Negocio

Ofrecemos un modelo predictivo de scoring crediticio cuyo principal objetivo es averiguar si un cliente va a poder afrontar el pago de un préstamo concedido por una entidad bancaria, mediante la creación de una nueva variable de clasificación que resume el modelo entero, donde (1, el cliente podrá afrontar el préstamo), (0, el cliente no podrá afrontar el pago).Hemos incluido una variable nueva que hace referencia a la comisión que paga el cliente en forma de intereses por la devolución del préstamo, ya que un cliente con un scoring bajo que tenga que pagar elevadas comisiones lo mas probable es que no acabe haciendo frente al préstamo…

También estamos trabajando en un nuevo modelo similar pero que vuelca todo el estudio en una nueva variable que explica la probabilidad de que el cliente pueda hacer o no frente al préstamo. Donde si la probabilidad es > 85% se le concede el préstamo, mientras que toda probabilidad inferior nos hará un cálculo automático de cuanta cantidad se le podrá conceder como máximo al cliente para que la probabilidad de devolución del préstamo sea igual o superior al 85%, si en este segundo punto seguimos viendo que el cliente tiene muchas papeletas para no pagar ninguna cantidad, no se le concedería ninguna clase de préstamo.

GCP

Como poder desplegar un modelo en GCP para hacer llamadas a la API y poder realizar las predicciones.

* Usa una canalización de scikit-learn para entrenar un modelo en el [conjunto de datos Iris](https://scikit-learn.org/stable/auto_examples/datasets/plot_iris_dataset.html).
* Guarda el modelo en forma local.
* Sube el modelo guardado a Cloud Storage.
* Crea un recurso y una versión de modelo de AI Platform Prediction.
* Obtén predicciones en línea para dos instancias de datos.

CLUSTERING

El **Clustering** es una tarea que consiste en agrupar un conjunto de objetos (no etiquetados) en subconjuntos de objetos llamados **Clusters**. Cada **Cluster** está formado por una colección de objetos que son similares (o se consideran similares) entre sí, pero que son distintos respecto a los objetos de otros Clusters. En nuestro modelo hemos utilizado el ‘Elbow method’ para averiguar el número de clusters que maximiza el rendimiento del modelo.

Gráfico, Gráfico de líneas

Descripción generada automáticamente

En la siguiente gráfica se puede observar como en k=2 la pendiente del modelo cambia radicalmente, mostrándonos el número de clusters necesarios. Para poder implantarlo en el modelo, en primer lugar tuvimos que realizar el train, y dado que los resultados eran correctos, pudimos implantarlo en el test.

Gráfico, Gráfico de dispersión

Descripción generada automáticamente

Si observamos el siguiente gráfico de dispersión de los clusters, se observan los dos subgrupos en los que se dividen el modelo, el único problema que hemos observado han sido que en el segundo cluster hay extremos que se dispersan de la tendencia del resto, de todos modos hemos decidió no eliminarlos y seguir manteniéndolos en el modelo para evitar underfitting.

PCA

Principal Component Analysis (PCA) es un método estadístico que permite simplificar la complejidad de espacios muestrales con muchas dimensiones a la vez que conserva su información. Es decir, su objetivo es obtener un conjunto de datos nuevo que mantiene la esencia del conjunto original, pero a que diferencia maximiza el accuracy del modelo. En nuestro caso, en el primer train con el pca obtuvimos un 0.79 de accuracy del modelo, pudiéndose observar todo el código utilizado en nuestro git.

Feature importance

Es una análisis que se realiza para obtener la eficiencia e importancia de las variables existentes del modelo, con el objetivo final de quedarnos con aquellas variables que maximizan nuestro nivel de accuracy. Hemos utilizado el método de feature engineering para la extracción del conjunto de datos que maximiza la utilidad de las variables.

Imagen que contiene Tabla

Descripción generada automáticamente