Informe predicción de transacciones de clientes

Medellín, 28/05/2023

RAÚL RAMOS POLLÁN
Introducción a la inteligencia de artificial
Facultad de ingeniería

Introducción

Este informe ejecutivo presenta una visión general de la competencia "Santander Customer Transaction Prediction" organizada por Santander, una destacada institución financiera. El objetivo de esta competencia es desarrollar modelos de aprendizaje automático que puedan predecir si un cliente realizará una transacción específica en el futuro. Este informe proporcionará una descripción general de la competencia, los desafíos que aborda y las posibles soluciones que se pueden aplicar.

Descripción de la competencia:

La competencia "Santander Customer Transaction Prediction" se centra en utilizar un conjunto de datos anónimos proporcionados por Santander para predecir si un cliente realizará o no una transacción futura. El conjunto de datos contiene una amplia gama de características numéricas relacionadas con cada cliente. El objetivo principal es construir modelos predictivos precisos utilizando técnicas de aprendizaje automático para identificar aquellos clientes que son más propensos a realizar una transacción en particular.

Posibles soluciones:

Para resolver los desafíos mencionados, se pueden aplicar varias técnicas y enfoques. Algunas de las posibles soluciones incluyen:

- Análisis exhaustivo de datos: Es fundamental realizar un análisis exploratorio de datos detallado para comprender las características clave y su influencia en la variable objetivo.
- Ingeniería de características: La creación de características nuevas y relevantes, así como la selección adecuada de características existentes, pueden mejorar significativamente la precisión del modelo.
- Modelos predictivos avanzados: Se pueden utilizar modelos de aprendizaje automático avanzados, como árboles de decisión, bosques aleatorios, gradient boosting o redes neuronales, para capturar patrones complejos y relaciones en los datos.
- Técnicas de muestreo y ponderación: Para abordar el desequilibrio de clases, se pueden aplicar técnicas como submuestreo, sobre-muestreo o ponderación de clases durante el entrenamiento del modelo.

Exploración descriptiva del dataset

El dataset a utilizar es el conjunto de datos de la competición "Santander Customer Transaction Prediction" alojado en Kaggle (Santander Customer Transaction Prediction | Kaggle). Este conjunto de datos contiene 200,000 instancias de entrenamiento y 200,000 instancias de prueba, y consta de 200 características numéricas (columnas) anónimas que representan los atributos de los clientes y su historial

de transacciones. En estos datos se tiene la variable target la cual es el objetivo a predecir y está compuesta por 1 o 0, lo cual. Teniendo esto en cuenta, se hizo una revisión de los datos:

```
## KEEPOUTPUT
import pandas as pd
import seaborn as sns
import numpy as np
import matplotlib.pyplot as plt
%matplotlib inline
d = pd.read_csv("train.csv")
d.head()
```

	ID_code	target	var_0	var_1	var_2	var_3	var_4	var_5	var_6	var_7	...	var_190	var_191	var_192	var_193	var_194	var_195	var_196	var_197	var_198	var_199
0	train_0	0	8.9255	-6.7863	11.9081	5.0930	11.4607	-9.2834	5.1187	18.6266	...	4.4354	3.9642	3.1364	1.6910	18.5227	-2.3978	7.8784	8.5635	12.7803	-1.0914
1	train_1	0	11.5006	-4.1473	13.8588	5.3890	12.3622	7.0433	5.6208	16.5338	...	7.6421	7.7214	2.5837	10.9516	15.4305	2.0339	8.1267	8.7889	18.3560	1.9518
2	train_2	0	8.6093	-2.7457	12.0805	7.8928	10.5825	-9.0837	6.9427	14.6155	...	2.9057	9.7905	1.6704	1.6858	21.6042	3.1417	-6.5213	8.2675	14.7222	0.3965
3	train_3	0	11.0604	-2.1518	8.9522	7.1957	12.5846	-1.8361	5.8428	14.9250	...	4.4666	4.7433	0.7178	1.4214	23.0347	-1.2706	-2.9275	10.2922	17.9697	-8.9996
4	train_4	0	9.8369	-1.4834	12.8746	6.6375	12.2772	2.4486	5.9405	19.2514	...	-1.4905	9.5214	-0.1508	9.1942	13.2876	-1.5121	3.9267	9.5031	17.9974	-8.8104

5 rows × 202 columns

Figura 1. Tabla con las primeras 5 filas y algunas variables que contiene el data set.

