Predictive Model Development using Machine Learning Techniques

Eric Lozano Férriz

Abstract – The School of Engineering at UAB has a didactic blog composed of multiple posts where problems are solved by using Machine Learning techniques. The main objective of the project is to expand this blog with different educational publications where, in addition to explaining the development of the model, there are also explained previous steps such as data exploration, data preparation, and data analysis. Finally, the project also consists of different improvements on the blog and the participation in a competition where different teams compete to achieve the model with the best results.

Keywords – Machine Learning, Data analysis, Artificial Intelligence.

Resumen – Desde hace años, la universidad posee un blog didáctico formado por diversos posts donde se resuelven problemáticas mediante el uso de técnicas de Aprendizaje Computacional. El objetivo principal del proyecto es el de ampliar dicho blog con distintas publicaciones didácticas donde, además de la explicación del desarrollo del modelo, se expliquen las fases previas tales como la exploración, preparación y análisis de los datos. Por último, el proyecto también cuenta con una serie de mejoras en el blog y la participación en una competición abierta donde distintos equipos compiten por conseguir el modelo con mejores resultados.

Palabras clave – Aprendizaje Computacional, Análisis de datos, Inteligencia Artificial.

1 INTRODUCTION

MACHINE Learning and Deep Learning techniques have become a very important and useful way of solving complex problems, even in some cases, better than humans do.

Nowadays, engineers with high knowledge of these fields are in high demand so the School of Engineering at UAB has created the Data UAB Blog [1] in which students can publish didactic posts to prove they understand and can solve this type of problems as well as helping the community.

The blog provides students with tutorials so they can easily understand the concepts and follow the different machine learning techniques applied. Also, the blog could be very useful for companies looking for professionals in the field since it proves that the authors understand and are capable of explaining the techniques used to solve this type of problems. Furthermore, they are even capable of seeing how the authors work by taking a look at the code of their posts.

The content of the post explains all phases of the problem solving. Starting by understanding and exploring the data that is going to be used. Followed by an extensive data analysis including some graphical data visualization to understanding and proving important facts or relationships. And finally, after the model creation, exposing the results and conclusions obtained.

On the other side, companies also look for new engineers at the open machine learning competitions where a lot of people and teams compete to achieve the best results. This is the main reason for participating in a competition, to learn, understand, and prove you are able to achieve competitive results.

2 DATASETS SELECTION

All datasets have been chosen through a specific selection process prioritizing the use of different Machine Learning techniques to achieve a richer diversity the blog. During this process, the datasets have been also analyzed in order to find the more interesting ones that could be considered as a real-life problem that requires to be solved.

• Contact e-mail: ericlozano98@gmail.com
- Specialization realized: Computación
- Project tutored by: Jordi Gonzalez Sabaté & Pau Rodríguez (Ciencias de la computación)
- Academic Course: 2019/2020

The chosen datasets are listed below:

- **Medical Cost Personal Datasets [2]**: The main objective of this dataset is regression, basically to predict the charges of medical insurance. This dataset is composed of a total of 1338 clients and 7 attributes each corresponding to their personal information.
- **The Movies Dataset [3] and TMDB Movie Dataset [4]**: Both datasets have been used to compare two different types of movie recommender systems. The first one is composed of approximately 5000 movies and 20 attributes each indicating some information about the movie. On the other hand, the second dataset contains over 71,000 user ratings about movies.
- **League of Legends Ranked Games Dataset [5]**: The main objective is a binary classification, which is to predict which team is going to win the game. This dataset is composed of over 51,000 games and 61 attributes each representing important information about events that happened and choices made by the players during each of those events.

Two different recommender systems are implemented for the second dataset while regression and classification models will be implemented for the other two datasets respectively.

3 Machine learning posts

During this project, three different didactic posts have been created to expand the Data UAB blog [1]. These posts consist of different problem resolutions by using Machine Learning techniques. Moreover, their structure shows to the readers which should be the following steps when working in these type of problems.

3.1 Medical Insurance Charges Prediction

In this post, the Medical Cost Personal Dataset [2] has been used to create a model that predicts the charges of clients. First of all, the data has been analyzed to understand and find important facts and correlations between data attributes. Then, different models have been created and last of all, results have been compared to find which is the best one.

3.1.1 Data analysis

The first section of the post includes the exploration and analysis of the data. The main objective of data exploration is to introduce to the reader the information that is going to be used during the post. However, the data analysis objective is to learn interesting facts of the data as, for example, which attributes are highly correlated.

During the exploration, the shape of the dataset has been check to make an idea of how much information is available. Also, attributes have been shown and explained as well as some interesting metrics about numerical ones such as mean, standard deviation, etc. In addition, the dataset has been check to find and process missing values, but no missing values have been found.

The data analysis includes exploring attributes and relationships between them, representing these relations with histograms, pie charts, and other graphical visualizations.

One of the first analysis of the data done has been checking if the sex genre was balanced, which was true. The attribute representing the region of the clients has been also checked and discovered how the number of clients per region is practically balanced.

One important step when analyzing the data is to calculate the correlation values between the target attribute and the other ones. In order to this, the correlation matrix of the dataset attributes has been calculated and also has been discovered how there are two attributes strongly correlated with the charges: the BMI and the smoker status (Figure 1).

