Universitat Autònoma de Barcelona

Dipòsit digital de documents de la UAB

This is the **published version** of the bachelor thesis:

Vázquez Díaz, Leopoldo José; Garcia Calvo, Carlos, dir. Generación y comparativa de modelos analíticos AutoML con el uso de datos no tabulados mediante NLP contra modelos de Data Science. 2021. (958 Enginyeria Informàtica)

This version is available at https://ddd.uab.cat/record/248536

under the terms of the BY-NC-ND license

Generación y comparativa de modelos analíticos AutoML con el uso de datos no tabulados mediante NLP contra modelos de Data Science

Leopoldo José Vázquez Díaz

Resumen– El autoaprendizaje automático es una subárea del aprendizaje de máquinas, donde se aplica algoritmos de aprendizaje automatizado al procedimiento de aprendizaje automático común de problemas reales. Enfocado en automatizar varias partes del proceso, disminuyendo su complejidad y coste de desarrollo. Hoy en día, existen varias afirmaciones sobre millones de empresas, que tienen un gran potencial de mejora, no puedan explotar las capacidades del aprendizaje automatizado. Ya que, no logran alterar o mejorar su negocio por medio del análisis de sus datos, por falta de conocimientos o de experiencia en el sector. El uso de este proceso, ofrece ventajas al programador, de no necesitar definir su propia arquitectura, así logrando la producción de soluciones más rápidas y más sencillas. El objetivo de este proyecto, es emplear este proceso de aprendizaje automático, a una serie de datos no estructurados, aplicando un procesamiento de lenguajes naturales. De este modo, logrando generar un modelo analítico, sobre el cual poder extraer y crear unas conclusiones, comparándolas y confrontándolas contra un sistema común de aprendizaje automatizado.

Palabras clave– Aprendizaje automático, Autoaprendizaje automático, AutoML, Inteligencia Artificial, PLN, Procesamiento de lenguajes naturales.

Resum– L'autoaprenentatge automàtic és una subàrea de l'aprenentatge de màquines, on s'aplica algoritmes d'aprenentatge automatitzat al procediment d'aprenentatge automàtic comú de problemes reals. Enfocat a automatitzar diverses parts del procés, disminuint la seva complexitat i cost de desenvolupament. Avui en dia, hi ha diverses afirmacions sobre milions d'empreses, que tenen un gran potencial de millora, no puguin explotar les capacitats de l'aprenentatge automatitzat. Ja que, no aconsegueixen alterar o millorar el seu negoci per mitjà de l'anàlisi de les seves dades, per falta de coneixements o d'experiència en el sector. L'ús d'aquest procés, ofereix avantatges al programador, de no necessitar definir la seva pròpia arquitectura, així aconseguint la producció de solucions més ràpides i més senzilles. L'objectiu d'aquest projecte, és emprar aquest procés d'aprenentatge automàtic, a una sèrie de dades no estructurats, aplicant un processament de llenguatges naturals. D'aquesta manera, aconseguint generar un model analític, sobre el qual poder extreure i crear unes conclusions, comparant-les i confrontant-les contra un sistema comú d'aprenentatge automatitzat.

Paraules clau– Aprenentatge automàtic, Autoaprenentatge automàtic, AutoML, Intel·ligència Artificial, PLN, Processament de llenguatges naturals.

Abstract– Automatic machine learning is a sub-area of machine learning that applies its algorithms to a common process of machine learning with real problems, which is focused on automating various parts of the process, reducing its complexity and development cost. Nowadays, there are a lot of investigations that shows how millions of companies cannot exploit their big potential in machine learning, because of the lack of experience in the sector. So, they cannot alter or improve their business through the analysis of their data. The use of this process offers advantages to the programmer, not needing to define his own architecture, so achieving the complete production with faster and easier solutions too. The objective of this project is to use this automatic machine learning process with a group of unstructured data, applying natural language processing. Thereby, managing to generate a model, which one can extract and create conclusions, comparing and confronting them against a common machine learning system, used in the field of data science.

Index Terms– Automated machine learning, AutoML, Artificial Intelligence, Machine learning, Natural Language Processing, NLP.

1 Introducción

El aprendizaje de máquina, conocido en inglés como "Machine Learning" (ML), es un subcampo de la Computación y una rama de la Inteligencia Artificial (IA). Su origen proviene de los años 60, donde, con su construcción se tenía en mente, el estudiar el reconocimiento de patrones y el aprendizaje de las máquinas. El ML ha ido evolucionando con los años y actualmente, es un método de análisis de datos que automatiza la construcción de modelos analíticos, cuyo objetivo es desarrollar técnicas que permitan a los ordenadores aprender de los datos, identificar patrones y tomar decisiones con la mínima intervención humana.

La idea principal del proyecto es la búsqueda de una solución más ágil y eficaz a un problema del ámbito de AutoML, realizando un posible planteamiento y desarrollo inicial de un proyecto de ML, de forma más rápida, ahorrando tiempo y costes. El origen del proyecto viene dado al percatarse de

• E-mail de contacto: leopoldojose.vazquez@e-campus.uab.cat
• Mención realizada: Computación
• Trabajo tutorizado por: Carlos García Calvo (Departamento de Ciencias de la Computación).
• Curso 2020/21

que este posible avance y mejora, incrementaría la eficacia de creación e introducción de modelos de ML en el proceso de Operaciones de "Machine Learning" (MLOps), que normalmente, son realizados en las empresas que utilizan procesos de ML. De ahí que, toda esta mejora puede venir dada por el uso e implementación de AutoML en el ciclo de vida del MLOps.

Para esto, en este proyecto se hace uso de unos Datasets o conjunto de datos, los cuales están creados mediante una combinación de correos electrónicos, y que pueden ser utilizados en una empresa típica del sector bancario. Cada correo electrónico está clasificado y se busca enseñar a modelos de ML con estos datos a predecir posibles próximos casos. Al ser creados por combinaciones aleatorias, se respeta la privacidad de los mismos, pero manteniendo la integridad y significado necesarios para poder aprender de ellos y utilizarlos posteriormente para el aprendizaje de los modelos, creados mediante el proceso de AutoML. Esto conlleva al tratamiento y uso de un procesamiento de lenguaje natural, NLP, en el proceso de desarrollo y ejecución de los modelos del proyecto. El procesamiento de lenguaje natural (PLN) o abreviado en inglés NLP, es un campo de las ciencias de la computación, de la inteligencia artificial y de la lingüística que estudia las interacciones entre las computadoras y el lenguaje humano.

Con la finalización de este proyecto, se busca ver la viabilidad, capacidad y rentabilidad del AutoML comparado con el método actual y manual de ML aplicado en MLOps, habiendo investigado profundamente sus opciones y capacidades.

2 Estado del Arte

Para profundizar un poco más y terminar de introducir el proyecto, en este apartado se procederá a explicar el concepto principal, AutoML que es el centro del proyecto, explicando su estado y evolución actual.

El autoaprendizaje automático o Automated Machine Learning, en inglés, es el conjunto de ideas, técnicas, prácticas, herramientas y soluciones que permiten automatizar la producción de modelos de ML. AutoML se define como una combinación de automatización y ML. Tiene como objetivo automatizar tantos pasos como sea posible dentro de un flujo de trabajo de ML, logrando un mejor rendimiento del modelo con la mínima mano de obra posible. Además, simplifica la realización de tareas repetitivas y mecánicas, que son necesarias o habituales en el ciclo de vida de preparación de un modelo de ML, especialmente cuando estas se realizan mediante conocimientos y prácticas estandarizadas.

En los últimos años, la incursión de la IA y el ML en el desarrollo de soluciones empresariales ha ido incrementando con los años, y a día de hoy, representa una parte importante para varias compañías y tema de interés para muchas otras. Este aumento del uso del ML viene determinado por la madurez y avance de las nuevas tecnologías. También, es provocado por la capacidad que ofrece el "Cloud"para disponer de infraestructuras a medida, de gran potencia de computación y con una eficiencia en costes muy elevada. El problema que existe con el ML, son sus posibles problemas de viabilidad, de elevada complejidad y el coste al intentar desarrollar un modelo para una empresa. Eso genera, que muchas compañías y usuarios, por su falta de experiencia y conocimientos en el ámbito de ML, no hayan dado el paso a esta tecnología y no puedan explotarla.