Se observaron el nombre de las columnas, el tipo de datos en ellas y la información faltante. En este paso se observaron diferentes problemas:

1. La cantidad de datos era exagerada teniendo en cuenta los requerimientos mínimos del proyecto y el gasto computacional que estos demandarían.
2. Las variables no tenían nombres que dieran información sobre qué se estaba observando
3. No se contaba con la cantidad mínima de variables categóricas que exigía la entrega.
4. El dataset estaba completo, no contenía variables faltantes. Mientras que los requisitos exigen al menos un 5% de datos faltantes.

Para solucionar estos inconvenientes se llevaron a cabo los siguientes pasos:

- Se recortaron los datos iniciales a un total de 50 columnas y 10000 filas. Este recorte se hizo teniendo en cuenta que los requisitos mínimos eran 30 columnas y 2000 filas, sin embargo, no se quiso limitar estrictamente a esos valores ya que sería una gran pérdida de información.
- Se seleccionaron de manera aleatoria el 10% de las columnas para convertirlas en variables categóricas por medio de la función Kmeans. Se seleccionó como parámetro 5 categorías.
- Se eliminaron los datos de algunas columnas para simular datos faltantes. Esto se hizo por medio de una lista de columnas aleatorias. En este código, ‘numpy.random.choice’ se usa para seleccionar aleatoriamente tres columnas en el conjunto de datos en la lista ‘columnas_faltantes’. Luego, se calcula el número de celdas que deben estar ausentes para cumplir con el requisito del 5% en ‘num_celdas_faltantes’. Se seleccionan filas aleatorias en cada columna elegida y se reemplazan los valores correspondientes con NaN. Finalmente, el conjunto de datos modificado se guarda en un nuevo archivo CSV usando la función ‘to_csv’.
- Se volvió a analizar el dataset resultante para saber si ahora cumple con las características necesarias.

Al verificar que se cumplían los requisitos básicos, se siguió explorando el data set. Al visualizar la variable objetivo, que en este caso es target, se observó que hay una gran cantidad de 0 (indica que el cliente no ha hecho una transacción) a diferencia de 1 (indica que el cliente ha hecho una transacción)

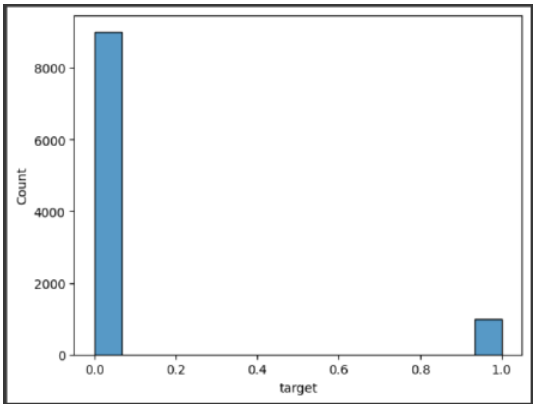


Figura 3. Inspección de la variable objetivo

También se hizo una inspección de algunos como la media, desviación estándar, el máximo, el mínimo, entre otros, de cada una de las variables. Se observan diferentes desviaciones estándar entre las variables, algunas altas y otras no tanto. Además, una alta diferencia entre las medias. Esto indica una alta variabilidad entre los datos. Una alta desviación estándar en los datos indica que los valores de la muestra están muy dispersos alrededor de la media. En otras palabras, los datos están más alejados de la media y son más heterogéneos. Una baja desviación estándar, por otro lado, indica que los valores están más cercanos a la media y son más homogéneos. Por otro lado, se observa una variabilidad entre las medias de cada variable. La presencia de diferentes medias en un conjunto de datos (data set) indica que hay diferencias o variaciones en los valores que se están midiendo. En otras palabras, los datos no son homogéneos y pueden haber subgrupos o categorías en los datos que están influyendo en las diferentes medias. Se puede decir entonces que la media no es necesariamente representativa de los datos en general.