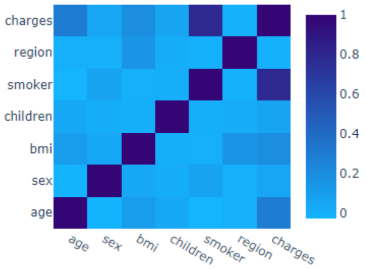


Fig. 1: Correlation Matrix of the attributes.

Once known which attributes are strongly correlated, the analysis has been based on the fact that medical insurance charges are highly influenced by attributes and relationships indicating the health risks.

Moreover, one important attribute analyzed is the BMI (Body Mass Index) of the client. The value of this attribute has been used to create four different groups representing the weight status of the client which can be: Underweight, Normal weight, Overweight, or Obese. After grouping all the clients by this new attribute, it can be seen how most of them have not a healthy weight (Figure 2). Furthermore, after analyzing the relationship between this attribute and the charges it can be appreciated how the more unhealthy weight status, the more expensive the charges are.

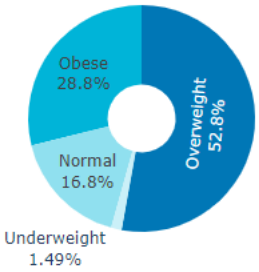


Fig. 2: Weight status proportions.

After analyzing the BMI of the clients, the next step has been grouping the clients in three different categories by their age. Then, it has been seen how the age of the client also presents a relationship with the insurance charges. As expected, the charges are more expensive for older clients because more health problems and risks need to be considered (Figure 3).

Another health indicator attribute that presents a strong relationship with the charges is the one that represents if

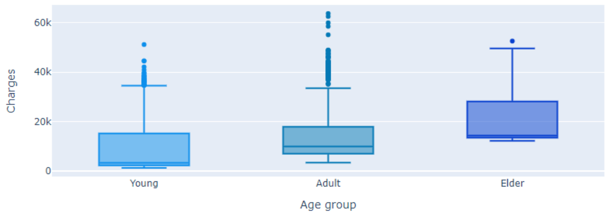


Fig. 3: Relationship between age categories and charges.

a client is smoker or not. It can be observed how being smoker results in insurance charges increase.

Finally, it has been analyzed the relationship between the two attributes that most represents the health of the client and the insurance charges. During this analysis, a scatter plot has been used to visually appreciate these relationships from which can be extracted that there is a clear difference in the charges between the smokers and the non-smokers. Furthermore, the charges difference between smokers and non-smokers increases a lot when the BMI exceeds 30, which means this person is considered to be obese so the risk is higher (Figure 4).

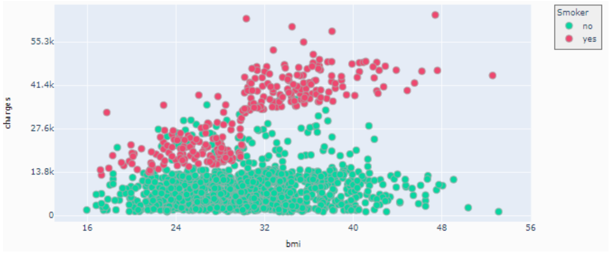


Fig. 4: Relationship between smoker, weight and charges.

3.1.2 Model Creation

The second section of the post, consists in the creation and evaluation of a Machine Learning model that predicts the insurance charges.

First of all, the data has been split into two different subsets, one for training the model and the other one to test it. Also, the test subset helps to check if there is overfitting, which means that our model has learned the specific data used for training and is no able to correctly predict on new data.

For this dataset, three different types of regression models have been developed: **Multiple Linear Regression** [6], **Decision Tree Regressor** [7] and **Random Forest Regressor** [8].

3.1.3 Results

One of the most significant information obtained from the data analysis is the fact that there is an important correlation between the smoker status, the BMI, and the charges (Figure 4). Moreover, there is an important increase in the charges of smoker people whenever their BMI value exceeds 30, which means they are considered to be Obese so the risk is higher.

Since the predicted value is the medical insurance charges, all models have been evaluated and compared by using the R-Squared Score and MAE (Mean Absolute Error) metrics (Figure 5).

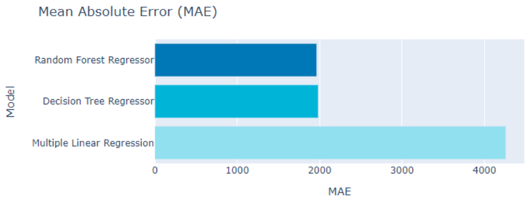


Fig. 5: MAE results for each model.

The MAE metric represents the absolute mean error of the model predictions, in other words, this value indicates approximately by how many dollars (over or under) each model is wrong when predicting. It can be seen how the Decision Tree model and the Random Forest one are the best models having similar MAE values of approximately 2000 dollars (Figure 5). These values are quite acceptable due to the fact that usually medical insurance charges are very expensive and $2000 is a margin error that could be accepted. Regarding the Multiple Linear Regression model, its predictions have approximately the double of error than the other two ones. This is because there is no linear correlation between the variables.

Furthermore, it can be observed how the Decision Tree and the Random Forest models are very accurate, while the MLR model is obviously the worst of them (Figure 6).