El aprendizaje profundo se ha aplicado en varios campos y se ha utilizado para resolver muchas tareas desafiantes de IA, en áreas como la clasificación de imágenes, la detección de objetos y el modelado del lenguaje. Se han ido proponiendo redes neuronales cada vez más complejas y profundas. Cabe destacar, sin embargo, que todos los modelos hasta ahora, fueron diseñados manualmente por expertos mediante un proceso de prueba y error, lo que significa que incluso los expertos requieren recursos y tiempo sustanciales para crear modelos que funcionen bien.

Esto es posible, gracias a la generación de modelos, los cuales se han creado empleando un conjunto de algoritmos, que otorgan la autonomía necesaria a los ordenadores para que puedan aprender de sus errores u optimicen sus aciertos de forma autónoma. El objetivo que se persigue, es aumentar la eficiencia, y así, las máquinas se pueden encargar de trabajos automatizables, los programadores pueden centrarse en otras tareas más difíciles e importantes.

También persigue el sistematizar la ejecución de tareas de carácter exploratorio y heurístico orientadas, a decidir los elementos estructurales que conforman la arquitectura y topología del modelo de ML desarrollado. Promete la liberación de los expertos en ML, al automatizar algunos de los pasos, como la optimización automatizada de hiperparámetros y que les permite centrarse en pasos más sutiles, pero críticos, del flujo de trabajo de ML, como en definición del problema. Claramente, AutoML está diseñado para aumentar y no reemplazar a los expertos en ML, además de equipar a los usuarios no expertos de ML, permitiéndoles extraer información de sus datos sin tener una gran experiencia.

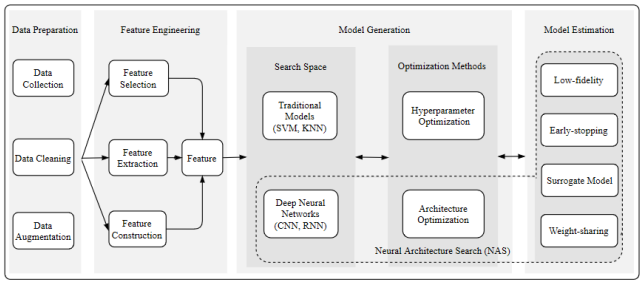


Fig. 1: Esquema AutoML

Con el crecimiento exponencial de la potencia informática, AutoML se ha convertido en un tema candente tanto en la industria como en el mundo académico. Un sistema AutoML completo puede realizar una combinación dinámica de varias técnicas para formar un sistema de canalización ML de extremo a extremo fácil de usar (como se muestra en la Figura 1). Muchas empresas de IA han creado y compartido públicamente dichos sistemas (por ejemplo, Cloud AutoML de Google, Azure AutoML de Microsoft o SageMaker Autopilot de Amazon) para ayudar a las personas con poco o ningún conocimiento de ML a crear modelos personalizados de alta calidad.

Fig. 2: Ciclo de Vida ML

Además, AutoML ofrece algunos beneficios adicionales, como evitar la introducción de errores provocados por el trabajo manual y también, pudiendo lograr establecer rápidamente una línea de base competitiva para un problema empresarial determinado. Estos objetivos generan una serie de cambios en el ciclo de vida que seguiría un proceso de ML normal (Figura 2) que está formado por ocho fases. El AutoML también está formado por el mismo número de fases, como se puede apreciar en la Figura 3, pero cada fase es distinta parcialmente o totalmente a la normal.

Fig. 3: Ciclo de Vida AutoML

3 Objetivos

El objetivo principal de este trabajo final de grado, es la generación y comparativa de modelos ML. Por una parte, generado de forma tradicional y ya presente en la metodología de MLOps y por la otra, creado mediante AutoML.

El resultado final ha de ser, obtener un modelo funcional creado mediante AutoML, permitiendo producir unas conclusiones para extraer unos resultados de esta nueva herramienta, comprobando y demostrando las ventajas del uso del AutoML y una realizar una comparativa.

Para cumplirlo, se plantean los siguientes objetivos:

- Buscar, recolectar y aprender información sobre AutoML.
  - Aprendizaje sobre conceptos MLOps y AutoML.

  Se realiza una investigación y aprendizaje sobre los conceptos y plataformas necesarios en el proyecto.
- "Data Preparation" (Preparar Dataset).
  - Generación del código creador de correos electrónicos.

  Se busca obtener y adaptar los datos para su uso óptimo en el desarrollo del proyecto.
- Ejecutar proceso AutoML sobre una solución Cloud.
- Obtener resultados de ambos métodos de ML.
  - Optimizar parámetros y resultados para su comparación.

  Recolectar y analizar preliminarmente de los resultados obtenidos durante y posteriormente la ejecución.
- Realizar comparativa de modelos, AutoML y MLOps (Data Science).
- Buscar posibles integraciones con la plataforma actual de MLOps.
- Realizar conclusiones.

  Al conseguir y adquirir todos los resultados, realizar una comparativa para mostrar similitudes y diferencias. Así, logrando extraer unas conclusiones y posibles mejoras e integraciones del proyecto. Se puede consultar la planificación seguida en el proyecto, en el Apéndice A.1.

4 Metodología

La metodología usada en este proyecto, ha seguido un formato ágil o "Agile". Al realizar este trabajo en un entorno de grupo, la metodología ágil permite una versatilidad y ejecución, que está orientada a la consecución de objetivos. Así, logrando un desarrollo incremental y flexible, recolectando y analizando preliminarmente los resultados obtenidos durante y posteriormente su realización. Consiguiendo una transparencia, en la obtención y cumplimientos de objetivos, gracias a las reuniones semanales con el tutor de la empresa y de la universidad. Permitiendo una evolución y desarrollo del proyecto de forma eficiente y ágil, teniendo en cuenta la posible aparición de problemas y su corrección y resolución.

La metodología se podría dividir en estas partes:

1. Recopilación de información.
2. Generación a una solución, al objetivo a cumplir.
3. Análisis de los resultados obtenidos.
4. Recopilación y comprobación de los resultados, con sus respectivas posibles mejoras.

5 Búsqueda de Herramientas AutoML

Al haber comenzado y realizado la base del proyecto, se dio principio a la idea de cómo plantearse el cómo desarrollar y usar el AutoML, se efectuó una investigación y recopilación de información sobre el sector y plataformas que proporcionan esta funcionalidad. Al descubrir la existencia de varias opciones y plataformas, el objetivo era filtrar y seleccionar dos para el proceso de desarrollo del proyecto, teniendo presente la necesidad y compatibilidad de la plataforma con el NLP. Como el origen y composición de los datos que se usan en este proyecto son correos electrónicos, es necesaria que esta funcionalidad esté presente en la plataforma AutoML.

Se seleccionó para investigar e indagar en estas principales soluciones del mercado: H2O AutoML, Microsoft Azure AutoML, Amazon SageMaker Autopilot, DataRobot y Google Cloud AutoML.

5.1. H2O AutoML, Microsoft Azure AutoML y Amazon SageMaker Autopilot

H2O AutoML, Microsoft Azure AutoML y Amazon SageMaker Autopilot son tres grandes plataformas que han estado ofreciendo un conjunto de productos de ML, como parte de sus servicios de computación en la nube durante algún tiempo.

Se descártaron estas plataformas, debido a que el uso NLP no están bien definidos dentro de las plataformas, y parece de difícil implementación, no está implementado o aún no son del todo compatibles, estando en una fase muy temprana.

5.2. DataRobot

DataRobot lanzó su solución AutoML a finales de 2015 lo que la convierte en una de las primeras soluciones empresariales de AutoML. La solución se centra principalmente en el desarrollo y la implementación rápidos del modelo, así como en la facilidad de uso.