Iteraciones de desarrollo

Modelos supervisados

Regresión Logística: La regresión logística es un modelo de aprendizaje supervisado utilizado para problemas de clasificación binaria. A diferencia de la regresión lineal, que se utiliza para predecir valores continuos, la regresión logística se utiliza para predecir la probabilidad de que una muestra pertenezca a una clase específica. La regresión logística utiliza una función logística o sigmoide para transformar una combinación lineal de características en una probabilidad en el rango de 0 a 1. Si la probabilidad estimada es superior a un umbral predefinido (por lo general, 0.5), la muestra se clasifica en la clase positiva; de lo contrario, se clasifica en la clase negativa.

Árboles de Decisión: Los árboles de decisión son modelos de aprendizaje supervisado que utilizan una estructura de árbol para tomar decisiones basadas en reglas condicionales. Cada nodo interno del árbol representa una característica o atributo, y las ramas salientes representan los posibles valores que puede tomar esa característica. Las hojas del árbol representan las etiquetas de clasificación. El árbol de decisión divide recursivamente el conjunto de datos en subconjuntos más pequeños basándose en las características más informativas y utiliza criterios de impureza (como la ganancia de información o la ganancia de Gini) para determinar la mejor manera de dividir los datos en cada nodo. El parámetro de profundidad representa la cantidad de niveles o divisiones que se realizan en el árbol. Cuanto mayor sea la profundidad, más complejo será el modelo y más características y patrones podrá capturar.

Modelos no supervisados

Clustering: K-means clustering es un algoritmo de aprendizaje no supervisado utilizado para agrupar un conjunto de datos en subconjuntos más pequeños y coherentes. El objetivo principal del algoritmo es dividir los datos en grupos, llamados "clusters", donde los elementos dentro de cada cluster son más similares entre sí que con aquellos en otros clusters. Con el dataset preprocesado se procedió a realizar la agrupación de los datos como se observa a continuación:

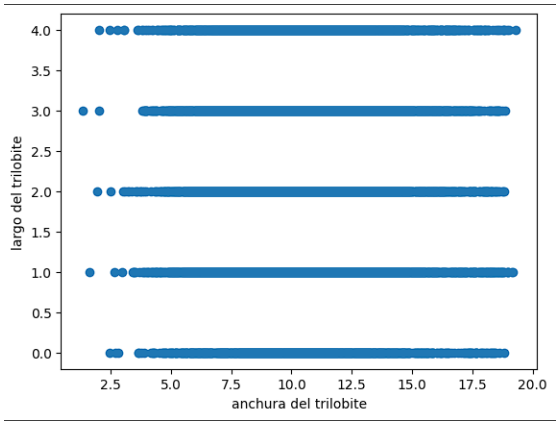


Figura 4. Datos agrupados para el método cluster.

El algoritmo K-means comienza definiendo el número de clusters (k) que se desean crear. Luego, asigna aleatoriamente los puntos de datos a k centroides iniciales. Los centroides representan puntos centrales dentro de cada cluster. A continuación, el algoritmo itera en dos pasos hasta que se alcance la convergencia:

- Asignación de puntos a clusters: Cada punto de datos se asigna al cluster cuyo centroide está más cerca de él en términos de distancia euclidiana u otra métrica de distancia.
- Actualización de los centroides: Los centroides de los clusters se recalculan tomando como referencia la media de todos los puntos asignados a ese cluster.

Estos dos pasos se repiten hasta que los centroides ya no cambien significativamente o se alcance un número máximo de iteraciones. En cada iteración, los puntos se reasignan a los clusters y los centroides se actualizan en base a la nueva asignación. El algoritmo converge cuando los puntos se estabilizan en clusters y los centroides no cambian significativamente. El resultado final del algoritmo K-means es una partición de los datos en k clusters, donde los puntos dentro de cada cluster son similares entre sí en comparación con los puntos de otros clusters. Con lo anterior, los resultados obtenidos para el dataset asignado fueron:

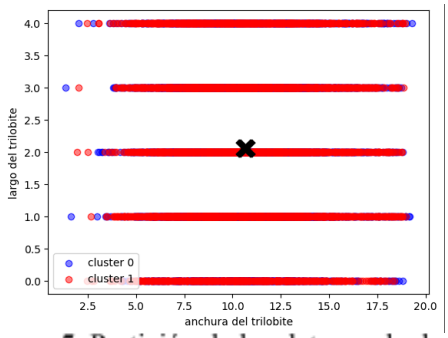
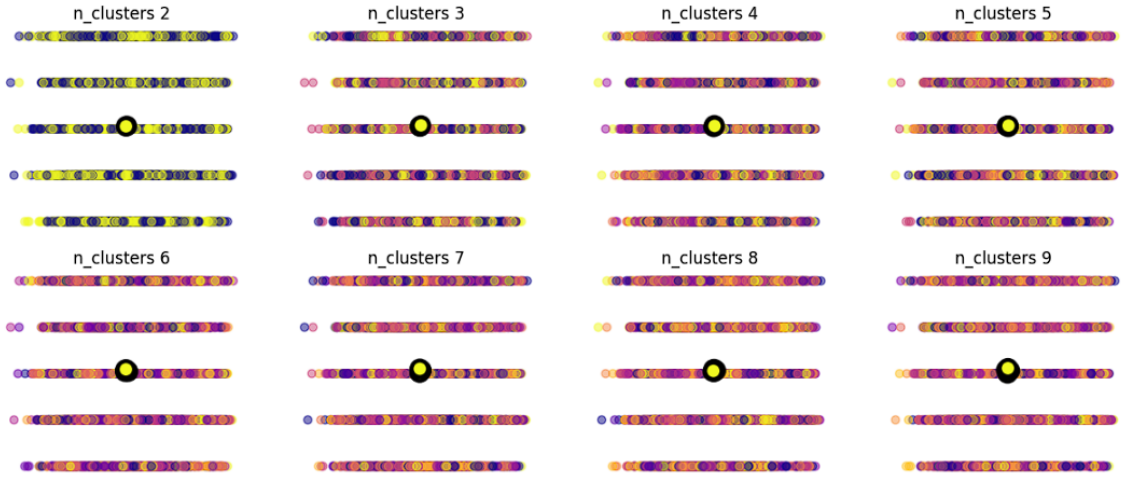


Figura 5. Partición de los datos en k clusters

Y evaluando para diferentes número de clusters se obtienen resultados similares:



|Figura **6**. Partición de los datos para diferentes números de clusters

PCA(Análisis de Componentes Principales): es una técnica de reducción de dimensionalidad utilizada en el análisis de datos y la minería de datos. Su objetivo principal es transformar un conjunto de variables originales en un conjunto de variables no correlacionadas llamadas "componentes principales", que son combinaciones lineales de las variables originales.

El PCA busca capturar la mayor cantidad de variabilidad en los datos utilizando un número menor de componentes principales. Al hacerlo, permite simplificar la estructura de los datos y reducir la dimensionalidad, lo que facilita la **visualización**, el análisis y la interpretación de los datos.

El PCA ofrece varias ventajas, como la capacidad de reducir la dimensionalidad de los datos, identificar patrones y tendencias, eliminar la multicolinealidad entre variables y simplificar el análisis. Se utiliza en una amplia gama de aplicaciones, incluyendo análisis de datos, reconocimiento de patrones, compresión de imágenes, clasificación y clustering

Al implementar este método en el data set, se obtuvieron las siguientes varianzas:

```
Varianza explicada por cada componente principal: [3.40965745e-01 8.27161499e-02 5.08843521e-02 4.85150147e-02
 4.65267526e-02 4.57749383e-02 4.50911602e-02 3.35131791e-02
 2.67094527e-02 2.64590539e-02 2.59886172e-02 2.55267362e-02
 2.51188035e-02 2.24902296e-02 1.57292444e-02 1.38345708e-02
 1.34861248e-02 1.22213608e-02 1.05895499e-02 8.76719548e-03
 8.15649451e-03 7.16243264e-03 6.84769644e-03 6.06611061e-03
 5.98956423e-03 5.19020878e-03 5.13168178e-03 4.92782839e-03
 4.81301797e-03 3.73209099e-03 3.67964080e-03 3.50556294e-03
 3.08037882e-03 2.00251239e-03 1.60676906e-03 1.43648172e-03
 1.39632232e-03 1.17279191e-03 1.12263075e-03 5.60941567e-04
 4.44292030e-04 3.54334309e-04 2.19280600e-04 2.09824978e-04
 1.23690153e-04 7.21893761e-05 6.05389308e-05 2.64599225e-05]
Varianza total explicada: 1.0
Varianza explicada acumulada: [0.34096574 0.42368189 0.47456625 0.52308126 0.56960801 0.61538295
 0.66047411 0.69398729 0.72069674 0.7471558  0.77314442 0.79867115
 0.82378996 0.84628018 0.86200943 0.875844   0.88933012 0.90155149
 0.91214104 0.92090823 0.92906473 0.93622716 0.94307485 0.94914097
 0.95513053 0.96032074 0.96545242 0.97038025 0.97519327 0.97892536
 0.982605   0.98611056 0.98919094 0.99119345 0.99280022 0.9942367
 0.99563303 0.99680582 0.99792845 0.99848939 0.99893368 0.99928802
 0.9995073  0.99971712 0.99984081 0.999913   0.99997354 1.        ]
```

