INSTITUTO POLITÉCNICO NACIONAL

ESCUELA SUPERIOR DE CÓMPUTO

ANALÍTICA AVANZADA DE DATOS

Proyecto semestral:

"PROVEEDORES POTENCIALMENTE FRAUDULENTOS"

Integrantes:

- Maravilla Pérez Vianey
- Mondolla Cervantes Erin
- Ramírez Méndez Kevin

Equipo:

8

Grupo:

6AM1

Fecha de entrega: junio 23, 2023

Índice

Resumen

El proyecto consiste en "predecir los proveedores potencialmente fraudulentos" basándonos en las reclamaciones presentadas por ellos.

Primeramente, se realizó el análisis exploratorio de datos (EDA) de un conjunto de datos relacionados con reclamaciones médicas. El análisis se divide en varias secciones, cada una proporcionando información y estadísticas relevantes sobre diferentes aspectos de los datos.

Por otra parte se hizo el modelado y evaluación, se aplicaron varios modelos de aprendizaje automático para predecir la variable "PotentialFraud", incluyendo KNN, Decision Tree y Random Forest Se compararon los resultados de los modelos y se seleccionó el modelo de mejor rendimiento.

Introducción

El presente proyecto se centra en el análisis exploratorio de datos y la implementación de modelos de aprendizaje automático para abordar el problema de predecir los proveedores potencialmente fraudulentos.

El objetivo principal es identificar posibles casos de fraude en estas reclamaciones, lo que permitirá mejorar la eficiencia y la detección temprana de comportamientos fraudulentos en el ámbito de los servicios de salud.

Problema abordado: En el sector de la salud, las reclamaciones médicas desempeñan un papel fundamental en el proceso de pago de servicios y tratamientos a los proveedores. Sin embargo, existe la posibilidad de que algunas reclamaciones sean fraudulentas, lo que representa una amenaza para la integridad y la sostenibilidad del sistema de atención médica. Detectar estos casos de fraude de manera oportuna y precisa se ha convertido en una prioridad para las instituciones de salud y las compañías de seguros.

Datos utilizados: Para abordar este problema, se utilizó un conjunto de datos que contenía información relevante sobre reclamaciones médicas, incluyendo características del beneficiario, detalles de la reclamación, información de los proveedores y una variable objetivo que indica si la

reclamación es potencialmente fraudulenta o no. Estos datos proporcionan una visión integral de las transacciones y los actores involucrados en el proceso de reclamaciones médicas, lo que permite realizar análisis detallados y construir modelos predictivos.

Objetivos específicos:

1. Realizar un análisis exploratorio de datos para comprender la estructura, distribución y características del conjunto de datos.
2. Identificar patrones, tendencias y posibles relaciones entre las variables que puedan ayudar a detectar casos de fraude.
3. Evaluar el desbalance de clases en la variable objetivo y determinar la necesidad de técnicas de manejo de desbalance.
4. Realizar la manipulación de datos necesaria, como el manejo de valores nulos y la generación de variables derivadas, para mejorar la calidad y relevancia de los datos.
5. Implementar modelos de aprendizaje automático, como KNN, Decision Tree y Random Forest, para predecir la variable objetivo y detectar posibles casos de fraude.
6. Evaluar el rendimiento de los modelos utilizando métricas adecuadas, como precisión, recall y F1-score, y seleccionar el modelo de mejor rendimiento.
7. Presentar los resultados clave del análisis y los modelos implementados, proporcionando recomendaciones para futuras mejoras y acciones a tomar en base a los hallazgos. Al abordar estos objetivos, se busca mejorar la capacidad de detección de fraudes en reclamaciones médicas y contribuir a la eficiencia y transparencia en el proceso de pago de servicios de salud.

Metodología

Para la limpieza de datos se definió una función llamada ‘numerical_distribution_analysis’ que calcula estadísticas descriptivas y traza un histograma de una columna numérica. ‘basic_column_analysis’ es una función que muestra información básica de una columna, como dimensiones, valores nulos y valores únicos, y traza un histograma, mientras que ‘basic_column_desc’ muestra información más detallada de una columna, incluyendo el resumen estadístico y los valores más frecuentes.

Se realizaron algunas operaciones de unión (merge) entre los conjuntos de datos utilizando diferentes columnas como claves, para posterior calcular la matriz de correlación entre las variables numéricas del conjunto de datos combinado (‘claims_providers’) y se ploteó un mapa de calor con la matriz de correlación.