En cuanto a sus principales características, la funcionalidad de DataRobot aporta un proceso automatizado, con selección de modelos y funcionalidades de aprendizaje. Proporciona una tabla de clasificación de soluciones candidatas donde los usuarios pueden confiar en un gráfico de velocidad frente a precisión para elegir la solución deseada mientras se mantiene un equilibrio entre la eficiencia y la complejidad del modelo.

Posee una versatilidad que puede abordar el tipo de clasificación, regresión y "forecasting" de problemas ML. Además, se admiten problemas que involucran datos de texto estructurados, así como datos de texto no estructurados NLP.

Es una solución empresarial y requiere una licencia comercial. Ofrece una GUI (interfaz gráfica de usuario) limpia para que los usuarios interactúen con la solución. También se proporciona una API de cliente Python. Se puede instalar en las instalaciones, así como en la mayoría de las principales plataformas en la nube.

Su pago y suscripción, te permiten un periodo de prueba limitado a 30 días y a 500 dólares de uso.

5.3. Google Cloud AutoML

Google Cloud AutoML se introdujo a principios de 2018, para ofrecer capacidades de aprendizaje automático automatizadas, principalmente, a los que no son expertos en ML. La versión inicial se limitó solo a aplicaciones de visión. Los usuarios pueden aprovechar los modelos prediseñados de propiedad de Google, tal cual, mediante API simples.

Como principales características, centrándose en su funcionalidad, proporciona un servicio de etiquetado de datos, para ayudar con el paso de preparación de datos del flujo de trabajo de ML. Para los datos no estructurados, los servicios de Google AutoML Vision, NLP y Video se basan principalmente en redes neuronales profundas. Además, se realiza la selección y el aprendizaje automáticos del modelo. Se ofrecen varias métricas de rendimiento del modelo y gráficos para ayudar a los usuarios a realizar la validación del modelo y elegir un modelo final.

Su versatilidad le permite resolver problemas de clasificación y regresión que involucran texto no estructurado, imágenes e incluso datos de video, así como datos estructurados. Los modelos de ML que se utilizan para datos no estructurados, se basan principalmente en redes neuronales profundas.

Es una solución empresarial en la nube en la que los usuarios pagan por los servicios que utilizan a lo largo del tiempo. Tiene una interfaz de usuario web y es relativamente fácil de administrar y usar. Las bibliotecas de API de cliente se proporcionan en varios lenguajes, incluidos Python, Java, PHP y más.

En cuanto a su precio y suscripción, Google ofrece durante 90 días y con 300 dólares para el uso en Google Cloud, el uso de su plataforma de forma gratuita.

5.4. Selección

Las dos últimas plataformas introducidas, DataRobot y Google Cloud AutoML, fueron las herramientas AutoML seleccionadas para su uso y desarrollar este proyecto. Como es explicado anteriormente en sus apartados, ofrecen unas soluciones bastantes completas y una mayor madurez en la sección de datos no estructurados NLP.

6 DESARROLLO AUTOML

La fase de desarrollo del proyecto, tiene como meta la explicación y demostración de la realización de las diferentes etapas seguidas a lo largo del mismo. Estará compuesto por la explicación y uso de las plataformas seleccionadas, y el inicio del proceso de ejecución para obtener los modelos y sus resultados. Hay que tener en cuenta, las dos plataformas han sido utilizadas en sus versiones gratuitas.

6.1. Datos Utilizados

Para la ejecución y creación de los diferentes modelos con el uso del AutoML, este proyecto tiene como origen de datos una recolección de correos electrónicos. Estos son recolectados y contienen información privada por parte de sus usuarios.

El objetivo del uso y la creación de los modelos a partir de estos datos, es poder clasificarlos y asignarlos a una categoría de correo electrónico, generando así un orchestrator (permite automatizar la creación, supervisión e implementación de recursos en su entorno) de correos, automatizando el proceso de ordenación y permitiendo obtener información de los mismos.

6.2. Generación de los Datasets: Generador de emails

En la generación de los Datasets, para poder respetar el anonimato y privacidad de los datos, se procedió a crear un código Python para la creación automática de correos electrónicos. Este código tiene como entrada un archivo

CSV con los correos proporcionados y otro CSV con las palabras claves y su clasificación. Con esto, primero se crean los "dataframes" correspondientes y se aplica una limpieza de datos, obviando los "tags html", caracteres especiales y de puntuación. Convirtiendo estos "dataframes" en listas, mediante bucles y de forma aleatoria, se generan los cuerpos de los correos, añadiendo primero una parte no relevante de palabras, posteriormente se añade la palabra clave, y por último, se vuelve a añadir otro conjunto de palabras no relevantes. Así, se generan correos aleatorios que contienen todo un significado y se pueden clasificar. Al final del código se genera un archivo CSV con todos los correos generados.

Se dispone de una serie de parámetros implementados en el programa para modificar la creación de los emails, como el número de correos electrónicos a generar por categoría o la longitud fija o variable de las palabras de los mismos dentro de un rango. También, la limitación de categorías a generar dentro del "Dataset", de las cuales hay un máximo 41 categorías en las que se puede clasificar un correo.

Si se desea ver el código completo, está desarrollado en lenguaje Python y se puede consultar todo el código en el [Apéndice A.2].

6.3. Toma de contacto: DataRobot

Con la creación de unos "Datasets" robustos y preparados, se procedió a acceder a la plataforma de AutoML en DataRobot. Se realizó por medio de la versión "trial", se trata de una versión gratuita limitada en varias capacidades y opciones, además de que toda la realización fue por medio de la plataforma y servicios web de DataRobot. El proceso para el uso de la plataforma que se sigue, es el del ciclo de vida de AutoML, desde que se introduce el archivo CSV de correos generado hasta la creación del modelo y su predicción de datos (se puede apreciar el ciclo en la anterior figura 3). Entrando en más detalle, son los siguientes:

6.3.1. Interacción

Toda esta metodología comienza con la fase de "Interaction", que premia que la herramienta sea programable y cuente de una interfaz web. DataRobot mediante su versión web, no hay forma de poder programar o automatizar. En la versión completa, existe "DataRobot Python Client", que es una API en Python que permite el uso de DataRobot en "On-Premise". Al ser una página web, tiene una interfaz muy bien adaptada y de fácil uso que permite interactuar sin ninguna necesidad de usar código.

6.3.2. Data Ingestion

En esta etapa de "Data Ingestion" o ingestión de datos, se tiene en cuenta el formato de los datos y origen de los mismos. DataRobot se centra en el proceso AutoML, no permite una limpieza apropiada de los datos. La "Data Ingestion" es preferible realizarla parcialmente o completamente antes.

6.3.3. Data Processing

El "Data Processing" o el procesamiento de datos, tiene como enfoque, ver si los datos son soportables, la limpieza de los mismos y realizar el "feature engineering", "data splitting" y "labelling" automáticamente. DataRobot soporta texto, imágenes y tablas. Estos pueden ser importados mediante varios tipos de archivos.

En la carga de datos, implementa de forma automática una limpieza ligera de los datos (p. ej. valores nulos). Muestra y aporta información de los datos realizando por medio de un EDA ("Exploratory Data Analysis"). También, permite separar los datos por diferentes métodos: CV ("Cross Validation") y T-V-H ("Train-Validation-Holdout") y, soporta el uso de "Labelling" o etiquetas para indicar cuál es el objetivo a predecir.

6.3.4. Data Visualization

En la fase de "Data Visualization" o visualización de los datos, DataRobot muestra estadísticas de los datos y su visualización. Permite analizar mediante gráficos, tablas y esquemas diferentes estadísticas de los "Datasets".

6.3.5. Algorithm Specification

En la etapa de "Algorithm Specification" o especificación de algoritmo se realiza la autoselección del modelo, creación del modelo, "hyperparameter tunning" y búsqueda automática de Redes Neuronales. DataRobot implementa de forma automática todas estas funciones en el proceso de AutoML.