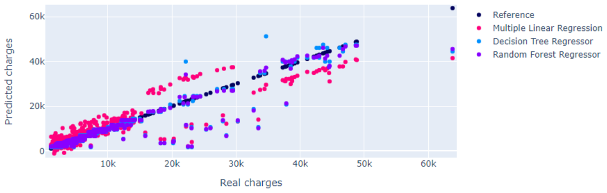


Fig. 6: Representation of each model predictions.

In conclusion, the data analysis has been very useful for understanding some insurance companies criteria when establishing the charges of the clients. Also, it has been possible to create even two models that given some personal information about a client, can predict the charges quite well.

In fact, the R-Squared score of our model is practically the same as the one achieved in the best kernel [9].

3.2 Movies Recommender System

Nowadays Recommender Systems are present in a lot of services used practically every day such as Amazon, Netflix, etc. So, this post main objective is to help the reader to understand how all these services successfully achieve to recommend the perfect movie, series, or whatever the product is.

There exist different types of Recommender Systems, most common ones are based on finding the similarities between recommended items or system users. In this post, two different types have been selected: Content-based and Collaborative-filtering.

On the one hand, **Content-based Recommender Systems** [10] are based on the fact that similar objects will have similar ratings. More concretely, this type of model uses information that defines the items content to create a similarity matrix. This way, knowing the items that a user likes the most, this similarity matrix can be used to search for the most similar items.

On the other hand, **Collaborative-filtering Recommender Systems** [11] are based on the fact that similar users will have similar item ratings. When using this type of model, recommendations for a specific user are generated by choosing the better-rated items of the most similar users.

3.2.1 Data analysis

Before starting with the model creation, the post exposes how exploring and analyzing the data is essential and necessary.

During the exploration, the number of instances and attributes of each dataset have been exposed. Furthermore, all attributes have been presented to the reader, as well as their data type, their distribution and some interesting metrics for the numerical attributes. A few examples of these metrics are the mean, the maximum and minimum values, or even the standard deviation.

One of the most important processes realized during data exploration has been checking if there are missing values in the data. The dataset containing the user ratings does not contain any missing value while the other one contains some missing values but, in this case, there has been no data processing applied because none of them has been used to fit the recommendation system.

Once all the data has been clearly exposed and the reader has knowledge of the available data, next section of the post is the data analysis. During this section attributes and relationships between them have been explored by representing them with histograms, pie charts, and other graphical visualizations.

To implement the Content-based model, all the attributes need to be analyzed and select only the ones that better represent a movie. This step is normally one of the most difficult ones to implement because it requires to create a list with words that represent all the important information of the movie. But, in this case, it does not represent a big problem, since the dataset selected contains a specific attribute with a list of keywords of the movie.

Although there is already one attribute representing the main keywords for each movie, other attributes that are not contained in this list have been analyzed to decide whether they should be added to the existing keywords list.

One of the attributes that have been analyzed and added to the keywords list has been the one specifying the genres of the movies. During this attribute analysis a wordcloud has been used to discover which are the most frequent genres: Drama, Comedy, and Thriller (Figure 7).

In addition, the movies spoken languages have been also analyzed during the data analysis and it has been observed how English is the most frequently spoken language in the dataset. More specifically, in approximately the 70% of movies.

However, before creating a Collaborative-filtering model, one interesting fact to know is the number of ratings per user. It has been seen how approximately 90% of the users have 200 ratings or less, which is normal since having this number of ratings is already a lot for a single user (Figure 8).

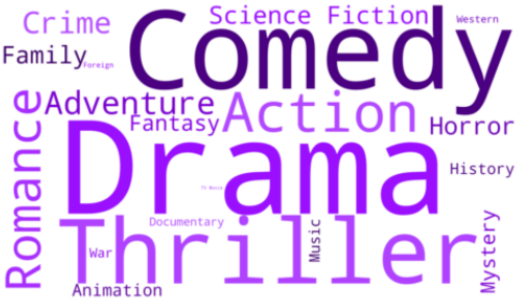


Fig. 7: Genres frequency represented by word size.

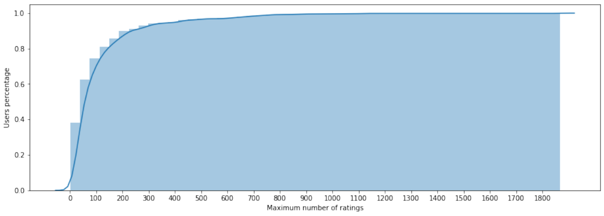


Fig. 8: Accumulative histogram of ratings per user.

Finally, after comparing the dataset containing all the movies with the other one containing the user ratings it has been discovered that about 20% of the movies have no ratings. From this information, it can be expected that a lot of movies will not be possible to be recommended when using the Collaborative-filtering model since nobody has ever rated them.

3.2.2 Model Creation

The creation of both types of Recommender Systems is explained and implemented in the next section of the post. During this section, the processing of the data and the creation of the model are explain in detail.

In order to create the Content-based model, it has been necessary to represent the content of movies by one attribute containing the keywords, genres, and spoken languages. After this attribute has been defined, it has been processed by generating TF-IDF [12] vectors for each movie, which have been used later to calculate the cosine similarity between movies and creating a similarity matrix of all the movies.