Varianza explicada por cada componente principal: Estos valores indican la cantidad de varianza en los

datos originales que es explicada por cada componente principal. Cuanto mayor sea el valor, más información captura ese componente principal. En este caso, los primeros componentes principales explican la mayor parte de la varianza, mientras que los últimos componentes explican una cantidad cada vez menor de varianza.

Varianza total explicada: Este valor indica la fracción total de varianza en los datos originales que es explicada por todos los componentes principales combinados. En este caso, la varianza total explicada es igual a 1.0, lo que significa que todos los componentes principales juntos capturan la totalidad de la varianza en los datos originales.

Varianza explicada acumulada: Estos valores muestran la acumulación de la varianza explicada a medida que se agregan más componentes principales. La varianza explicada acumulada puede ayudar a determinar cuántos componentes principales se deben conservar para capturar una cantidad significativa de varianza en los datos. En este caso, se puede observar que los primeros componentes principales contribuyen con una gran cantidad de varianza, y a medida que se agregan más componentes, la varianza explicada acumulada aumenta gradualmente.

Resultados, métricas y curvas de aprendizaje

Método supervisado

A la hora de evaluar el desempeño de un modelo de clasificación se pueden emplear métricas como el accuracy, la cual mide la proporción de predicciones correctas en comparación con el total de predicciones realizadas. Al llevar a cabo un primer modelo de **regresión logística** calibrando el modelo con los datos de prueba y luego evaluando el modelo con datos nuevos de validación se obtuvo como resultado un accuracy de 0.9011 con los datos de evaluación y de 0.9006 con los de evaluación, pero estos datos no son confiables ya que el algoritmo de optimización utilizado en la regresión logística no ha convergido por completo, lo que podría afectar el rendimiento del modelo y su capacidad para realizar predicciones precisas.

Para tratar de mejorar los resultados del modelo de regresión logística se aumentó el número máximo de iteraciones, sin embargo, al modificar este parámetro el algoritmo seguía sin converger por lo que puede tener dificultades para ajustarse adecuadamente a los datos y se optó por probar más modelos supervisados. Posteriormente, se probó un modelo de **árboles de decisión con profundidad 2** y los resultados se consignan en la tabla I:

Tabla I. Métricas de desempeño árbol de decisión 1

Métrica	Entrenamiento	Validación
Accuracy	0.9006	0.90110
Precision	0.0000	1.00000
Recall	0.0000	0.00503
F1 Score	0.0000	0.01001

Con base a la tabla I se aprecia que el accuracy de entrenamiento es de 0.9006, lo que significa que el 90.06% de las predicciones en los datos de entrenamiento son correctas. El accuracy de validación es de 0.9011, lo que indica que el 90.11% de las predicciones en los datos de validación son correctas.
Sin embargo, la métrica de precisión mide la proporción de instancias positivas correctamente

identificadas en relación con el total de instancias etiquetadas como positivas. Para el entrenamiento, se obtuvo una precisión de 0.0, lo que sugiere que no se han identificado correctamente instancias positivas. Sin embargo, en la validación, la precisión es de 1.0, lo que significa que todas las instancias etiquetadas como positivas se identificaron correctamente. Además, el Recall mide la proporción de instancias positivas correctamente identificadas en relación con el total de instancias que deberían haber sido etiquetadas como positivas. En este caso, el recall de entrenamiento es de 0.0, lo que indica que no se han identificado correctamente las instancias positivas. En la validación, el recall es de 0.00503, lo que significa que solo se ha identificado correctamente un pequeño porcentaje de las instancias positivas.