6.3.6. Model Training & Evaluation

En el "Model Training & Evaluation", el AutoML de DataRobot realiza un "Model Versioning", iterativamente donde hace ajustes del "training", calcula el "Loss Function" y proporcionando una transparencia, distribución y paralelismo en el "training" de los modelos. También realiza "transfer learning", métricas automáticas y genera una explicación de cada modelo que calcula.

Para realizar el entrenamiento, existe en DataRobot cuatro formas de ejecución: Autopilot, Quick, Comprehensive y Manual. Estos varían en la exhaustividad del cálculo de los modelos, el número de los mismos aplicados en los datos y el tiempo de ejecución. El proceso de "Training" puede ser totalmente paralelo, mediante el uso de las CPU que proporciona DataRobot, así pudiéndose ejecutar varios modelos simultáneamente.

6.3.7. Deployment

En la fase de "Deployment" se ejecuta el despliegue automático del modelo y portabilidad del Modelo: El despliegue no es automático, al finalizar el proceso de AutoML se ha de seleccionar un modelo a desplegar. En el caso de la versión utilizada (Trial), solo se puede desplegar en MLOps.

6.3.8. Model Prediction

Con el paso de "Model Prediction" se busca que se ejecuten los modelos online o en batch, que proporcione "Functional Monitoring" y el formato en que se puede usar el "Prediction Data". DataRobot contiene un apartado de

MLOps, donde se puede desplegar los modelos Online y configurar "Deployments" y "Jobs" para la creación de ejecuciones batch. También contiene una "Prediction API" (versión completa). Se pueden monitorizar todos los despliegues con sus ejecuciones por medio de la "Overview tab", que contiene diferentes datos y estadísticas para el control.

6.4. Toma de contacto: Google Cloud AutoML

Haciendo uso de los Datasets usados en DataRobot, se procedió al uso del AutoML en Google Cloud AutoML. Teniendo presente que se utilizó la versión gratuita de prueba, se procederá a la explicación del ciclo de vida y funcionamiento de Google Cloud AutoML de forma más reducida que DataRobot, ya que al ser un proceso de AutoML siguen proceso global compartido y similar, enfocándonos las características de la herramienta de Google:

6.4.1. Interacción

Mediante la versión web en el apartado de "Tablas" (usado con nuestros datos), no hay forma de poder programar o automatizar directamente. Pero dentro de Google Cloud, hay otros apartados, como "Cloud Functions", donde dentro del mismo navegador se puede crear funciones y llamadas a modelos ya creados. Además, Google aporta APIs para la realización de llamadas a la nube localmente. Al ser una página web, tiene una interfaz muy bien adaptada y de fácil uso que permite interactuar sin ninguna necesidad de usar código.

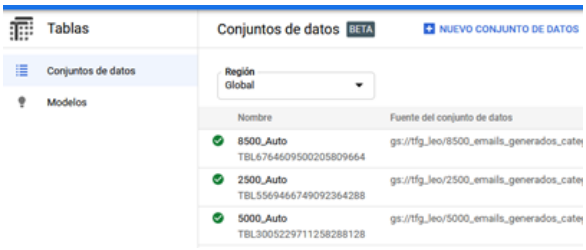


Fig. 4: Interfaz de inicio de la función "Tablas" de AutoML.

6.4.2. Data Ingestion

Accediendo en el apartado de "Tablas", existen dos partes, donde la primera de ellas son los conjuntos de datos que se han subido [Fig. 4] y la segunda, son los modelos creados. En esta sesión, Google se centra principalmente en la preparación de los datos, para su uso en los modelos en el proceso AutoML, la "Data Ingestion" es preferible realizarla parcialmente o completamente antes de subir en la plataforma. Solo se realiza la comprobación de la compatibilidad de los datos.

6.4.3. Data Processing

El procesamiento de datos de Google soporta texto, imágenes y tablas en diferentes apartados de IA presentes en la plataforma Cloud. Centrándonos en "Tablas", Estos pueden ser importados mediante 2 tipos de procesos: importando archivos tipo CSV o usando la herramienta de "BigQuery".

En la carga de datos, la plataforma carece de muchas funcionalidades estando casi vacía. Realiza un EDA (Exploratory Data Analysis) muy básico, permite separar los datos por medio de "Train-Validation-Test" y soporta, el uso de "Labelling" o etiquetas para indicar cuál es el objetivo a predecir [Fig. 5].

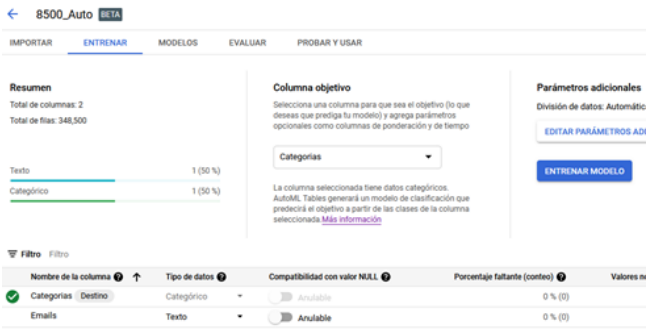


Fig. 5: Sección de Data Processing, previa a la creación de modelos.

6.4.4. Data Visualization

En la fase de "Data Visualization", Google muestra un breve resumen de los datos, careciendo de mayor profundidad en este apartado.

6.4.5. Algorithm Specification

En la etapa de "Algorithm Specification" se realiza la autoselección del modelo, creación del modelo, "hyperparameter tunning" de forma cerrada poca transparente, no aportando información sobre el proceso.

6.4.6. Model Training & Evaluation

En la formación y evaluación de modelos, el AutoML de Google realiza una ejecución "Model Versioning", iterativamente hace ajustes del "Training", calcula el "Loss Function". Este proceso está oculto y no es mostrado al usuario. No se explica que proceso ni que modelos son ejecutados, demostrando una poca transparencia. Para realizar el entrenamiento, existe una selección de la distribución de la separación de los datos y la forma ejecutarlo, es mediante la indicación previa del tiempo aproximado para el proceso de AutoML.

El proceso de "Training" solo puede ser individual, no paralelo, y Google indica que hacen uso de CPU y GPU propios, en los procesos de entrenamientos de modelos.

Al final de la ejecución del proceso AutoML, surgen un análisis y evaluación de los resultados obtenidos del modelo creado [Fig. 6].

6.4.7. Deployment

En la fase de "Deployment" o despliegue, no es automático, al finalizar el proceso de AutoML se ha de seleccionar un modelo a desplegar. Pero una vez desplegado, la versión

Fig. 6: Sección de evaluación de un modelo creado.

utilizada no te limita en funcionalidades y permite el despliegue y predicción "online", exportarlo a un proyecto de "BigQuery" o el uso de sus APIs para explotar los modelos creados.

6.4.8. Model Prediction

En el "Model Prediction", Google contiene un apartado de "BigQuery", donde se pueden crear proyectos y desde donde se puede obtener un conjunto de datos de entrada, predecirlos y el resultado redirigirlo automáticamente otra vez a "BigQuery", permitiendo desplegar los modelos Online y realizar predicciones por lotes.

7 Resultados AutoML

En este apartado se procederá a la explicación detallada de la ejecución, creación y obtención de los resultados de las ambas plataformas de AutoML utilizadas, extrayendo unas características y conclusiones de cada una.

7.1. Datasets utilizados

Se han creado tres Datasets de diferentes tamaños para la realización de los diferentes entrenamientos. El primero de ellos contiene 2500 muestras por categoría de correos, las cuales son 41, eso indica que hay un total de 102500 correos generados en el Dataset. El segundo contiene un total de 205000, los cuales se distribuyen en 5000 por categoria. El último y tercero, está compuesto por 8500 casos de correos, generando un total de 348500 emails. Todos estos contienen dos "feature", donde la primera es un campo de texto que contiene los correos, y el otro, posee la clasificación del correo de la misma fila, para el aprendizaje supervisado del modelo.

También, se ha creado un Dataset de prueba que contiene 1000 correos, para el apartado de "test" y así, predecir sus categorías y poder compararlas con las originales y lograr evaluar los modelos generados con el mismo conjunto de datos.