Once created the similarity matrix, the Content-based model is ready to recommend. All it needs to do is, given a movie, search in the similarity matrix for the most similar movies and then recommend them.

On the other hand, the dataset containing the ratings has been used to create the Collaborative-filtering model. After the information collected during the data analysis, it has been decided to filter this dataset keeping only the users and movies over a certain threshold. In other words, the movies with less than 100 ratings and the users with less than 40 rated movies have not been used to create the model.

Once the ratings data have been filtered, an **SVD** model [13] has been created and used to predict the ratings of the movies that a certain user has not seen yet. This is how the model predicts the ratings of movies and recommends the ones with higher rating scores that the user has not seen yet.

3.2.3 Results

The last section of the post contains the results of the models created and the general conclusions.

Regarding the Content-based model, the results obtained are really good and it can be seen how this model uses the content of the movies to recommend the most similar ones (Table 1).

Movie	Cosine similarity
Planet of the Apes	0.44
Gravity	0.43
Aliens	0.43
Alien 3	0.43
Star Trek Into Darkness	0.42

Table 1: Recommended movies given Avatar.

On the other hand, the Collaborative-filtering model has obtained a mean MAE value of 0.64, which is a valid error range value. This error can be seen in Table 2, where the predicted ratings and the real ratings of some movies are exposed.

Movie	Ground-Truth Rating	Predicted Rating
Hulk	3	3.29
End of Days	4	4.08
The Haunted Mansion	3	3.60
Ben-Hur	3	3.35
The Dilemma	3	3.58

Table 2: Comparison between real and predicted ratings.

From these results can be concluded that the Content-based model works really well since the attributes used to describe the movies define really well the content. Also, the Collaborative-filtering works well, but it would be better to have more ratings since this type of Recommender Systems needs a lot of them.

Finally, after comparing our Recommender Systems created with the best Kernel [14], it can be observed how the recommendations of our models are also pretty good.

3.3 League of Legends Winning Team Prediction

This post is dedicated to creating a model able to predict which team is going to win in one of the most famous team-based strategy games: League of Legends. In order to create this model, a dataset containing information about more than 51,000 games has been used.

Like all strategy games, there are several objectives that give a huge advantage to the team who take them. So, the information about these objectives for each team and other important characteristics contained in the dataset will help in the way to train a winner prediction model.

3.3.1 Data analysis

The first section of the post is the exploration and analysis of the dataset to introduce all the data to the reader and expose the reasons for the decisions taken when filtering or processing some attributes.

During the exploration of the data, it has been seen how the dataset is composed of over 51,000 games and 61 attributes which is really good because having a lot of information would make it easier to achieve a model with good accuracy.

Moreover, in this section of the post has been also observed how there are different types of attributes. Some of them are repeated twice because they represent information associated with each team and include information about all the player's choices before starting the match (champions, spells, etc.) and also about the number of objectives taken for each team. In addition, there are also attributes representing information about the game settings and which team has first taken certain objectives.

There has been also a process of checking whether the dataset contains missing values, which has result in a negative response, meaning there are no missing values in the data.

After calculating some metrics for the numeric attributes it has been observed that all games of the dataset are from the same season of the game. This means that main strategy of games is the same for all of them since critical strategy changes are implemented whenever a season ends. Also, it has been discovered that being the winning team does not depend on which team you belong to.

When analyzing the data needed to create the model one important step was checking if all the attributes were really useful when predicting which team is going to win. To do this, the correlation matrix has been generated and the values of the winner attribute have been analyzed. After observing and analyzing these correlation values, it has been possible to filter the attributes and keep only the ones that contribute to predicting which team wins (Figure 9).

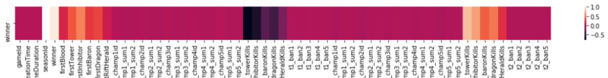


Fig. 9: Winner attribute correlations.

The number of attributes after filtering by their correlation values has been reduced from 61 to 11. This shows how important it is to analyze the data and keeping only the useful information.

Once the attributes have been chosen, the next step has been to analyze and graphically visualize the relation between them and the winner attribute. It can be observed how there is an obvious relation between the attributes representing which team has been the first one on taking some specific objective and the winning team (Figure 10). Same thing happens when analyzing the relation between the attributes representing the number of objectives taken by each team and the winning team.

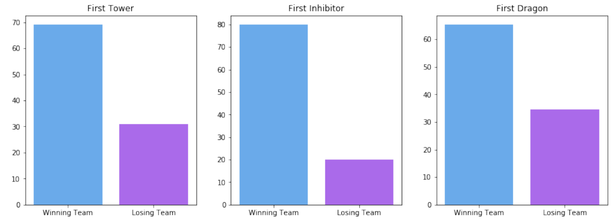


Fig. 10: Relations between first team taking a certain objective attributes and being the winner.

3.3.2 Model Creation

Once the data has been analyzed and introduced to the reader, the next section of the post is about creating a classification model that predicts which team is going to win. In fact, after analyzing different possible machine learning models for resolving this problem, due to the attribute types and the small range of their values, it has been decided that the most appropriate model would be a **Decision Tree**.