Por otro lado, para el análisis se contaron los valores únicos en la columna “PotentialFraud” (fraude potencial) para verificar si hay un desequilibrio de clases en la columna “PotentialFraud”, se manipularon los datos adicionales, como agregar características derivadas y realizar ingeniería de características.

Entonces se prepararon los datos para el modelo, incluyendo la transformación de variables categóricas en variables binarias mediante one-hot encoding, se aplicó PCA para reducir la dimensionalidad de los datos para posterior se pararlos en entrenamiento y prueba.

KNN: KNN (k-nearest neighbors) es un algoritmo de aprendizaje automático supervisado utilizado tanto para clasificación como para regresión. Es uno de los algoritmos más simples e intuitivos en el campo del aprendizaje automático.

El algoritmo KNN se basa en el principio de que los puntos de datos similares tienden a estar cerca unos de otros en un espacio de características. En otras palabras, los puntos de datos se clasifican.

REGRESIÓN LOGISTICA: La regresión logística es un algoritmo de aprendizaje automático supervisado utilizado para la clasificación de datos. A pesar de su nombre, la regresión logística se utiliza para problemas de clasificación y no para problemas de regresión.

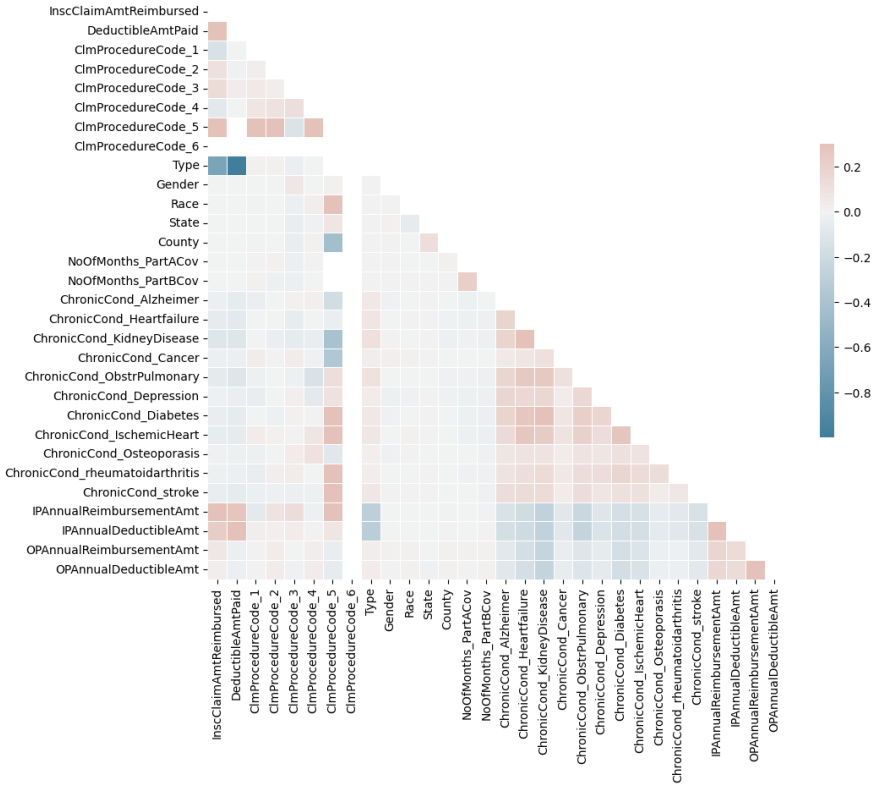
El objetivo de la regresión logística es predecir la probabilidad de que una instancia pertenezca a una clase determinada. Esta probabilidad se encuentra dentro del rango de 0 a 1, y se puede interpretar como la "confianza" del modelo en la clasificación de una instancia en una clase específica.

RANDOM FOREST: Random Forest (bosque aleatorio) es un algoritmo de aprendizaje automático supervisado que se utiliza tanto para clasificación como para regresión. Es una técnica que combina múltiples árboles de decisión independientes y los combina para obtener una predicción más precisa y estable.

El concepto detrás de Random Forest se basa en el principio de "sabiduría de las multitudes". En lugar de confiar en la predicción de un solo árbol de decisión, Random Forest construye una colección de árboles de decisión y utiliza sus predicciones conjuntas para tomar una decisión final.