7.2. Resultados obtenidos

En esta fase del proyecto se procederá a mostrar y explicar los resultados recibidos al momento de realizar el proceso de AutoML en ambas plataformas.

7.2.1. DataRobot

Se han realizado y seleccionado finalmente 9 combinaciones de ejecuciones, para representar las capacidades de AutoML en el campo de NLP de DataRobot. Se ha hecho uso de los tres Datasets (2500, 5000 y 8500 muestras por categoría) para combinarlos con las opciones que aporta la plataforma para entrenar los modelos. Se toma en cuenta el uso de las diferentes variantes exhaustividad de la ejecución y la modalidad de "Data Splitting" o particionado de los datos para el AutoML.

Dentro del particionado del Dataset, en los casos de CV ("Cross Validation"), se hace uso de 5 folds o pliegues y un de 20 % test. Se realizaron pruebas con diferentes números de folds, como 10 o 20, pero se descartaron al ver que los resultados no tenían repercusión en los resultados, mostrando ni un 1 % de variación en el "Accuracy" o precisión del modelo comparado con la configuración inicial. En los casos de TVH (Training-Validation-Holdout), se probaron dos configuraciones en la distribución: 80 % training, 10 % validación y 10 % testing (Holdout), y la segunda: 60 %, 20 %, 20 % correspondientemente. También, en cuanto a la optimización de métricas, DataRobot permite el uso de varias. Se hicieron pruebas con las tres principales, Accuracy, AUC y Log Loss, pero no se apreciaron cambios sustanciales en el aprendizaje de los modelos, apostando finalmente por la predeterminada, Log Loss (pérdida logarítmica).

En cuanto a modos de ejecución, se eligieron los 3 principales: Quick, Autopilot y Comprehensive.

En la siguiente [Tabla 1], se puede apreciar los resultados obtenidos:

Parám. entrada al modelo			**Accur**	**AUC**	**Log L.**
Dat.	**Ejec.**	**Data Split.**			
2500	Autop.	CV	0,1977	0,778	3,144
2500	Autop.	TVH: 80 %	0,1986	0,779	3,104
5000	Autop.	CV	0,2013	0,793	3,055
5000	Compr.	CV	0,1981	0,810	2,902
5000	Quick	CV	0,1607	0,730	3,278
5000	Autop.	TVH: 60 %	0,2025	0,790	3,055
5000	Autop.	TVH: 80 %	0,2057	0,797	3,016
8500	Autop.	CV	0,2491	0,836	2,799
8500	Autop.	TVH: 80 %	0,2569	0,844	2,770

TABLA 1: Resultados obtenidos de los modelos de la plataforma DataRobot.

La repercusión de las modalidades de ejecución se puede apreciar en la siguiente tabla [Tabla 2]:

Parámetros		**Log Loss**	**Tiempo aprox. ejec.**
Dat.	**Ejec.**		
5000	Quick	3,2784	15 mins
5000	Autopilot	3,0552	30 mins
5000	Compreh.	2,902	1h 30 mins

TABLA 2: Certeza y tiempos de ejecución asociados con los modos de AutoML de DataRobot.

Observando los resultados de DataRobot [Tabla 1], se aprecia un bajo nivel de aprendizaje en los modelos. Esto puede ser ocasionado por las limitaciones de la versión Trial en la carga de datos o también, otro factor y más propenso a ser, una baja compatibilidad del proceso AutoML en este caso de NLP.

Obviando las puntuaciones, cabe remarcar el uso de los diferentes modos de ejecución afectan considerablemente a los resultados, teniendo en cuenta que el tiempo de ejecución entre ellos varía bastante [Tabla 2]. Y también, el correcto uso del "Data Splitting" genera un cambio positivo en el aprendizaje de los modelos haciendo uso del Dataset con más muestras por categoría.

7.2.2. Google Cloud AutoML

La realización en Google Cloud AutoML es más limitada y cerrada e incluso oculta. En el caso de los Datasets usados, Google limita el proceso a unos parámetros predeterminados, sin opción a modificar o configurar la ejecución del AutoML. Solo se permite un particionamiento, del 80, 10, 10 de los datos, además de solo un tipo de optimización de métricas que es el Log Loss.

El conjunto de pruebas realizado se basa en el uso de los tres Datasets anteriormente utilizados [Tabla 3]:

Parámetros entr. al modelo			**Accur.**	**AUC**	**Log L.**
Dat.	**Ejec.**	**Data Split.**			
2500	Predet.	TVH: 80 %	0,6960	0,805	3,053
5000	Predet.	TVH: 80 %	0,6520	0,806	3,116
8500	Predet.	TVH: 80 %	0,7347	0,862	2,689

TABLA 3: RESULTADOS OBTENIDOS DE LOS MODELOS DE LA PLATAFORMA GOOGLE CLOUD AUTOML.

Google demuestra una baja capacidad de personalización y configuración de parámetros, pero en el ámbito del aprendizaje, se aprecia un gran nivel de comprensión en el NLP y aporta unos buenos resultados en los modelos creados.

7.3. Conclusiones AutoML

En este apartado se concluirá y explicará el rendimiento de ambas plataformas, además de también describir el mejor modelo concebido a lo largo de la fase de desarrollo. Estas conclusiones sobre el AutoML contienen 3 partes: usabilidad y funcionalidad, comparativa de resultados y explicación del mejor modelo.

La usabilidad y funcionalidades encontradas en DataRobot, demuestran la madurez y el mayor número de años que posee la plataforma sobre la de Google:

1. Permite un alto nivel de configuración.
2. Genera descripciones y reportes explicativos para todos los modelos automáticamente.
3. Muestra un alto nivel de transparencia en todos los pasos en el proceso de ejecución.
4. Incluye una avanzada plataforma de monitorización que provee varias métricas.
5. La herramienta permite control y aporta bastante información sobre las features.
6. Soporta ejecución de modelos de forma paralela.

Google posee el proceso básico para la generación de modelos, no aportando información y con poca configuración, convirtiéndolo en una caja negra.

Procediendo con la comparativa de los resultados, si anteriormente Google se quedaba rezagada en comparación con DataRobot, en este caso es lo contrario. Google Cloud AutoML ha sido capaz de crear y obtener mejores modelos y resultados [Tabla 4]. Demostrando en esta casuística un mayor nivel de comprensión de los correos electrónicos y un NLP mejor implementado.

Caract. del modelo		**Tpo. ejec.**	**Log L.**	**Accur.**
Dat.	Plataforma			
2500	Google	1 hora	3,0530	0,6960
2500	DataRobot	20 min	3,1041	0,1986
5000	Google	1 hora	3,1160	0,6520
5000	DataRobot	30 min	3,0161	0,2057
8500	Google	1 hora	2,6890	0,7347
8500	DataRobot	50 min	2,7707	0,2569

TABLA 4: COMPARATIVA RESULTADOS OBTENIDOS DE LOS MODELOS DE DATAROBOT Y GOOGLE CLOUD AUTOML.

Esta comparativa es equitativa e igualitaria, esto es debido al mismo uso de parámetros y Datasets en ambas partes. Se ha utilizado los métodos predeterminados de ambas plataformas, los resultados son medidos con un mismo umbral o "Threshold" de 0,50 y con un mismo "Data Splitting" de TVH (80/10/10).

En la Tabla 4, se puede apreciar mejores resultados en las versiones de Google, obteniendo una mayor precisión al momento de predecir y clasificar los correos. Google aporta como resultado sobresaliente, el modelo de 8500 categorías por muestra, donde su precisión es alrededor de 73 % y Log Loss es de 2,6890. Por parte de DataRobot, el modelo más destacado es también el de 8500, pero con solo un 26 % de precisión y una pérdida logarítmica de 2,7707.