Con respecto al F1 Score, corresponde a una medida que combina la precisión y la exhaustividad, proporcionando un equilibrio entre ambas métricas. Para el entrenamiento se obtuvo que el F1 score es de 0.0, lo que sugiere que hay un desequilibrio entre la precisión y el recall. En la validación, el F1 score es de 0.01001, mostrando un ligero incremento pero aún señalando una falta de equilibrio entre la precisión y el recall.

En general, los resultados muestran que el modelo tiene un alto accuracy en ambos conjuntos de datos, lo cual es positivo. Sin embargo, es importante tener en cuenta que el modelo muestra una baja capacidad para identificar correctamente las instancias positivas, ya que tanto la precisión como el recall son bajos. Esto podría indicar un desequilibrio en los datos o la necesidad de ajustar los parámetros del modelo.

Con base a lo anterior, se ajustó la profundidad del árbol de decisión a 10 para evaluar si existe mejoría en los resultados clasificatorios del modelo, los cuales se registraron en la tabla II y la figura 8.

Tabla II. Métricas de desempeño árbol de decisión 2

Métrica	Entrenamiento	Validación
Accuracy	0.932300	0.909100
Precision	0.949008	0.988506
Recall	0.337022	0.086519
F1 Score	0.497402	0.159112

En el contexto de los datos del dataset, en la clasificación de clientes potenciales (1 si lo es y 0 no lo es), una de las métricas más importantes a considerar es la precisión. Esta métrica indica cuántas de las muestras clasificadas como positivas son realmente positivas.

En este caso, el modelo de árbol de decisión con profundidad 2 tiene una precisión perfecta de 1.0000 en el conjunto de validación, lo que significa que todas las muestras clasificadas como clientes potenciales son realmente positivas. En contraste, el modelo de árbol de decisión con profundidad 10 tiene una precisión ligeramente menor de 0.9885 en el conjunto de validación.
Sin embargo, es importante tener en cuenta que el modelo de árbol de decisión con profundidad 2 tiene un recall muy bajo de 0.0050 en el conjunto de validación, lo que indica que clasifica muy pocos clientes potenciales correctamente. Esto sugiere que el modelo de árbol de decisión con profundidad 2 podría estar perdiendo muchos casos positivos.

Por lo tanto, teniendo en cuenta la precisión y el recall, el modelo de árbol de decisión con profundidad 10 parece ser mejor en términos de clasificación de clientes potenciales. Aunque tiene una precisión ligeramente inferior, su capacidad para identificar clientes potenciales verdaderos (recall) es significativamente mejor que la del modelo de árbol de decisión con profundidad 2.

Método no supervisado

Clusters: Al evaluar diferentes números de clusters con el algoritmo K-means todos convergen hacia el mismo punto, esto puede indicar que los datos no presentan una estructura clara y no son adecuados para

ser agrupados utilizando el algoritmo K-means. Hay algunas interpretaciones para esta situación:

1. Datos no agrupables: Es posible que los datos no tengan una estructura natural de grupos o que la variabilidad entre los puntos sea muy baja. En este caso, es difícil para el algoritmo K-means encontrar diferencias significativas entre los puntos y, por lo tanto, todos los clusters terminan colapsando en un solo punto.
2. Desequilibrio en los datos: Si hay una gran desproporción entre la cantidad de puntos en diferentes clusters, el algoritmo K-means puede ser dominado por el cluster más grande. Esto puede hacer que los otros clusters se colapsen hacia el centroide del cluster más numeroso.
3. Inicialización inadecuada: La inicialización aleatoria de los centroides puede influir en el resultado del clustering. Si los centroides iniciales se seleccionan de manera desafortunada y están muy cerca unos de otros, es posible que todos los clusters se fusionen en un solo punto.
4. Número incorrecto de clusters: Es importante seleccionar el número adecuado de clusters (k) para tus datos. Si eliges un valor incorrecto de k, puede afectar negativamente la capacidad del algoritmo para identificar y separar los clusters correctamente

