PARTE I

PROYECTO OPEN DATA II: IMPLEMENTACIÓN DEL ALGORITMO DE REGRESIÓN LINEAL

Objetivo general del proyecto:

El propósito general del proyecto es establecer un modelo para la relación entre características y una variable objetivo. Para el dataset utilizado en este caso, la primera consiste en una serie de calificaciones obtenidas por cierto estudiante y la segunda será la probabilidad de entrada de este estudiante en un máster en concreto. Se desea predecir la probabilidad de entrada a este máster en cuanto a las calificaciones obtenidas.

Algoritmos utilizados:

Hemos optado por la implementación de un algoritmo de regresión lineal, cuyo propósito es establecer un modelo para la relación entre características y una variable objetivo. En este caso, para realizar una predicción, utilizando unos datos de entrenamiento que ya tenemos es el algoritmo adecuado para ello (es un ejemplo de aprendizaje supervisado).

Herramientas, software, hardware y lenguajes utilizados:

El marco general del desarrollo de nuestro proyecto (implementado en la asignatura de Open Data) es Jupyter, donde programamos utilizando el lenguaje Python. Nos sumergimos en el mundo de programación con pyspark, utilizado en la dinámica de programación en distribuido. A través de pyspark hemos trabajado con un algoritmo de machine learning, así pues pudiendo trabajar con las bases de uno de los fenómenos más populares en el mercado ahora mismo.

El hardware consiste en un ordenador portátil personal de gama media.

Selección de atributos y etiquetas:

Las columnas con las calificaciones de las distintas pruebas las juntamos bajo una única columna ‘features’ y la probabilidad de entrada en otra llamada ‘label’. Utilizamos estas dos etiquetas porque son con las que suelen trabajar los algoritmos machine learning.

Todo esto lo hicimos para poder seleccionar nuestro training y testing sets. En este conjunto de datos, separamos los datos en datos de entrenamiento y de testeo de tal manera: (85% para training y 15% para testing), ya que contamos con pocos datos, y debemos utilizar una mayor proporción en el entrenamiento.

Tiempo total en obtener las respuestas:

La implementación de Pyspark tardaba varios minutos en ejecutarse. Es un tiempo adecuado ya que no requerimos datos en tiempo real.

Ficha descriptiva del dataset usado:

Nuestros parámetros son los siguientes:

1. GRE Scores (de 0 a 340): una prueba que consta de uno de los requisitos de admisión en las escuelas de postgrado en los Estados Unidos y en otros países anglosajones.

2. TOEFL Scores (de 0 a 120): prueba de inglés.

3. Valoración de la universidad (de 0 a 5)

4. Declaración de propósito y carta de recomendación (de 0 a 5)

5. GPA Scores (de 0 a 10): valoración general de tus notas finales (según el sistema estadounidense de calificaciones)

6. Experiencia en investigación (0 o 1)

7. Probabilidad de ser admitido (entre 0 y 1)

El dataset son nueve columnas y 400 filas (12.6 KB de datos). Los datos se presentan en formato CSV.

Aplicación práctica del proyecto en vida real

Las predicciones realizadas son de especial utilidad para aquellos posibles interesados en acceder a este máster en esta universidad en particular. A través de los resultados se puede valorar cuánto debes alcanzar en cada calificación, sobre todo si quieres asegurarte una plaza en el máster, o quieres optar a una muy alta probabilidad de entrada. Filtrar las pruebas que tienen más peso es una buena acción a tomar con los resultados, de tal forma que puedes organizar cuánto esfuerzo le debes dar a cada prueba en proporción con el resto. Puedes informarte de los pasos a seguir para alcanzar los resultados requeridos.

A su vez, si es el caso en el que un estudiante ya tiene sus calificaciones, con este estudio puede tener una idea de cuáles son sus probabilidades de entrada en esta universidad en particular.

Se podría realizar un estudio parecido con otras universidades de interés, y hacer una comparación de resultados, comparando las notas necesarias de entrada. Con todo esto se puede hacer un estudio global y detallado que fortalezca el poder de decisión a la hora de elegir universidad.

Consideraciones legales y éticas en la obtención/tratamiento del dataset y resultado

La licencia de los datos es de carácter CC0: Public Domain, es decir que al ser los datos de dominio público, se pueden copiar, modificar y distribuir, incluso con fines comerciales. Todo esto no requiere de ningún permiso. Por lo tanto no habrá ninguna consecuencia legal tanto en el proceso de obtención como se procesado y análisis de los datos.