The first thing to do before starting with the model creation is to filter the attributes and keep only the useful ones seen during the previous data analysis section. Furthermore, the dataset needs to be split into two different sets: one of them to training the model and the other one to testing and evaluating it.

While training of the model, the GridSearchCV function from scikit-learn has been explained to the reader and used to find which combination of some of the Decision Tree parameters achieves the best results. More concretely, the parameters modified have been the maximum depth, the criterion, and the splitter.

3.3.3 Results

During the last section of the post the model created has been evaluated and the results have achieved a 97% of accuracy, which is a really good result for this problem.

Once we know the model created can correctly predict which team is going to win with a high accuracy score, the next step has been to represent the ROC curve of the model to see how it works (Figure 11). The ROC curve is a visual representation of the relation between the True Positive Rate (TPR) and the False Positive Rate (FPR) with different thresholds.

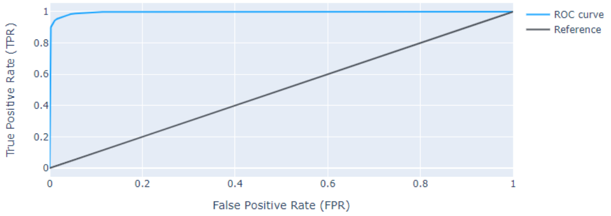


Fig. 11: ROC curve of the Decision Tree model.

By taking a look at the ROC curve generated can be appreciated the results are positive since the curve reaches practically 100% of the TPR with low FPR.

The last step during the evaluation of the model has been trying to represent the Decision Tree model in a graph way to explain to the reader some interesting things about how this model internally works. But, due to the huge size of the model, representing it has been impossible and the solution chosen has been to represent and explain a more simpler model (Figure 12).

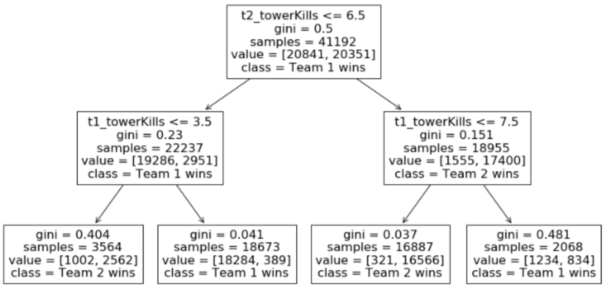


Fig. 12: Simpler Decision Tree representation.

From all the analysis and results obtained can be concluded that the data has been enough to create a good model able to predict with high accuracy which team is going to win a game. In fact, it can also be concluded how not all information about each game is relevant. At first, the dataset had more than 60 attributes and after analyzing them, only 11 have been used to train the model.

Furthermore, one interesting conclusion about the data analysis is that attributes indicating the number of towers taken by each team are the most important ones because of the strong correlation with the winner attribute. In fact, it can be seen how in the simpler Decision Tree representation where depth is limited to 3, these are the only attributes used in all decision nodes.

Finally, it is necessary to highlight the fact that the precision of the created model is even better than the achieved in the best kernel [15].

4 Kaggle Competition

One interesting part of this project has been the opportunity of participating in a machine learning competition as a team. The team is composed of another student realizing a similar project and myself. However, the main objective of collaborating and participate as a team has been the challenge of solving the problem by ensembling the models of both members of the team.

The competition is named **Shelter Animal Outcomes [16]** and is about the life of the animals when they leave the animal shelter. The dataset used is composed of multiple attributes about animals and the objective is predicting the outcome of them when they leave. More concretely, these outcomes can be: being adopted, die, being sacrificed, being returned to the owner, or being transferred.

4.1 Data analysis

Before starting with the model creation, the first thing done has been exploring the data and analyzing the relationships between the attributes.

While exploring the data, it has been discovered that the dataset has over 26,500 animal instances that would be used to create the model. Also, each instance is composed of a total of 10 attributes.

Another important fact seen while analyzing the dataset, is that the attribute indicating the sex genre of the animal also indicates the reproduction status, which means this attribute needs to be processed to split this information. Moreover, there is also a problem with the age attribute because it has been defined in different age units as for example: months, years, or even weeks.

After obtaining some information on attributes, it has been seen how the attributes Breed and Color have a total of 1380 and 366 unique values each. This will probably be a problem when creating a model because of the large number of possible values causing these attributes to not provide any relevant information.

The last step realized during the data exploration has been checking if the data has missing values. Fortunately, most of the missing values found wherein two attributes that would not be present on the set that has to be predicted for the competition. In other words, the attributes containing most of the missing values will not be used. On the other hand, due to the small number of missing values on the Sex and Age attributes, it has been decided to delete the instances containing them.

During the data analysis one of the first things discovered has been that the dataset only contains dogs and cats as animals. Moreover, the next step has been analyzing the outcome to take a look on the proportions of each outcome class (Figure 13).

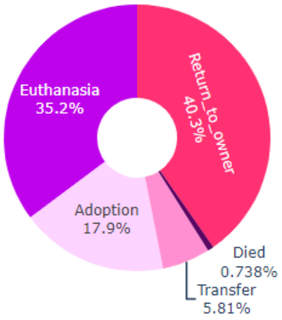


Fig. 13: Outcome proportions of the train dataset.

