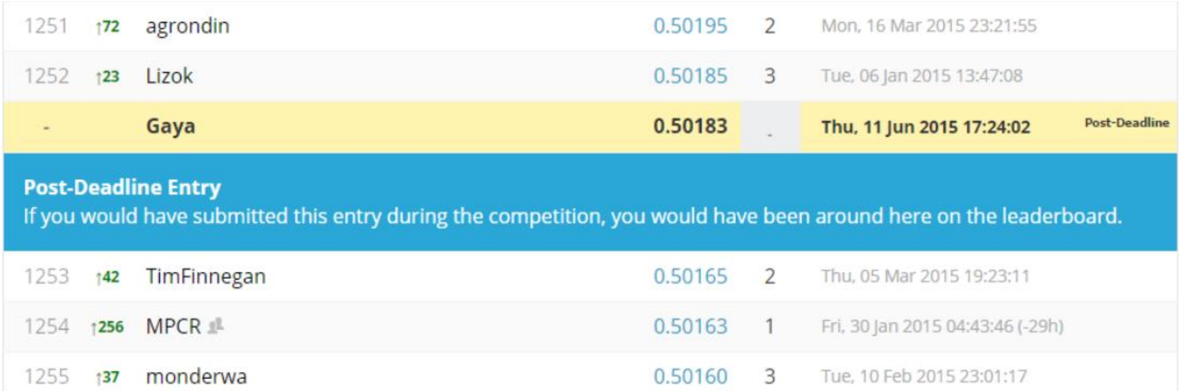
Due to the trunk model, or the basis model, works with non-dependence of the amount of data provided for the model[35],the only difference with respect to Case study 1 is the data. This time, instead of using 4 drivers, we have around 3000 of drivers that result in much higher computational costs.

Moreover, ROC Curves are not applied as they are used to confirm the truthfulness of the model and determine the specify and sensibility of it. However, once this is done, there is no need to probe it again. So ROC curves are removed from the code.

As Logistic regression is not a complex model, and the features do not require high time expenses to reckon them, the total time to compute the whole data sample with the following computing performance; Intel® Core™ i7-4500U CPU 1.80GHZ 2.40Hz and a usable memory of 7,89GB, is over 1 hour and half.

Although the competition was already finished when the code's output was submitted, Kaggle allows testing how your result would be in case you would have participated.

The result with the logistic regression approach is illustrated in the following screenshot:



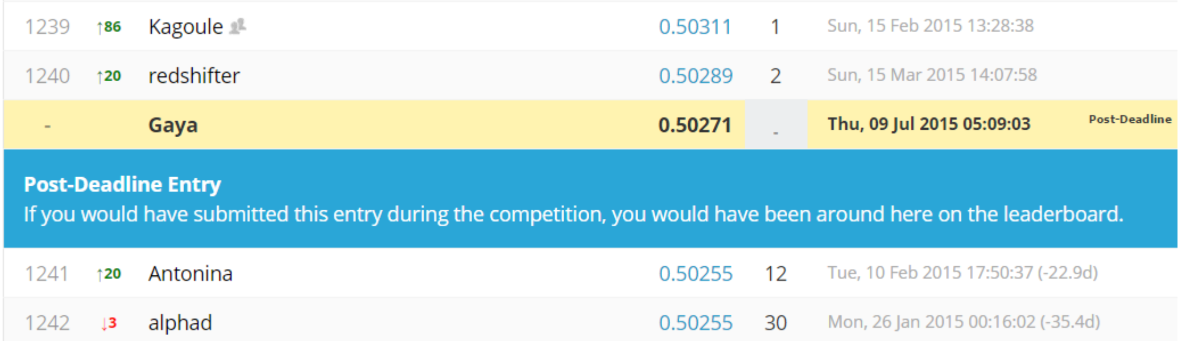
1251	↑72	agrondin	0.50195	2	Mon, 16 Mar 2015 23:21:55	
1252	↑23	Lizok	0.50185	3	Tue, 06 Jan 2015 13:47:08	
-		**Gaya**	**0.50183**	-	**Thu, 11 Jun 2015 17:24:02**	**Post-Deadline**

Post-Deadline Entry
If you would have submitted this entry during the competition, you would have been around here on the leaderboard.