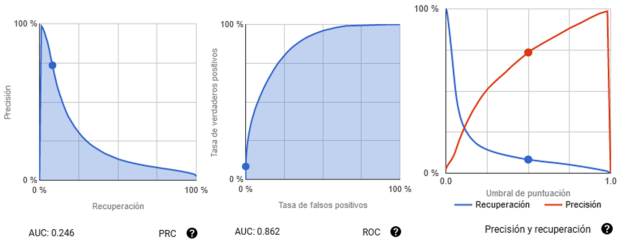
En cuanto a tiempos de ejecución, DataRobot muestra una metodología con mayor velocidad, donde el proceso de AutoML termina automáticamente con la finalización del aprendizaje de los modelos. En contra parte, Google exige previamente al proceso de AutoML, la indicación del tiempo total de la ejecución por parte del usuario, en estos casos de una hora.

Realizando un enfoque con más detalle al 8500 de Google, el modelo con más puntuación, se procederá a mostrar información y gráficos sobre los resultados obtenidos.

En cuanto a métricas, el modelo ha obtenido una precisión del 73,5 %, una recuperación del 8,3 %, una pérdida logarítmica del 2,689 y un área debajo del ROC del 0.862 (definiciones de las métricas utilizadas en [Apéndice A.3]). Estos valores pueden verse representados en las siguientes figuras:

Los puntos marcados dentro de las gráficas, indican en valor actual del modelo al estar con un umbral del 0,5.

En la figura 7, la imagen (a), la curva de precisión-recuperación (PR) muestra la compensación entre estos dos valores en distintos umbrales de clasificación. Un umbral más bajo genera una mayor recuperación, pero la precisión es, por lo general, más baja. En cambio, un umbral más alto genera una recuperación menor, pero la precisión suele ser mayor.



(a) Precisión. (b) Curva ROC. (c) Precisión y recuperación.

Fig. 7: Gráficos resultados métricas.

En la figura 7, la imagen (b), la curva de característica operativa del receptor (ROC) muestra la compensación entre la tasa de verdaderos positivos y la tasa de falsos positivos. Un umbral más bajo genera una tasa más alta de verdaderos positivos (y una tasa de falsos positivos más alta). En cambio, un umbral más alto genera una tasa de verdaderos positivos más baja (y una tasa de falsos positivos más baja).

Y, por último, en la figura 7, la imagen (c), un umbral de puntuación más alto produce menos falsos positivos, lo cual aumenta la precisión. Un umbral de puntuación más bajo produce menos falsos negativos, lo cual aumenta la recuperación.

Para finalizar, remarcar que la realización del proyecto desde cero, con la preparación del Dataset hasta el punto de la obtención del modelo del AutoML, el tiempo es de alrededor de una 1 semana o 168 horas.

8 MLOps

Una vez finalizado el estudio y análisis del AutoML, en este bloque del proyecto, se realiza las mismas actividades, pero con el foco en el desarrollo tradicional de ML, basado en la metodología MLOps.

La creación de modelos con su correspondiente transición a un formato mayor, con problemas e implementaciones superiores y más profesionales, conlleva a una gran dificultad en cuanto a metodología, ya que, pese a todos los estudios y resultados realizados, la elección de las arquitecturas de los modelos, se guían más por la experiencia e incluso por el ensayo y error, que por una teoría sólida. Esto es apreciable notablemente, en las redes neuronales de "Deep Learning", el número y tipo de capas, las formas de aprendizaje y activación, entre otras cosas, aportan una complejidad alta a los problemas. A pesar de ello, existe una cierta metodología de trabajo, llamada MLOps, que ayuda en cómo afrontar un proyecto o problema de ML.

Si es necesario consultar o obtener mayor información sobre el modelo de trabajo o ciclo de vida, que se sigue de una manera más o menos generalizada MLOps, se puede consultar en el [Apéndice A.4].

8.1. Caso de Desarrollo MLOps

En el caso de un proyecto, uno que puede ser aplicado normalmente en empresa del sector bancario típica, en este apartado se explica cómo es el procedimiento seguido para hacer un clasificador de correos "tradicional". Los datos y las ideas empleadas en el AutoML, provienen de este mismo proyecto y datos. El desarrollo explicado anteriormente, describe de forma correcta y general el proceso que se siguió al resolver el problema. Gracias a eso, se llegó a la solución que se explicará a continuación (la definición de varios de los próximos términos está presentes en [Apéndice A.4]):

Este procedimiento empieza con una limpieza del texto de los correos. Primero, se realiza un "Cleaning" de los datos, de las letras, acentos, caracteres e información redundante. Segundo, la información de poca importancia o poco informativas, se convierten en una expresión regular y así se normalizan. Y al final de esta fase, se buscan "Stop Words" y puntuaciones en el texto y se eliminan.

Como segundo fase, una vez con los datos depurados, se continúa con la masterización de los mismos. Esto se realiza mediante dos formas distintas según el algoritmo base que se utiliza. Los datos a utilizar en una FCNN ("Fully Convolutional Neural Network"), siguen una operación TFIDF ("Term Frequency-Invers Document Frequency"). Se busca la frecuencia de las palabras que contienen los correos y se filtran las más utilizadas. Los datos para los LSTM ("Long Short-term Memory"), también se filtra por la frecuencia, pero de forma menos restrictiva. Con estos datos se crean los diccionarios para cada caso.

Como tercera fase, se realiza la vectorización del texto. Donde para la FCNN, se representan las palabras por el conteo de las mismas y para la LSTM, dentro de la misma red neuronal, existe una capa de "embedding" o incrustación. El resto de variables se les realiza One-hot Encoding. Por último, se aplican los modelos, que son tres: un FCNN y dos diferentes LSTM. Los resultados de clasificación para los correos son el promedio simple de las probabilidades de las categorías de entre las tres redes neuronales.

8.2. Resultados MLOps

Los resultados obtenidos con la realización de todo el procedimiento son los siguientes: un "recall" o recuperación global del 38,8 %, una precisión del 82,5 % y 6,5 % de errores. Esto demuestra un modelo bastante fuerte y con una precisión solvente para la clasificación de los correos para las diferentes categorías.

El tiempo de realización del proyecto, desde cero hasta este punto, es de alrededor de unas 6 semanas o 1000 horas, con dos personas trabajando como mínimo.

9 Comparativa AutoML y MLOps

Los desarrollos de modelos en AutoML y MLOps son bastante eficaces al momento de enfrontar un problema donde es necesario el uso de ML. Debido a la explicación y realización de ambas opciones, representadas en los anteriores apartados, se procederá a la comparación y contraste de las mismas:

Comenzando por los conceptos, podemos definir a AutoML como una solución principalmente empresarial, pero que también utilizable por cualquier usuario, donde se busca un compromiso entre la efectividad, velocidad y facilidad

de uso a lo largo del proceso de desarrollo de un proyecto. También tiene un menor costo comparado con MLOps. Por otra parte, MLOps es una metodología y solución totalmente empresarial y de alto nivel, que no permite a muchas empresas pequeñas usarla debido a su complejidad y costes de implementación. Se busca y prioriza, la precisión, calidad y robustez de los modelos a creados. AutoML intenta adaptar toda esta metodología y automatizarla.

Adentrándose en las opciones de plataformas de AutoML, los ejemplos de DataRobot y Google demuestran diferentes compromisos y objetivos de usuarios. DataRobot se centra en una opción totalmente empresarial, asumiendo que sus usuarios tienen conocimientos en el sector, permitiéndoles un mayor control sobre la creación de los modelos y aportándoles información y descripciones más detalladas y técnicas. Google tiende a simplificar todo el procedimiento, tanto que, no se obtiene ningún tipo de información del proceso desde que se ingresan los datos, hasta que se crea el modelo, del cual se desconoce cuál algoritmo adoptó. Eso, permite a sus usuarios resolver problemas solventemente, sin tener ningún conocimiento de creación de modelos de ML, de forma automática.

Concluyendo, AutoML comparado con MLOps, demuestra unos resultados bastante capaces, y en el caso actual de un orquestador de correos, su principal diferencia es el tiempo de desarrollo. La opción creada por medio de MLOps, el tiempo para la realización desde cero hasta obtener un modelo útil, es bastante superior. La obtención de una solución de un modelo similar en AutoML, se realiza en bastante menor tiempo.