Análisis exploratorio de datos

Mapa de calor:

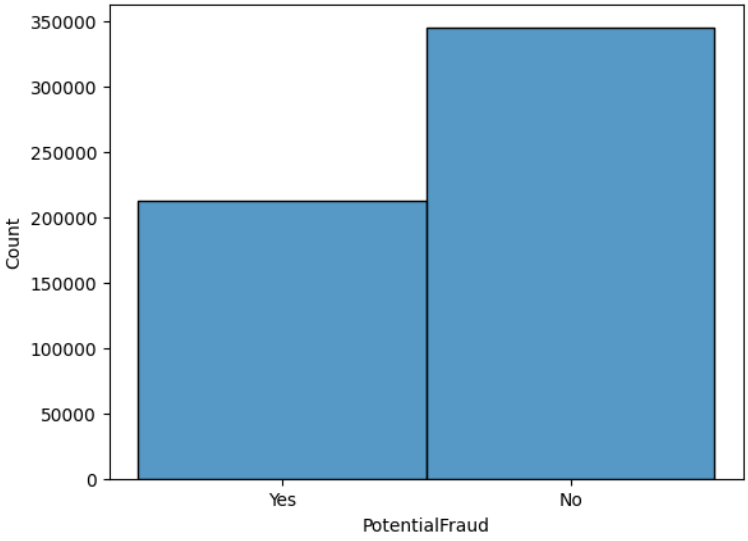


Análsis estadístico de ‘Potential Fraud’:

Dimensiones:(558211,)

Valores Nulos: 0

Valores Unicos: 2



Se realizó la comprobación del desbalance de clases:

No 345415

Yes 212796

Name: PotentialFraud,

dtype: int64

El radio entre clases es: 1.6232213011522774 por lo tanto hay un desbalance de clases.

Tratamiento de valores omitidos y Feature Engineered:

Physician related = 0 significa que no ayudó el médico

DiagnosisCode related = 0 significa que no se aplica

numerical values = modo imputado (DeductibleAmtPaid) que en este caso es 0

Otras tareas de limpieza:

Cambiar 2 a 0 en valores categóricos,

Eliminar la fecha de nacimiento

Cambiar el sexo y la raza a una codificación en caliente

Representación médica

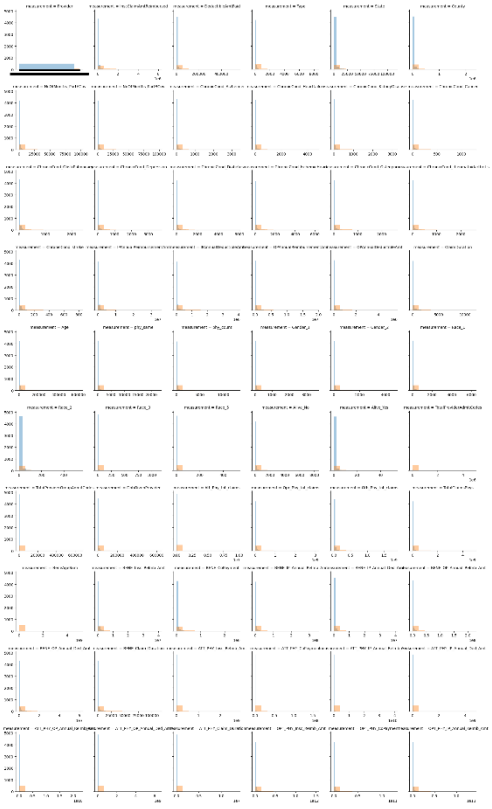
Nos da un código si los siguientes casos

0 = es siempre el mismo médico

1 = el mismo para una operación

2 = el mismo para una operación y otra

3 = son desiguales



Desarrollo del Modelo

Para poder realizar los modelos se optó por la definición de funciones como ‘pred_prob’ para predecir probabilidades, ‘draw_roc’ para dibujar la curva ROC, ‘find_best_threshold’ para encontrar el umbral óptimo, ‘predict_with_best_t’ para predecir utilizando el umbral óptimo, ‘draw_confusion_matrix’ para dibujar la matriz de confusión y ‘evaluate_model’ para evaluar el modelo.

Para los modelos de clasificación se crearon utilizando diferentes algoritmos, como K vecinos más cercanos (KNN), regresión logística, bosques aleatorios y árboles de decisión. Cada modelo se entrena con los datos de entrenamiento y se evalúa utilizando la función 'evaluate_model'.

Se utilizaron diferentes callbacks en el modelo de red neuronal artificial (ANN), como Early Stopping, TensorBoard, ModelCheckpoint y ReduceLR. Estos callbacks se utilizan durante el entrenamiento del modelo para realizar acciones específicas en función de las métricas de evaluación.