1253	↑42	TimFinnegan	0.50165	2	Thu, 05 Mar 2015 19:23:11
1254	↑256	MPCR	0.50163	1	Fri, 30 Jan 2015 04:43:46 (-29h)
1255	↑37	monderwa	0.50160	3	Tue, 10 Feb 2015 23:01:17

[35] The performance and the outcome accuracy are not supposed to be the same, but the functions and procedures are exactly the same.

On the other hand, the result with RandomForest approach is, as shown below, is slightly accurate.



1239	↑86	Kagoule	0.50311	1	Sun, 15 Feb 2015 13:28:38	
1240	↑20	redshifter	0.50289	2	Sun, 15 Mar 2015 14:07:58	
-		**Gaya**	**0.50271**	-	**Thu, 09 Jul 2015 05:09:03**	**Post-Deadline**

Post-Deadline Entry
If you would have submitted this entry during the competition, you would have been around here on the leaderboard.

1241	↑20	Antonina	0.50255	12	Tue, 10 Feb 2015 17:50:37 (-22.9d)
1242	↓3	alphad	0.50255	30	Mon, 26 Jan 2015 00:16:02 (-35.4d)

However, although the result is slightly better, the time consumption for this approach was around 5 hours, what is 3 times more than with logistic regression.

Now one should evaluate how much valuable is this improvement and if really it is necessary to lose the difference in time between both performances to get a gain as in this case.

Also this probe that in large degree the well-performance depend on the description parameters with which the model is based i.e. the engineering features that the model is hold on not that much in the model approach.

Conclusions

Throughout the project's development, the applicability of telematics' data in airports settings has been exhaustively analyzed. Telematics' technology was presented as an emerging technology aimed at improving decision making. For that reason, and to achieve this goal, telematics combine several technologies working together in order to add value in the decision making process; since those based on hardware and physical deployment, till the processing of data to extract the knowledge.

It was demonstrated in the thesis that data mining and machine learning methods could contribute to improve the data processing and automate the "learning" from data in order to give a boost in the performance and sweep away the doubts for purchasing this immature technology. Machine learning methods might help to manage big data from telematics sources and get knowledge of it through methods as performed in this project.

Telematics' leverage is based on the ability to get features that would be impossible to get with ordinary technologies. However, the more features and its refreshing values are in the model, the more complexity is added to its processing and management. Concurrently, machine learning methods permit to treat big data in order to, not just manage it, but rather to improve the knowledge and take profit of it.

Aeronautics' field, due to the nature which has to deal with, i.e. aircraft perform its natural operations isolated across the sky, is a forwarding and pioneer in the use of telecommunications and informatics. Big amounts of data are created and transmitted, especially in the air-side.

However, in ground operations all these data transmitted are not used, neither created with the purpose of increasing the efficiency of ground systems. Through data telematics for instance applied in bus companies working in airports, it would allow assess features as driving behavior, fuel consumption and other features across machine methods that and extract new knowledge with which study better circumstances in the service provided.

In the concerning case of bus services in the airport zone, would be interesting to assess a model according to the resources available and the expected desirable service offered in order to get the optimum performance of the system. Somehow it is pretended to set the parameters for which the system is optimized. And later, with the use of machine learning, find those patterns in the model that are far away or can be drawn to the optimum.

In that way, driver behavior such as; acceleration profile, fuel consumption, eco-friendly habits, etc. could be assessed with the purpose of attempting to change it. Likewise fuel consumption and fuel emissions would be controlled and reduced, turning out in a more desirable circumstance for the business which might be useful to improve the company image and reduce fuel consumption and exhaust emissions.