As can be observed, the Died and Transfer classes have small proportions so they can be expected to not get accurate predictions. In addition, the most probable scenario will be the one where the three bigger classes absorb the predictions, especially the Return_to_owner and Euthanasia classes.

Another analysis realized has been checking the proportions of the outcome classes grouping them by the sex genres. What has been discovered is that the outcomes are equally proportioned for both sex genres.

Finally, the last analysis realized consisted of grouping the data by the age values. Due to the different age units used for representing this attribute, the first step has been normalizing it converting all values to years units. Then, three different categories have been defined: Puppy, Adult, and Senior. Once the data has been grouped, the relationship between the age categories defined and the outcome classes has been graphically represented (Figure 14).

There are two interesting facts that can be extracted from Figure 14 as for example that Puppy is the most common age category when talking about transfer or adoption. Also, it can be observed how the ones returned to owners tend to be adult animals.

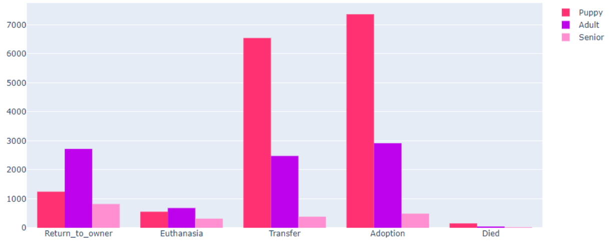


Fig. 14: Relationship between age categories and outcomes.

4.2 Model Creation

After exploring and analyzing the data, it has been decided to create and train an **ensembling of different classifier models** to predict the outcome of the test dataset.

Furthermore, the test dataset does not have the outcome attribute since it is the one that has to be predicted and delivered on the competition platform. Once predictions are delivered, they are evaluated and the team receives a score based on how well the model works.

As a team, the work has been distributed. My team partner has chosen to train two different machine learning models: Random Forest Classifier and XGBoost. On the other hand, I have chosen to train **two Neural Networks**.

4.3 Ensembling of the models

Once all the models have been tested the ensembling model has been created.

An ensemble is a set of machine learning models where each one produces a different prediction and these are combined to obtain a single prediction. The advantage of combining different models is that errors tend to average out resulting in a better generalization error.

In this case, all models are combined into a Voting Classifier ensemble model. The main characteristic is that each model votes one class and the final prediction will be what most models vote for (Figure 15).

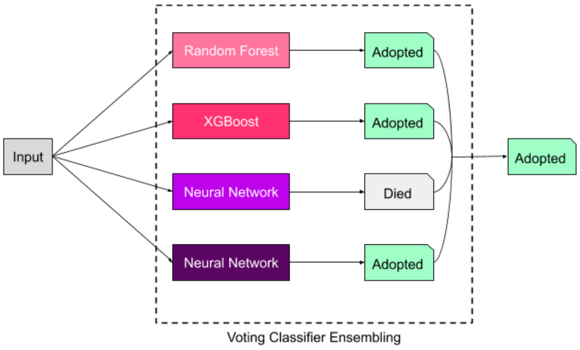


Fig. 15: Voting Classifier ensembler model example.

4.4 Results

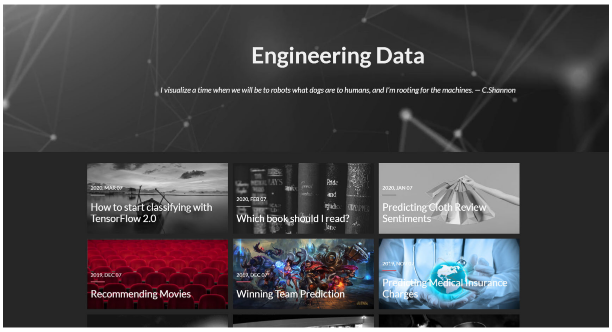
When evaluating the Neural Networks created separately, they have obtained an accuracy of approximately 61% and 60% each. However, when we evaluate the ensembling of all models created by the team, the accuracy increases to 64% which means that the combination of the models achieves better results than them separately.

After creating the ensembler model, all instances of the test dataset have been predicted and uploaded to the Kaggle competition platform obtaining a multiclass loss of 0.88 which is approximately 30% worse than the better model of the leaderboard.

Finally, we have studied the solution of best kernel available [17] and compare it to ours. It has been observed how they filter and apply more processes to the training dataset to generate new attributes that represent new information. Also, this particular team has only trained a Random Forest Classifier model.

5 Blog Improvement

The most important improvement applied to the blog has been the addition of the three new machine learning posts commented previously (Figure 16).



Engineering Data

I visualize a time when we will be to robots what dogs are to humans, and I'm rooting for the machines. — C.Shannon

2020, MAR 07 How to start classifying with TensorFlow 2.0

2020, FEB 07 Which book should I read?