No se desvelan datos personales sobre la fuente de datos del dataset, por lo tanto se reducen los factores éticos, ya que el trabajo realizado con estos datos no afecta a particulares que proporcionen los datos. El resultado, de la misma manera se puede difundir.

Conclusiones

Con este proyecto nos adentramos en el mundo del machine learning, pudiendo apreciar en directo la potencia de los algoritmos más utilizados. En este caso, manejar predicciones es un área que está en auge en muchas empresas. Incluir mejoras en el modelo, nos permite ver que los modelos pueden ser mejorados y que van a depender de varios factores. Es importante encontrar los datos correctos para hacer el entrenamiento, y obtener los mejores parámetros para nuestro modelo. De tal forma se obtiene la mejor predicción en base a nuestros datos de entrada.

PARTE II

Arquitectura utilizada para la migración (Pipeline)

Ya que nuestro dataset tiene un tamaño muy pequeño, no lo consideramos un problema de big data, por lo tanto no vamos a requerir un cluster como parte de nuestra arquitectura. Esto va a condicionar el setup general.

Al no necesitar un cluster, podemos trabajar directamente en BigQuery (que ya ofrece un cluster por detrás administrado y escalado automáticamente por la plataforma). Emplearemos BigQuery para que, en base a las consultas adecuadas, podamos obtener una predicción de la probabilidad de entrada al máster (se podría mostrar agregadas por combinación de calificaciones académicas). Cabe destacar que dado el tamaño pequeño de nuestro dataset, será conveniente usar una partición 85% - 15% para el entrenamiento y evaluación, a diferencia del típico 80% - 20%.

A su vez, debemos acudir a DataFlow para poder administrar las tareas y desarrollar eficazmente el pipeline (ingesta de datos, entrenamiento, evaluación del modelo, generación de predicciones). Con esta herramienta podremos programar la obtención, preparación y procesamiento de los datos a intervalos regulares, posiblemente de forma semanal.

Es importante mencionar que trabajaremos con datos en batch, y esto también es importante ponerle énfasis ya que también va a afectar la manera en la que desarrollemos sobre todo, la presentación de nuestros outputs. Hemos decidido que una página web sería un modelo interesante para presentar los resultados, ya que es fácilmente accesible para el público que pueda estar interesado en esa universidad, o que simplemente quiera barajar opciones. Esta última parte, posiblemente la gestionaremos con Firebase.

Productos a utilizar en GCP

Nuestro producto base será BigQuery, que es donde se desarrollara tanto la ETL de los datos, como el algoritmo machine learning con el que trabajamos (en este caso, regresión lineal). Usaremos Data Studio para visualizar los resultados del modelo. Almacenaremos los datos en un Storage estándar, y las predicciones resultantes en una coldline. Mostraremos los datos a los usuarios mediante Firebase ya que lo ofrece google, y por lo tanto se pueden conectar sus servicios fácilmente con los productos que utilizaremos de GCP. Y estas herramientas serán gestionadas mediante Dataflow.

Beneficios obtenidos en términos de tiempo y dinero

En este caso, no obtenemos muchos beneficios en cuanto a tiempo ni dinero, ya que es un análisis que podemos realizar en una plataforma gratuita (como jupyter) ya que no necesitamos especial poder de procesamiento. Sin embargo, GCP nos sigue ofreciendo un beneficio en cuanto a las opciones para representar nuestros datos, ya que podemos conectar nuestra arquitectura GCP directamente con un producto como Fire Storage en la que se pueden gestionar los resultados al gusto del cliente. En cuanto a dinero, si por alguna razón se incrementa el dataset, o se plantean nuevos proyecto en torno a este, el ahorro en coste sería grande, ya que podríamos procesar toda esta información, de forma rápida, administrada, sin costes de hardware.

Costes previstos en la migración y posterior explotación

Tras haber escogido nuestras herramientas, el coste de emplear GCP según el Price estimator de GCP seria:

BigQuery on demand, tabla muy pequeña y pocas instrucciones.

DataFlow 1 x n1-standard-1 workers in Batch Mode.

Google Cloud Storage, con 1GB nos sobra, y menos de 10000 operaciones.

Cloud Storage Coldline, con 1GB nos sobra, y menos de 10000 operaciones.

Firebase no aparece en la herramienta de estimación de precios. Tiene tres planes de pago, Spark (gratuito), Flame (25 $ por mes) y Blaze (es personalizado según uso). Asumiendo que usamos el plan gratuito, o que el personalizado no resulta demasiado caro, nuestro coste queda estimado en:

Total Estimated Cost: USD 0.25 per 1 month