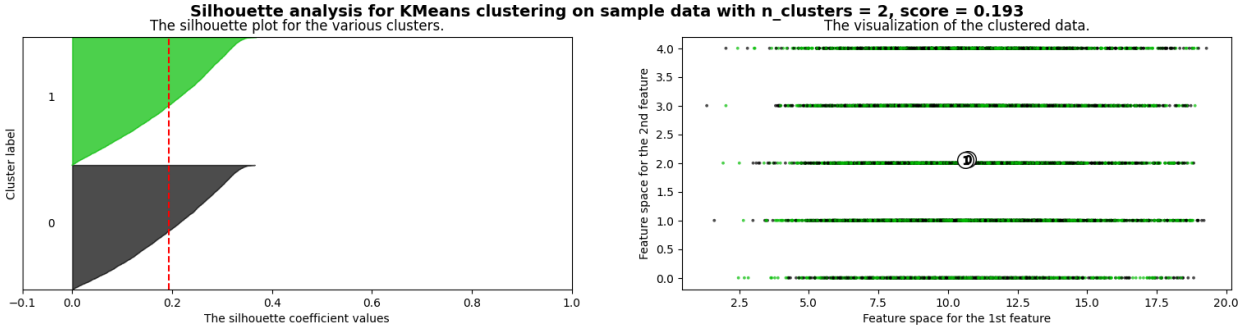


Figura 7. Análisis de silueta para K Means clustering

En estos casos, es recomendable realizar un análisis más profundo de los datos y considerar otros métodos no supervisados para obtener una mejor comprensión de la estructura.

Al implementar el método PCA en un modelo supervisado de árbol de decisión con una determinada profundidad para el dataset, se obtuvieron las siguientes metricas:

```
Metrics:
Accuracy: 0.871
Precision: 0.13953488372093023
Recall: 0.061224489795918366
```

Se observa que al reducir la dimensionalidad, no se obtuvieron mejoría en los resultados, ya que en las métricas se observa un menor precisión y un menor grado de acierto. Esto puede ser debido a diferentes situaciones, una de ellas es la pérdida de información relevante, ya que al reducir la dimensionalidad de los datos mediante PCA, es posible que se haya perdido información relevante para la tarea de clasificación. PCA busca maximizar la varianza en los datos, pero esto no garantiza que la información más discriminativa para la clasificación se conserve. En algunos casos, puede haber características específicas que sean más importantes para el modelo y que se hayan perdido durante el proceso de reducción dimensional. Otros factores importantes a tener en cuenta que el rendimiento del modelo no solo depende del método PCA, sino también de otros factores como la elección de hiperparámetros del

árbol de decisión, el preprocesamiento de los datos, la calidad y cantidad de los datos de entrenamiento, entre otros. Es posible que otros aspectos del modelo estén limitando su rendimiento y no estén relacionados directamente con el uso de PCA.

En general, cuando no se observan mejoras en el rendimiento del modelo al aplicar PCA, es importante evaluar cuidadosamente diferentes factores y considerar si PCA es la mejor opción para el conjunto de datos y la tarea de clasificación específica que se está abordando.PCA

Conclusiones

- Se probaron varias soluciones para abordar los desafíos de la competencia. Estas soluciones incluyen un análisis exhaustivo de datos, ingeniería de características, el uso de modelos predictivos avanzados y técnicas de muestreo y ponderación.
- Se realiza una exploración descriptiva del conjunto de datos utilizado en la competencia. Se mencionan la cantidad de instancias de entrenamiento y prueba, así como las características numéricas presentes en los datos. También se destaca la presencia de la variable objetivo y se observa un desequilibrio entre las clases 0 y 1.
- Se mencionan diferentes modelos de aprendizaje automático utilizados en el proyecto, tanto supervisados como no supervisados. Se mencionan la regresión logística, los árboles de decisión, el clustering con el algoritmo K-means y el análisis de componentes principales (PCA).
- Se presentan los resultados obtenidos al aplicar el algoritmo K-means para diferentes números de clusters. Además, se menciona el uso de PCA para la reducción de dimensionalidad en el análisis de datos.

Lisset Zea Monsalve, Julian Zaque Montoya, Brayan Daniel Oviedo Barreto
Estudiantes de Bioingeniería
Universidad de Antioquia
https://www.udea.edu.co/ - (604) 2198332

UNIVERSIDAD
DE ANTIOQUIA