2020, JAN 07 Predicting Cloth Review Sentiments

2019, DEC 07 Recommending Movies

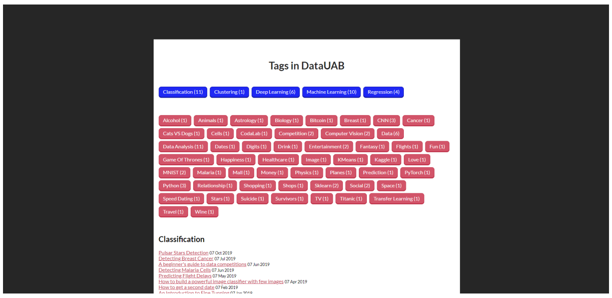
2019, DEC 07 Winning Team Prediction

2019, NOV 07 Predicting Medical Insurance Charges

Fig. 16: New three posts added.

The diversity of the posts is one important fact that needs to be considered and also the main reason of having multiple categories for each post. Since the blog is principally dedicated to students that want to learn, it would be interesting to have a section of the blog specially built to group the posts by their categories or tags (post-content keywords).

This section was already implemented but in a too much simple and basic way (Figure 17). So, given its importance, it has been selected to be the blog improvement apart from adding three new posts.



Tags in DataUAB

Classification (11) Clustering (1) Deep Learning (6) Machine Learning (10) Regression (4)

Alcohol (1) Animals (1) Astrology (1) Biology (1) Bitcoin (1) Breast (1) CNN (3) Cancer (1) Cats VS Dogs (1) Cells (1) CodaLab (1) Competition (2) Computer Vision (2) Data (6) Data Analysis (11) Dates (1) Digits (1) Drink (1) Entertainment (2) Fantasy (1) Flights (1) Fun (1) Game Of Thrones (1) Happiness (1) Healthcare (1) Image (1) KMeans (1) Kaggle (1) Love (1) MNIST (2) Malaria (1) Mall (1) Money (1) Physics (1) Planes (1) Prediction (1) PyTorch (1) Python (3) Relationship (1) Shopping (1) Shops (1) Sklearn (2) Social (2) Space (1) Speed Dating (1) Stars (1) Suicide (1) Survivors (1) TV (1) Titanic (1) Transfer Learning (1) Travel (1) Wine (1)

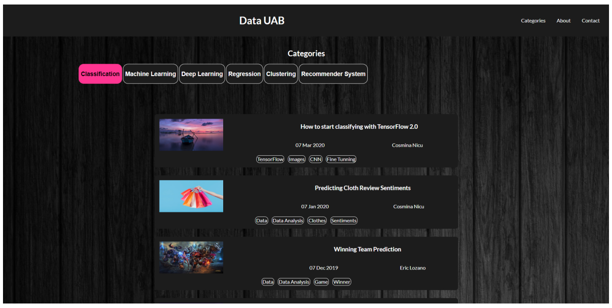
Classification

Pulsar Stars Detection 07 Oct 2019
Detecting Breast Cancer 07 Jul 2019
A beginner's guide to data competitions 07 Jun 2019
Detecting Malaria Cells 07 Jun 2019
Predicting Flight Delays 07 May 2019
How to build a powerful image classifier with few images 07 Apr 2019
How to get a second date 07 Feb 2019
An Introduction to Fine Tunning 07 Jan 2019

Fig. 17: Categories section before.

The section improvement has required to realize an extensive investigation of the page structure and how it is built. Moreover, in order to apply a more modern design it has been needed to modify some of the front-end files of the page.

Furthermore, the main idea of the new section is to allow the viewer filter the post depending on the category wanted. To do that, some back-end logic has been needed as well as some Javascript functions that toggle the visibility of the posts according to the category selected (Figure 18).



Categories

Classification Machine Learning Deep Learning Regression Clustering Recommender System

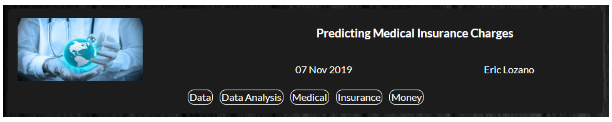
How to start classifying with TensorFlow 2.0
07 Mar 2020 Cosmina Nicu
TensorFlow Images CNN Fine Tunning

Predicting Cloth Review Sentiments
07 Jan 2020 Cosmina Nicu
Data Data Analysis Clothes Sentiments

Winning Team Prediction
07 Dec 2019 Eric Lozano
Data Data Analysis Game Winner

Fig. 18: Categories section now.

Besides, the representation of the different posts is similar to the homepage of the blog but with some extra information: the author and the post tags (Figure 19).



Predicting Medical Insurance Charges
07 Nov 2019 Eric Lozano
Data Data Analysis Medical Insurance Money

Fig. 19: Representation of a post in the improved section.

6 Conclusions

The main goal of the project has been successfully achieved by extending the Data UAB Blog [1] repository with three new didactic Machine Learning posts where all is clearly explained and detailed. Also, one important section of the blog has been considerably improved providing a friendly user interface where posts can be filtered according to their classifications (Section 5).

Regarding the Medical Insurance Charges Prediction (Section 3.1) it has been discovered that the attributes with the highest impact on the charges are the ones that have a relation with the health status of the client. This means that people with an unhealthy situation will have expensive charges on their medical insurance. Also, it has been possible to create a model able to predict the medical charges within a small mean margin error of 2,000 dollars.