With the purpose of attempting to find classification patterns the study cases was performed. These practical experiments have been conducted using real telematics data . Two models have been developed such as logistic regression and random forest, including previous pre-processing of data and engineering of features. These models have been applied to a "toy" case (Case Study 1) and 2) with a small dataset of 4 drivers, as well as to a real case (Case Study 3) using **AXA data** (a dataset of **547,200 anonymized drivers' trips** of 2736 different drivers, each driver having 200 trips).

In the Case Study 1, the logistic regression model trained on a dataset of 4 drivers provided was analyzed using a ROC Curve. The AUC equal to 0.6645 has been obtained. In the random forest ensemble model, the ROC Curves showed a higher tendency (AUC equal to 0.6345) although requiring a higher time for computations. Hence, that demonstrates that the model created has to take into account the necessary time to successfully process all data in accordance with the processors throughput available and the programming structure i.e. sequence computing, parallel computing, on-demand computing, because these two "variables" will determine the scope and restrictions of the model.

Once the models have been tested in Case Study 1, the next step was to validate them on a real case using a big amount of data. The efficiency of the logistic regression model has been evaluated by uploading the algorithm outputs to **the Kaggle platform**. As a result, the AUC equal to 0.50183 has been obtained by running the algorithm during more than 1 hour. This result has been quite expected due to the simplicity of the logistic regression approximation. The random forest applied to the same data gave the AUC equal to 0.50271, however at a higher computational cost yielding 5 hours of time computing against the 1 and half of logistics regression.

It can be concluded that the application of machine learning methods to real cases involving big amounts of data requires **the usage of parallel computing and cloud computing technologies**. It must be highlighted that, though we have developed complex features such as centripetal acceleration or number of intersections and loops in a trip, the calculation of these features has very high computational cost. Therefore it was practically unreasonable to integrate them into the models without using parallel computing and cloud computing.

Though the model can be further improved applying ensemble models and better engineering features, this one has the advantage of, even based on GPS telematics data for automobile vehicles, can be easily extrapolated and perfectly fit in other application fields such airports ground services or any other telematics vehicle assessment.

Nevertheless, as each field is subject to its own features, restrictions and conditions, this should be adapted in the obtaining of engineering features procedures in order to chase those features that better fit and explain the background of the model.

BIBLIOGRAPHY

1. **Advanced Driver Assistance Systems**. *Mobile Information Society. European Comission, 2000.* ftp://ftp.cordis.europa.eu/pub/telematics/docs/tap_transport/adas.pdf

2. **Focus on Transport and Logistics**. Web *designed by Web Workshop* www.focusontransport.co.za/regulars/best-practice/389-swiss-precision-thanks-to-mix-telematics.html

3. **Telematics Functions for Transport and Logistics.** *Idem telematics, by BWP group.* http://www.idemtelematics.com/en/functions/trailer.html

4. **Telematics for the airport environment barriers to success.** http://www.sts-technology.com/docs/Pinnacle-Air-GSE-&-Telematics.pdf

5. **Swiss precision thanks to MiX telematics**. Focus on transport and logistics http://www.focusontransport.co.za/regulars/best-practice/389-swiss-precision-thanks-to-mix-telematics.html

6. López, E. **La telemática en el transporte aéreo**. 1996 http://dialnet.unirioja.es/descarga/articulo/2781276.pdf

7. **MathWorks.** *Filtering and Smoothing Data. 1994-2015, The MathWorks Inc.* http://es.mathworks.com/help/curvefit/smoothing-data.html

8. W.Bowman, A & Azzalini, A. **Applied Smoothing Techniques for Data Analysis: The Kernel Approach with S-Plus Ilustrations.** OUP Oxford, 1997. https://books.google.es/books?id=7WBMrZ9umRYC&printsec=frontcover&dq=data+smoothing&hl=es&sa=X&ei=9tsFVYeeKIP_UqOugrgC&ved=0CCAQ6AEwAA#v=onepage&q=data%20smoothing&f=false

9. Wood , G. **Data smoothing and differentiation procedures in biomechanics**. University of Western Australia, 1891-1894 http://wweb.uta.edu/faculty/ricard/Classes/KINE5350/Wood%20(1982)%20Data%20smoothing.pdf