A continuación, se puede apreciar y comparar el rendimiento de ambas opciones en la siguiente tabla [Tabla 5]:

	AutoML	**MLOps**
Precisión	0,735	0,825
Recuper.	0,083	0,388
Tiempo	1 semana	6 semanas
Costo	Bajo costo.	Costes superiores de infraestructuras y capital humano.

TABLA 5: COMPARATIVA SOLUCIONES AUTOML Y MLOPS.

10 CONCLUSIONES

El proyecto surgió con el objetivo de realizar una búsqueda de una solución más ágil y eficaz a un problema propuesto de ML, realizando un planteamiento y desarrollo de forma más rápida, ahorrando tiempo y costes. Todo esto sería capaz mediante una nueva herramienta llamada AutoML, la cual prometía cumplir con creces estas metas. A lo largo del desarrollo de este proyecto, se ha podido apreciar que el AutoML es una herramienta todavía muy joven, pero con un gran potencial. Comparándola, con la opción utilizada actualmente por los "Data Scientists" y empresas, MLOps, demuestra su eficacia y capacidades, como la relación entre el tiempo de ejecución y resultados que son excelentes, pero también, muestra ciertas limitaciones. La falta de adaptabilidad y configuración a ciertos casos o la precisión y detalle existentes en MLOps, la cual al ser desarrolla desde cero y al gusto del usuario puede adaptarse mejor a los problemas que se deseen resolver.

Para concluir, AutoML no ha llegado para substituir o reemplazar a las personas y metodologías actuales, todavía está en una fase temprana y de maduración. Pero, como herramienta de apoyo al resto, puede aportar agilidad y mayor comodidad a la hora de comenzar un nuevo proyecto de ML. Habiendo investigado profundamente sus opciones y capacidades, si se buscan resultados no excesivamente precisos y una alta velocidad en el desarrollo, su viabilidad, competencias y rentabilidad, comparado con un método manual de ML, son superiores en muchos casos. Solo queda esperar a ver cómo evoluciona este sector en los próximos años.

AGRADECIMIENTOS

Primero, me gustaría agradecer a mis tutores, por parte de la universidad, a Carlos García Calvo, por guiarme y asesorarme constantemente en todo el proceso y en las diferentes entregas. También gracias a mis tutores de la empresa, Pol Rojas y Diego González, por permitirme trabajar con ellos, apoyarme constantemente en el desarrollo, confiar y darme la oportunidad de realizar este proyecto.

También agradecer a mi madre Yezmin, a mi padre Leopoldo y a mi hermano Diego, por la ayuda y soporte que me han dado todos estos últimos años. Como también, a todos los amigos que se han interesado por mí a lo largo del proyecto.

Y por último, pero muy importantes, dar la gracias a Leopoldo Vázquez, Olga León y Luis López Díaz, por su apoyo, amor y confianza en mí. Les dedico este trabajo.

REFERENCIAS

[1] Jesús, ¿Qué es AutoML?, DataSmarts, 2020. [Online]. Disponible en: https://datasmarts.net/es/que-es-automl/. [Accedido: 21-feb-2021]

[2] Ciencia de datos, Wikipedia. [Online]. Disponible en: https://es.wikipedia.org/wiki/Ciencia_de_datos. [Accedido: 21-feb-2021]

[3] Maricela Ochoa, Qué es Automated Machine Learning: la próxima generación de inteligencia artificial, Itmastersmag, 25-sep-2019. [Online]. Disponible en: https://itmastersmag.com/noticias-analisis/que-es-automated-machine-learning-la-proxima-generacion-de-inteligencia-artificial/. [Accedido: 21-feb-2021]

[4] Aprendizaje automático. Qué es y por qué es importante, SAS. [Online]. Disponible en: https://www.sas.com/es_es/insights/analytics/machine-learning.html. [Accedido: 28-feb-2021]

[5] 'Machine learning': ¿qué es y cómo funciona?, BBVA, 08-nov-2019. [Online]. Disponible en: https://www.bbva.com/es/machine-learning-que-es-y-como-funciona/. [Accedido: 28-feb-2021]

[6] Emérita Legal, El Machine Learning cambiará el mundo, Emérita Legal, 30-ago-2018. [Online]. Disponible en: https://www.emerita.legal/blog/innovacion-legal/el-machine-learning-cambiara-el-mundo/. [Accedido: 28-feb-2021]

[7] Gunjit Bedi, Auto Text Classification using Google's AutoML, Medium, 22-jun-2019. [Online]. Disponible en: https://medium.com/voice-tech-podcast/auto-text-classification-using-googles-automl-80f151ffa176. [Accedido: 05-mar-2021]

[8] Bahador Khaleghi, A critical overview of AutoML solutions, Medium, 02-abr-2020. [Online]. Disponible en: https://medium.com/analytics-vidhya/a-critical-overview-of-automl-solutions-cb37ab0eb59e. [Accedido: 05-mar-2021]

[9] Bahador Khaleghi, Why enterprise machine learning is struggling and how AutoML can help, Medium, 02-abr-2020. [Online]. Disponible en: https://medium.com/swlh/why-enterprise-machine-learning-is-struggling-and-how-automl-can-help-8ac03023bf01. [Accedido: 05-mar-2021]

[10] Procesamiento de lenguajes naturales, Wikipedia. [Online]. Disponible en: https://es.wikipedia.org/wiki/Procesamiento_de_lenguajes_naturales. [Accedido: 05-mar-2021]

[11] ¿Qué es MLOps? [Guía completa], Keep coding. [Online]. Disponible en: https://keepcoding.io/blog/que-es-mlops/. [Accedido: 12-abr-2021]

[12] Dragon Nomada, Flujo de Trabajo de Machine Learning, DataRobot, 17-abr-2021. [Online]. Disponible en: https://medium.com/inside-intelligence/flujo-completo-de-machine-learning-95a1c8219296. [Accedido: 29-abr-2021]

[13] Ignacio G.R. Gavilán, Un flujo de trabajo universal para Machine Learning, DataRobot, 06-may-2020. [Online]. Disponible en: https://ignaciogavilan.com/un-flujo-de-trabajo-universal-para-machine-learning/. [Accedido: 29-abr-2021]

[14] Red neuronal convolucional, Wikipedia. [Online]. Disponible en: https://es.wikipedia.org/wiki/Red_neuronal_convolucional. [Accedido: 19-may-2021]

[15] Long short-term memory, Wikipedia. [Online]. Disponible en: https://en.wikipedia.org/wiki/Long_short-term_memory. [Accedido: 10-may-2021]

APÉNDICE

A.1. Planificación

El desarrollo de este proyecto constó de tres fases bien diferenciadas. En la primera fase donde se investigó y estudió el tema tratado, haciendo una búsqueda de los conceptos necesarios. También, se buscaron las opciones disponibles para el desarrollo del proyecto, realizando un estudio de las herramientas y plataformas disponibles. Así, logrando la selección de las herramientas que se hicieron uso a lo largo del proyecto. En la segunda fase, se procedió a aprender, utilizar y crear todo el apartado de la creación de modelos para la predicción de los Datasets. Por último, pero no menos importante, en la tercera fase se realizó la agrupación y comparación de los resultados para cada una de todas las configuraciones y opciones para llegar a concluir con una serie de conclusiones. Se creó un diagrama de Gantt donde se asignó con un plazo cada una de las tareas que se realizaron a lo largo del proyecto.

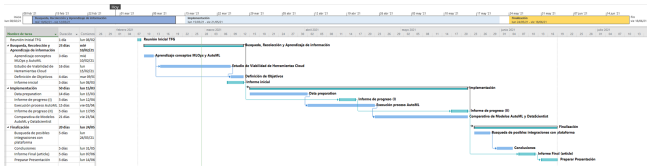


Fig. 8: Diagrama de Gantt: Planificación del TFG.

A.2. Consultar Código del Generador de Correos Electrónicos

Debido a la longitud del código y baja integración con el formato del informe, se ha creado un enlace para por consultarlo completamente. El enlace es el siguiente:

`https://pastebin.com/xx1aD6mJ`. [Creado: 25-may-2021]

A.3. Definición de Métricas

En esta sección del apéndice, se explicarán algunos conceptos clave para la comprensión de las métricas y los resultados obtenidos de un modelo de ML. Las definiciones a tener en cuenta son:

Precisión o Accuracy de clasificación, es la relación entre el número de predicciones correctas y el número total de muestras de entrada.