On the other hand, during the Movies Recommender System (Section 3.2) it has been seen how the number of ratings is a very important fact when creating a Collaborative-filtering model since this type of Recommender System needs a lot of ratings from users to become a good recommender. However, the Content-based model recommendations belong to a set of movies with a clearly similar content.

Moreover, in the League of Legends Winning Team Prediction (Section 3.3) has been seen how being the first team on taking certain objectives or the number of objectives killed (towers, inhibitors, or even dragons) are factors highly correlated with being the winning team. These attributes have provided enough information to train a model with a 97% accuracy.

Finally, despite not having achieved excellent results due to the difficulty of the data and the short time that has been available to dedicate to the competition, it has been possible to observe, understand and learn from what top leaderboard teams have done to achieve the best results.

Agreements

First of all, I would like to thank all the people that have taken care of adapting schedules and finding virtual alternatives to all events that unfortunately can not be done due to the COVID-19 situation.

Moreover, I would also like to thank my tutor Jordi González and co-tutor Pau Rodríguez for all the guidance and support provided during the project that would not have been possible without them. Also, thanks to Jordi Pons who has been resolving doubts and informing all students about different deliveries and deadlines.

Finally, thanks to Cosmina Ioana Nicu for being a hard-worker teammate during the Kaggle competition.

References

[1] ”Data UAB”, Datauab.github.io, 2020. [Online]. Available: https://datauab.github.io/. [Accessed: 25- Jun- 2020].

[2] “Medical Cost Personal Datasets.” [Online]. Available: https://kaggle.com/mirichoi0218/insurance. [Accessed: 25-Jun-2020]

[3] “The Movies Dataset.” [Online]. Available: https://kaggle.com/rounakbanik/the-movies-dataset. [Accessed: 25-Jun-2020]

[4] “TMDB 5000 Movie Dataset.” [Online]. Available: https://kaggle.com/tmdb/tmdb-movie-metadata. [Accessed: 25-Jun-2020]

[5] “(LoL) League of Legends Ranked Games.” [Online]. Available: https://kaggle.com/datasnaek/league-of-legends. [Accessed: 25-Jun-2020]

[6] W. Kenton, “How Multiple Linear Regression Works,” Investopedia. [Online]. Available: https://www.investopedia.com/terms/m/mlr.asp. [Accessed: 27-Jun-2020]

[7] R. S. Brid, “Decision Trees — A simple way to visualize a decision,” Medium, 26-Oct-2018. [Online]. Available: https://medium.com/greyatom/decision-trees-a-simple-way-to-visualize-a-decision-dc506a403aeb. [Accessed: 27-Jun-2020]

[8] J. M. Heras, “Random Forest (Bosque Aleatorio): combinando árboles,” IArtificial.net, 10-Jun-2019. [Online]. Available: https://iartificial.net/random-forest-bosque-aleatorio/. [Accessed: 27-Jun-2020]

[9] “EDA + Regression.” [Online]. Available: https://kaggle.com/hely333/eda-regression. [Accessed: 27-Jun-2020]

[10] B. Balu, “Content-Based Recommender System,” Medium, 16-Oct-2019. [Online]. Available: https://medium.com/@bindhubalu/content-based-recommender-system-4db1b3de03e7. [Accessed: 27-Jun-2020]

[11] “Collaborative Filtering — Recommendation Systems,” Google Developers. [Online]. Available: https://developers.google.com/machine-learning/recommendation/collaborative/basics. [Accessed: 27-Jun-2020]

[12] C. Maklin, “TF IDF — TFIDF Python Example,” Medium, 21-Jul-2019. [Online]. Available: https://towardsdatascience.com/natural-language-processing-feature-engineering-using-tf-idf-e8b9d00e7e76. [Accessed: 27-Jun-2020]

[13] M. Bhattacharyya, “Beginner’s Guide to Creating an SVD Recommender System,” Medium, 26-May-2020. [Online]. Available: https://towardsdatascience.com/beginners-guide-to-creating-an-svd-recommender-system-1fd7326d1f65. [Accessed: 25-Jun-2020]

[14] “Film recommendation engine.” [Online]. Available: https://kaggle.com/fabiendaniel/film-recommendation-engine. [Accessed: 27-Jun-2020]

[15] “Let’s Predict League of Legends Match Score!” [Online]. Available: https://kaggle.com/gulsahdemiryurek/let-s-predict-league-of-legends-match-score. [Accessed: 27-Jun-2020]

[16] “Shelter Animal Outcomes.” [Online]. Available: https://kaggle.com/c/shelter-animal-outcomes. [Accessed: 25-Jun-2020]

[17] “Quick Dirty RandomForest.” [Online]. Available: https://kaggle.com/mrisdal/quick-dirty-randomforest. [Accessed: 27-Jun-2020]

[18] D. Mishra, “Regression: An Explanation of Regression Metrics And What Can Go Wrong,” Medium, 06-Dec-2019. [Online]. Available: https://towardsdatascience.com/regression-an-explanation-of-regression-metrics-and-what-can-go-wrong-a39a9793d914. [Accessed: 25-Jun-2020]

[19] J. TORRES.AI, “Learning process of a neural network,” Medium, 28-Apr-2020. [Online]. Available: https://towardsdatascience.com/how-do-artificial-neural-networks-learn-773e46399fc7. [Accessed: 25-Jun-2020]