10. ***RPubs.*** Machine learning 2, by Ryan Kelly. https://rpubs.com/ryankelly/ml2

11. Kuhn, M. **Predictive Modeling with R and the caret Package.**

http://www.edii.uclm.es/~useR-2013/Tutorials/kuhn/user_caret_2up.pdf

12. **Data Preprocessing.** *Chapter 15, 2001 by CRC Press LLC*
    http://sedok.narod.ru/s_files/poland/2360_PDF_C15.pdf
13. **R Tutorial Series.** *By John M Quick.*
    http://rtutorialseries.blogspot.com.es/2012/03/r-tutorial-series-centering-variables.html

14. **Quantile explanations**
    https://www.stat.auckland.ac.nz/~ihaka/787/lectures-quantiles-handouts.pdf

15. **Statistica Help**. *By StatSoft Inc*
    http://documentation.statsoft.com/STATISTICAHelp.aspx?path=Spreadsheets/Spreadsheet/UsingSpreadsheets/BoxCoxTransformations/BoxCoxTransformationOverviewandTechnicalNotes

16. **Data Splitting.** *Created on 15 2015 using caret version 6.0-47 and R Under development.*
    http://topepo.github.io/caret/splitting.html

17. **An Analysis of Transformations.** *Journal of the Royal Statistical Society. Vol.26, No.2. (1964).*
    http://fisher.osu.edu/~schroeder.9/AMIS900/Box1964.pdf

18. *Gutierrez-Osuna,R.* **Lecture 13: ValidationWright.** *Intelligent Sensor Systems. State University*
    http://research.cs.tamu.edu/prism/lectures/iss/iss_l13.pdf

19. **Wolf, A. Cross-validation for detecting and preventing overfitting.** *School of Computer Science, Carnegie Mellon University.*
    http://www.autonlab.org/tutorials/overfit10.pdf

20. *Arlot, S & Celisse, A.* **A survey of cross-validation procedures for model selection.** *Statistics Surveys,Vol. 4 (2010) 40–79.*
    http://www.di.ens.fr/willow/pdfs/2010_Arlot_Celisse_SS.pdf

21. Domingos, P**. Machine Learning**. *University of Washington Pag 52-56*
    https://class.coursera.org/machlearning-001/lecture/161
22. Mitchel M., T. **Decision Tree Learning.** *Lecture slides for textbook Machine Learning,McGrawHil,1997*
    http://www.cs.cmu.edu/afs/cs/project/theo-20/www/mlbook/ch3.pdf

23. **The Comprehensive R Archive Network**
    http://cran.es.r-project.org/

24. Kuhn, M. **A Short Introduction to the caret Package.** *2015. Pag 58-59* http://cran.r-project.org/web/packages/caret/vignettes/caret.pdf

25. **Euroean Space Agency**. *From 1994 to 2014.* Pag 67-68 http://www.esa.int/ESA

26. **GoogleCloudPlatform/appengine-mapreduce.** *GitHub, MapReduce1.* https://github.com/GoogleCloudPlatform/appengine-mapreduce/wiki/1-Mapreduce

27. Dean,J & Ghemawat, S. **MapReduce: Simplified Data Processing on Large Clusters.** http://static.googleusercontent.com/media/research.google.com/es//archive/mapreduce-osdi04.pdf

28. **The Evolution of the Microprocessor.** *By Intel a trademark of Intel Corporation.* http://download.intel.com/newsroom/kits/40thanniversary/pdfs/40_anniversary_evolution_FV.pdf

29. Pérez, E. **Omicrono.** *Rompiendo las barreras del diseño de procesadores: ¿Está la ley de Moore obsoleta? 2013.* http://www.omicrono.com/2013/04/rompiendo-las-barreras-del-diseno-de-procesadores-esta-la-ley-de-moore-obsoleta/