El **AUC** proporciona una refleja el rendimiento en todos los umbrales de clasificación posibles. Es el área bajo la curva ROC. Este puntaje nos da una buena idea de qué tan bien funciona un modelo. Una forma de interpretar el AUC, es como la probabilidad de que el modelo clasifique un ejemplo positivo aleatorio más alto que un ejemplo negativo aleatorio.

La pérdida de entropía cruzada, o pérdida logarítmica (**Log Loss**), mide el rendimiento de un modelo de clasificación cuya salida es un valor de probabilidad entre 0 y 1. La pérdida de entropía cruzada aumenta a medida que la probabilidad predicha diverge de la etiqueta real. Entonces, predecir una probabilidad de 0.012, cuando la etiqueta de observación real es 1 sería malo y daría como resultado un valor de pérdida alto. Un modelo perfecto tendría una pérdida logarítmica de 0.

A.4. Definición y Conceptos de MLOps

Para completar la **definición** de MLOps, las Operaciones de "Machine Learning" o también determinado en inglés como "Machine Learning Operations", abreviado en MLOps. Actualmente, nos encontramos en un mundo orientado a datos, que está vinculado una cantidad exponencial de los mismos recogidos digitalmente. Además, nos encontramos con la ascendente importancia de la Inteligencia Artificial y la Ciencia de Datos, abocando a una tremenda cantidad de información generada que necesita ser procesada y utilizada por las empresas en sus procesos de ML.

MLOps es una extensión de la metodología DevOps (Figura 9) que permite a los equipos de ciencia de datos y tecnología de la información, en colaborar y aumentar el ritmo del desarrollo y la implementación de modelos, aplicándose a lo largo de todo el ciclo de vida de desarrollo del software. Así, logrando la supervisión, validación y gobernanza de los modelos de "Machine Learning" usados en la empresa (se puede apreciar en la figura 10).

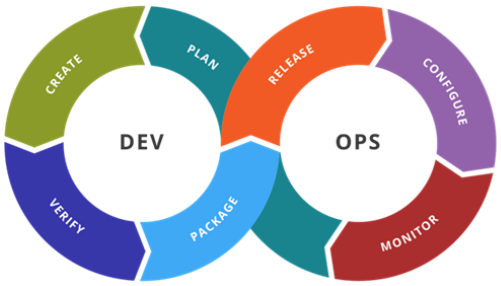


Fig. 9: Ciclo de Vida de DevOps.

Fig. 10: Ciclo de Vida MLOps.

El modelo de trabajo o ciclo de vida, que sigue MLOps de una manera más o menos generalizada, contiene las siguientes siete fases:

- Definición del problema y montaje de un conjunto de datos:

  Se ha de tener claro cuáles serán los datos de entrada para el algoritmo y cuáles los datos de salida. Al definir el problema o estudio, se debe considerar primero el "Dataset" que será utilizado o construido, sus componentes principales de análisis y los objetivos que serán alcanzados.

También se debe plantear cómo alcanzar los resultados y la forma en que serán evaluados.

- Análisis de las muestras:

  Al comenzar el análisis, se debe inspeccionar a modo de rutina, los conjuntos de datos que contienen las muestras de entrenamiento y demás. Se ha de buscar una descripción de los datos, inspeccionar la estructura y estadísticas descriptivas de los mismos, así logrando entender y conocer más el Dataset.

- Visualización de los datos:

  La forma en que se dividen y distribuyen los datos es importante para el entrenamiento del modelo. A fin de darse una idea del comportamiento de los datos, se utilizan gráficas que permitan visualizar los datos, buscando la correlación y distribución que poseen los datos entre ellos.

- Análisis estadístico:

  Antes de procesar los datos en cualquier algoritmo base seleccionado, es importante inspeccionar estadísticamente el conjunto de datos, así para poder enfocar el análisis con datos más precisos y segmentados. Se determinan dos actividades clave:

  1. La Curtosis y Asimetría Estadística, donde se determina un análisis sobre la distribución y varianza por cada eje, de tal modo que se pueda determinar el grado de Asimetría Estadística.
  2. Los Valores Atípicos, "Outliers" en inglés, que son los valores que no coinciden con el resto de los datos, por ejemplo, en una distribución normal, serían los datos que se encuentran en las colas. Estos pueden ser removidos del análisis para hacer más preciso el estudio.

- Algoritmos bases:

  En esta fase, es el momento de elegir el modelo que se ha de utilizar. Estos provienen de una serie de algoritmos base, que son las máquinas de aprendizaje que serán utilizadas en el análisis, y deberán ser ajustadas a los datos. Por ejemplo, las regresiones, máquinas de soporte vectorial o redes neuronales.

- Técnicas de Validación:

  Se realizan técnicas de validación, los cuales realizarán el entrenamiento y ajuste de los modelos de aprendizaje, tomando muestras de datos separadas, para analizar su comportamiento entre los diferentes bloques de muestras.

  En este proceso, se ha de tener en cuenta que los modelos de "Machine Learning" se mueven siempre entre el sobre-ajuste (se ha sobrepasado con el entrenamiento y el algoritmo predice muy bien sobre los datos de entrenamiento, pero no sobre otros diferentes) y el sub-ajuste en que al algoritmo no produce buenos resultados, ni siquiera con los datos de entrenamiento. El óptimo es situarse justo en la frontera entre el sub-ajuste ("under-fitting") y el sobre-ajuste ("over-fitting").

- Persistencia de los modelos:

  La fase más larga y donde se juega con el ensayo y error. Se busca optimizar y encontrar una solución más óptima al problema que se había planteado. Es importante como último punto hacer persistir los resultados obtenidos y modelos entrenados, para que estos puedan ser consumidos en el futuro por otras aplicaciones o ingenieros.

También se procede a explicar algunos **conceptos** nombrados en MLOps y que hay que tener en cuenta:

El modelo de red neuronal completamente convolucional (FCNN) es un modelo de aprendizaje profundo basado en el modelo de red neuronal convolucional (CNN) tradicional, con una primera capa completamente conectada y combina similitudes de expresión y similitudes de conocimiento previo como entrada. El CNN es un tipo de red neuronal artificial donde las neuronas corresponden a campos receptivos de una manera muy similar a las neuronas en la corteza visual primaria (V1) de un cerebro biológico.

Long short-term memory (LSTM) es una arquitectura de red neuronal recurrente artificial (RNN), utilizada en el campo del "Deep Learning". A diferencia de las redes neuronales de retroalimentación estándar, LSTM tiene conexiones de retroalimentación. No solo puede procesar puntos de datos individuales (como imágenes), sino también, secuencias completas de datos (como voz o video). Por ejemplo, LSTM es aplicable a tareas como el reconocimiento de escritura a mano conectada y no segmentada, el reconocimiento de voz y la detección de anomalías en el tráfico de red o IDS (sistemas de detección de intrusos).

TF-IDF, del inglés Term frequency – Inverse document frequency, es una medida numérica que expresa cuán relevante es una palabra para un documento en una colección. Esta medida se utiliza a menudo como un factor de ponderación en la recuperación de información y la minería de texto. El valor TF-IDF aumenta proporcionalmente al número de veces que una palabra aparece en el documento, pero es compensada por la frecuencia de la palabra en la colección de documentos, lo que permite manejar el hecho de que algunas palabras son generalmente más comunes que otras.

Stop-words o palabras vacías, es el nombre que reciben las palabras sin significado como artículos, pronombres, preposiciones, etc. que son filtradas antes o después del procesamiento de datos en un lenguaje natural (texto).

La codificación **One-Hot (One-Hot Encoding)** es un método para etiquetar a qué clase pertenecen los datos y la idea es asignar a 0, a toda la dimensión de categorías, excepto 1 para la clase a la que pertenecen los datos.