30. **Data Warehouse.** *Hadoop Big Data Greenplum MPP* https://dwarehouse.wordpress.com/2012/12/28/introduction-to-massively-parallel-processing-mpp-database/

31. *Bader, A.D & Pennington,R.* **Applications.** *The International Journal of High Performance Computing Applications, Volume 15, No. 2, Summer 2001, pp. 181-185.* http://www.cc.gatech.edu/~bader/papers/ijhpca.pdf

32. **TechNet.** *Evaluating the Benefits of Clustering* https://technet.microsoft.com/en-us/library/cc778629(v=ws.10).aspx

33. Tarca l, A. et al. **Machine Learning and Its Applications to Biology.** *Published: June 29, 2007DOI:10.1371/journal.pcbi.0030116.* http://journals.plos.org/ploscompbiol/article?id=10.1371/journal.pcbi.0030116#s4

34. **Telematics.com**. The home of telematics technology. http://www.telematics.com/

35. **TechTarget**, 2000-2015. http://searchnetworking.techtarget.com/definition/telematics

36. **Graphics Layout Engine**
http://glx.sourceforge.net/examples/2dplots/entropy.html

37. **Quick R**. *By Robert I. Kabacoff, Ph.D*
http://www.statmethods.net/advgraphs/layout.html

38. Shalizi, C. **Advanced Data Analysis from an Elementary point of View**. Chapter 12, May,2015. Pag 47 - 51

39. H. Witten, I & Frank, E. **Data Mining: Practical machine learning Tools and Techniques.** June 2005. Pag 1-2,

40. Kerns J., G. **Introduction to Probability and Statistics using R**. March, 2011.

41. Chapman & Hall. **Machine Learning: An algorithm approach**. October 8 , 2014 Pag 23 -26

42. Benjamin D. Gentle Introduction to Random Forests, Ensembles and Performance Metrics in a Commercial system. November 2012
https://citizennet.com/blog/2012/11/10/random-forests-ensembles-and-performance-metrics/

ANNEX

ANNEX 1: R Code

```
library(caret)
library(ROCR)

SpeedReckon <- function(trip)
{
   speed=3.6*sqrt(diff(trip$x,1,1)^2+diff(trip$y,1,1)^2)   return(speed)
}

meanSpeedPerTrip <- function(Speed)
{
  return(mean(Speed))
}

Acceleration <- function(speed)
{
  acceleration = diff(speed)[diff(speed)>0]
  if (length(acceleration) == 0)
    return(0)
  else
    return (acceleration)
}

Deceleration <- function(speed){

  deceleration = diff(speed)[diff(speed<0)]
  if (length(deceleration) == 0)
    return(0)
  else
    return (deceleration)
}

meanDeceleration <- function(decel)
{
  return(mean(decel))
}

SpeedDistribution <- function(speed)
{
  return(quantile(speed, seq(0.05, 1, by = 0.05)))
}

AccelerationDistribution <- function(accel)
{
  return(quantile(accel, seq(0.05, 1, by = 0.05)))
```

```
}

DecelerationDistribution <- function(decel)
{
  return(quantile(decel, seq(0.05, 1, by = 0.05)))
}

TripLength <- function(Trip)
{
  return(nrow(Trip))
}

NumbStops <- function(speed)
{
  stops <- with(rle(speed<10), sum(values & lengths>30) )
  return stops
}

Nloops <- function(trip)
{
  u1 = lapply(trip$x,function(x)diff(trip$x))
  u2 = lapply(trip$y,function(x)diff(trip$y))

  u3 <- list(u1,u2)
  as.data.frame(u3)
  degrees = sapply(u3,function(x)
acos((u3[x,]*u3[x+1,])+(u3[,x]*u3[,x+1])/(sqrt(u3[x,]^2+u3[,x]^2)+sqrt(u3
[x+1,]^2+u3[,x+1]^2))))
  return(grades)
}

Distanceorigin <- function(trip)
{
  distance = sapply(trip, function(x) sqrt((trip[x,]-
trip[1,])^2+(trip[,x]-Origin[,1])^2))
  return(distance)
}
getallFeatures <- function(trip)
{
  speed = SpeedReckon(trip)

  meanSpeed = meanSpeedPerTrip(speed)

  speedDistr = SpeedDistribution(speed)

  accel = Acceleration(speed)

  decel = Deceleration(speed)

  accDistr = AccelerationDistribution(accel)
  decDistr = DecelerationDistribution(decel)

  meanDecel = meanDeceleration(decel)

  length = TripLength(trip)

  Nstops = NumbStops(speed)
)


  features=as.numeric(c(driver,meanSpeed,speedDistr,accDistr,decDistr,
meanDecel,length,Nstops))

  return(features)
}


# main
drivers = list.files("D:TFG/Task1/Task1/data/drivers")

randomDrivers = sample(drivers,size = 1)
data_notdriver = NULL
target = 0
names(target) = "target"
for (driver in randomDrivers)
{
  dirPath = paste0("D:TFG/Task1/Task1/data/drivers", "/", driver,"/")
  for(i in 1:50)
  {
    trip = read.csv(paste0(dirPath, i, ".csv"))
    features = c(getallFeatures(trip),target)
    data_notdriver = rbind(data_notdriver, features)
  }

}
target = 1
names(target) = "target"
submission = NULL
driver_trip_probabilities = NULL
for( driver in drivers)
{
  print(driver)
  dirPath = paste0("D:TFG/Task1/Task1/data/drivers","/", driver,"/")
  data_driver = NULL
  for(i in 1:200)
  {
    trip= read.csv(paste0(dirPath, i, ".csv"))
    features = c(getallFeatures(trip),target)
    data_driver = rbind(data_driver, features)
  }



  allData = rbind(data_driver, data_notdriver)

  allData <- as.data.frame(allData)


  names(allData)[-ncol(allData)] <- paste0('X', 1:(ncol(allData)-1))

  rownames(allData) <- NULL  # this line deletes names of rows "features"

  #allData = as.data.frame(sapply(allData,as.numeric))

  trans  =  preProcess(allData[,1:ncol(allData)-1],  c("BoxCox","center",
"scale"))
  allData_trans = data.frame(trans = predict(trans, allData))
  allData_trans$target <- allData$target



  inTrain  <-  caret::createDataPartition(allData_trans$target,  p  =  .85,
list = FALSE)
  training_data <- allData_trans[inTrain,]
  crossvalidation_data <- allData_trans[-inTrain,]
  g <- glm(target~.,data=training_data,family =binomial("logit"))
  p <- predict(g,crossvalidation_data, type="response")

  pred <- prediction (p,crossvalidation_data$target)
  perf <- performance(pred,"tpr", "fpr")

  #plot(perf,main="ROC Curves",col="blue")
  #abline(0,1,col="grey")



  #auc_rdock <- performance(pred,"auc")
  #auc.area_rdock <- slot(auc_rdock,"y.values")[[1]]
  #cat("AUC: \n")
  #cat(auc.area_rdock)
  #cat("\n\n")
  data_driver <- as.data.frame(data_driver)
  names(data_driver)[-ncol(data_driver)] <- paste0('X', 1:(ncol(allData)-
1))

  rownames(data_driver)  <-  NULL   #  this  line  deletes  names  of  rows
"features"
```

```
  trans          =          preProcess(data_driver[,1:ncol(data_driver)-1],
c("BoxCox","center", "scale"))
  data_driver_trans = data.frame(trans = predict(trans, data_driver))
  data_driver_trans$target <- data_driver$target
  p_final = predict (g,data_driver_trans,type ="response")
  labels = sapply(1:200,function(x) paste0(driver,"_",x))
  result = cbind(labels, p_final)

  driver_trip_probabilities = rbind(driver_trip_probabilities,result)

}

colnames(driver_trip_probabilities) = c("driver_trip","prob")
write.csv(driver_trip_probabilities,"D:TFG/Task4/driver_trip_probabilitie
s.csv", row.names = F, quote = F)
```