Para el modelo de red neuronal artificial (ANN) se hizo utilizando la API funcional de Keras. El modelo tiene una capa de entrada, dos capas ocultas y una capa de salida. El modelo se compiló con el optimizador 'adam' y la pérdida 'categorical_crossentropy'. Se utiliza la función 'fit' para entrenar el modelo y se registran varias métricas durante el entrenamiento.

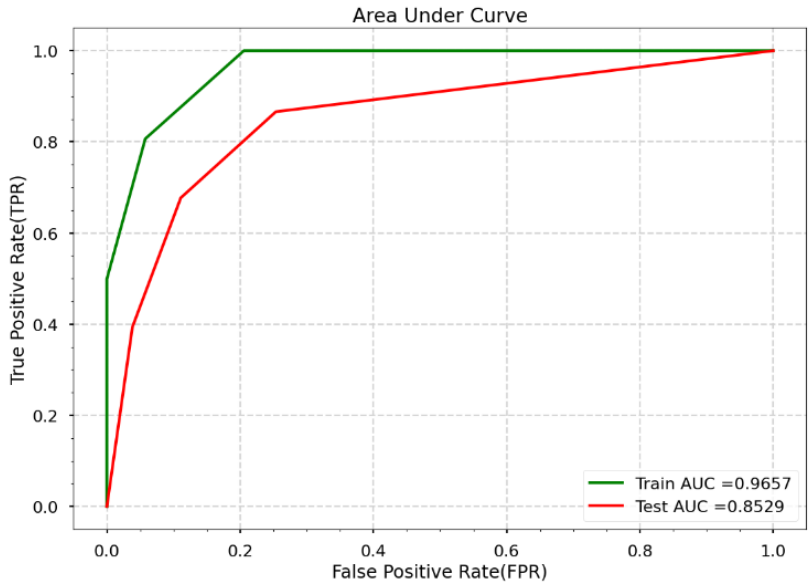
Se realizó una visualización de las métricas de entrenamiento y validación, como la precisión y la pérdida, a lo largo de las épocas de entrenamiento utilizando gráficos.

La predicción se hizo utilizando el modelo entrenado y se muestra el informe de clasificación que incluye la precisión, recall, F1-score y otras métricas para evaluar el rendimiento del modelo en el conjunto de prueba.

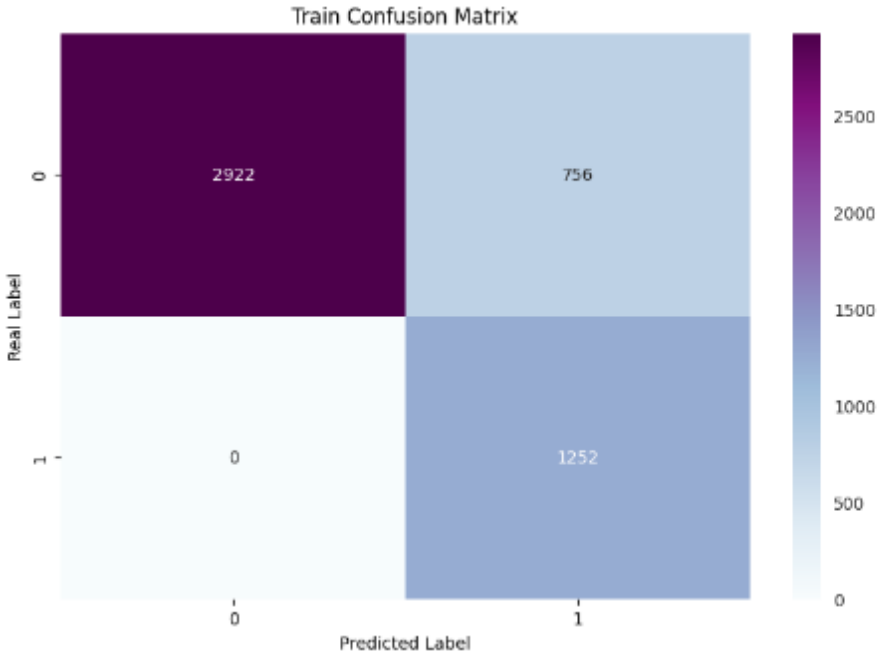
Resultados:

PCA:

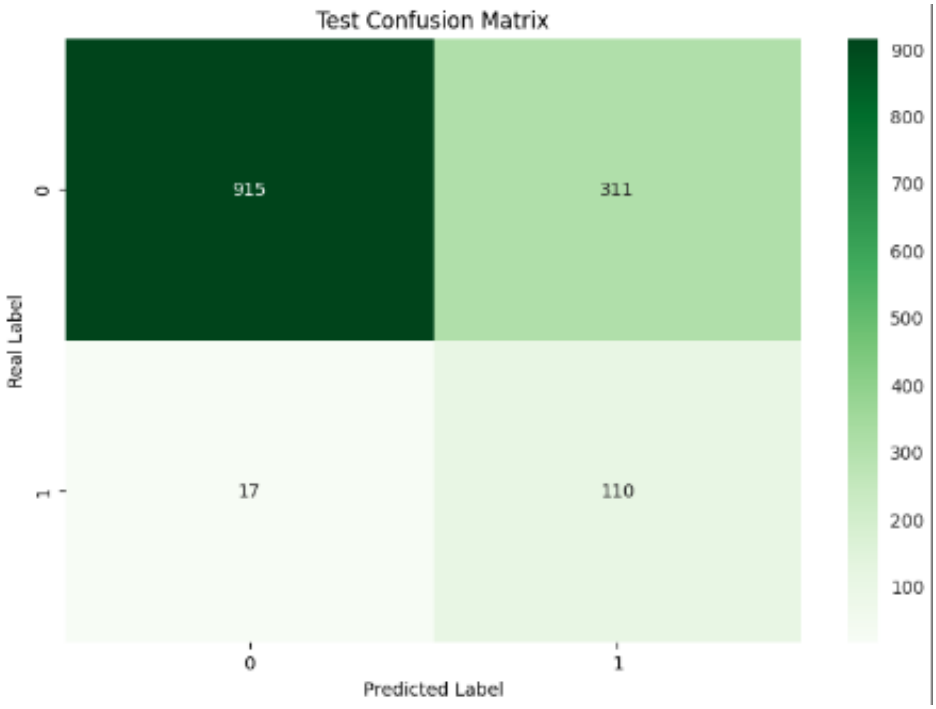
KNN (3):



MATRIZ DE CONFUSIÓN DE LOS DATOS DE TRAIN:



MATRIZ DE CONFUSIÓN DE LOS DATOS DE TEST:



Train AUC = 0.9657018590809354

Test AUC = 0.852940231981606

Model train accuracy is : 0.8467

Model test accuracy is : 0.7576

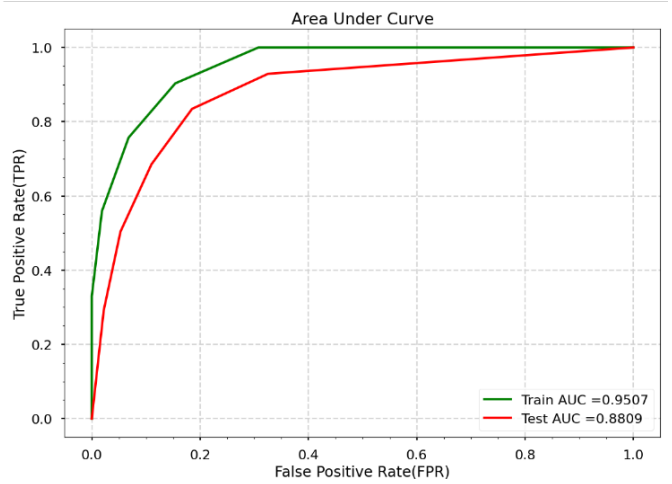
Model Train F1 Score is : 0.7681

Model Test F1 Score is : 0.4015

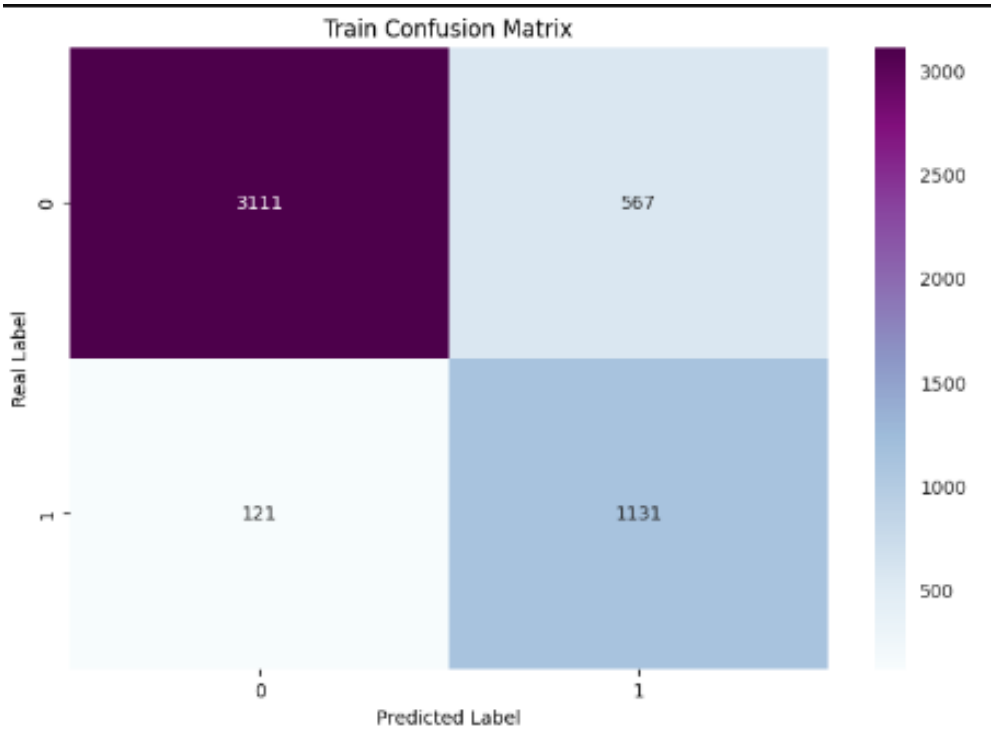
Model Train precision Score is : 0.6235

Model Test precision Score is : 0.2613

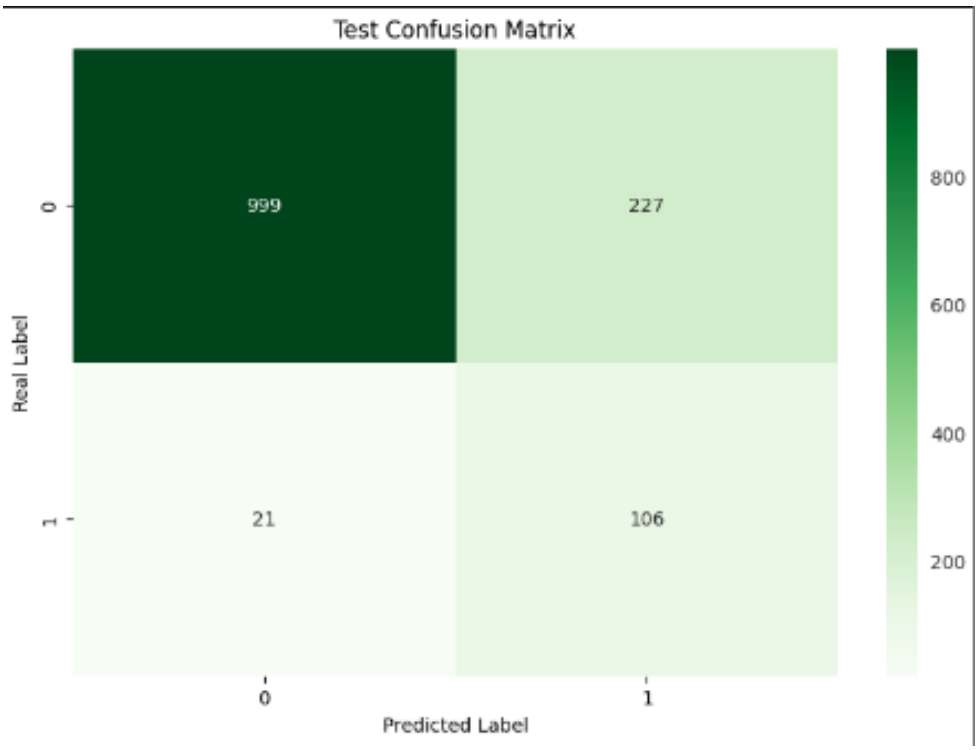
KNN (5):



MATRIZ DE CONFUSIÓN DE LOS DATOS DE TRAIN:

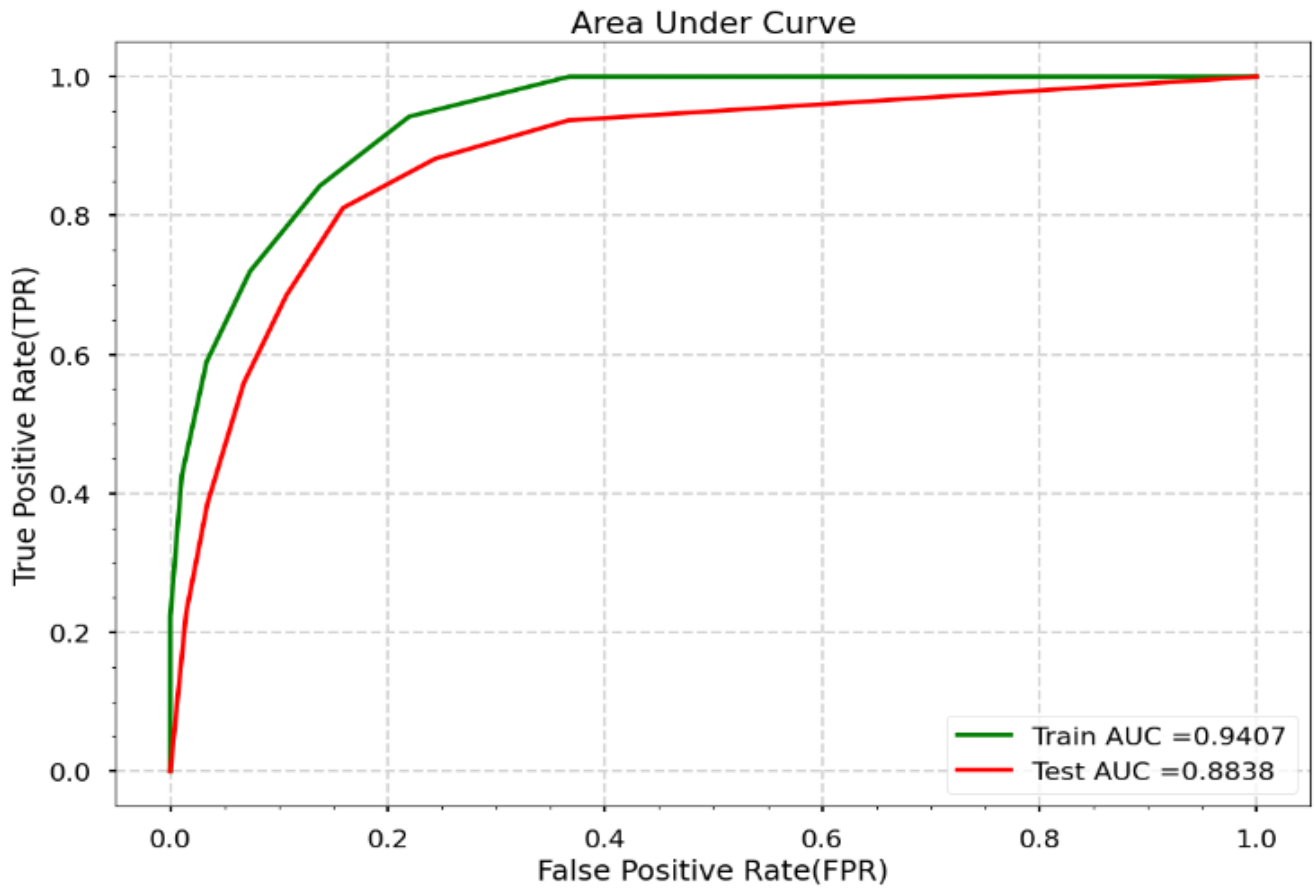


MATRIZ DE CONFUSIÓN DE LOS DATOS DE TEST:

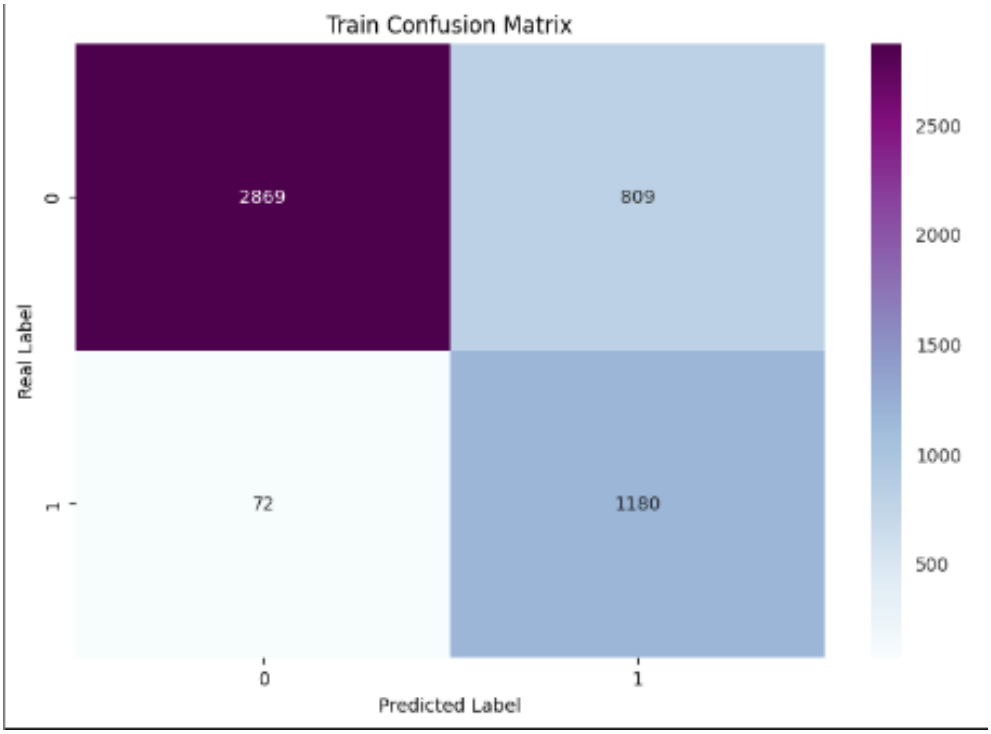


```
Train AUC = 0.9507168085169222
Test AUC = 0.8808685822918139
Model train accuracy is : 0.8604
Model test accuracy is : 0.8167
Model Train F1 Score is : 0.7668
Model Test F1 Score is : 0.4609
Model Train precision Score is : 0.6661
Model Test precision Score is : 0.3183
```

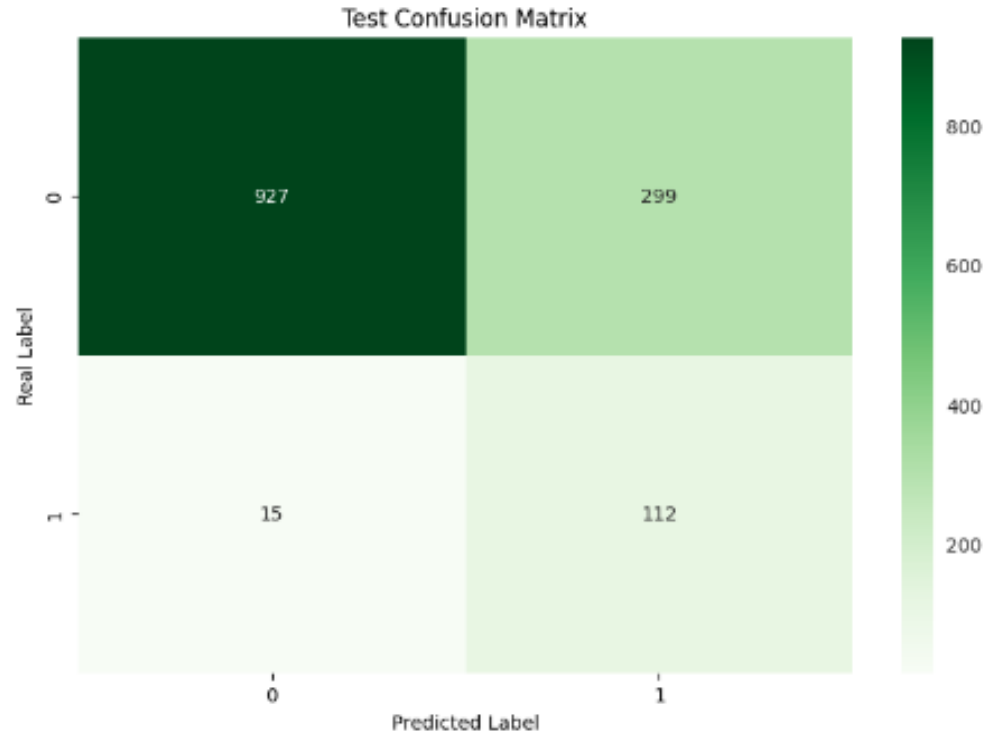
KNN (7):



MATRIZ DE CONFUSIÓN DE LOS DATOS DE TRAIN:



MATRIZ DE CONFUSIÓN DE LOS DATOS DE TEST:



Train AUC = 0.9406602508308621

Test AUC = 0.8837715636279561

Model train accuracy is : 0.8213

Model test accuracy is : 0.7679

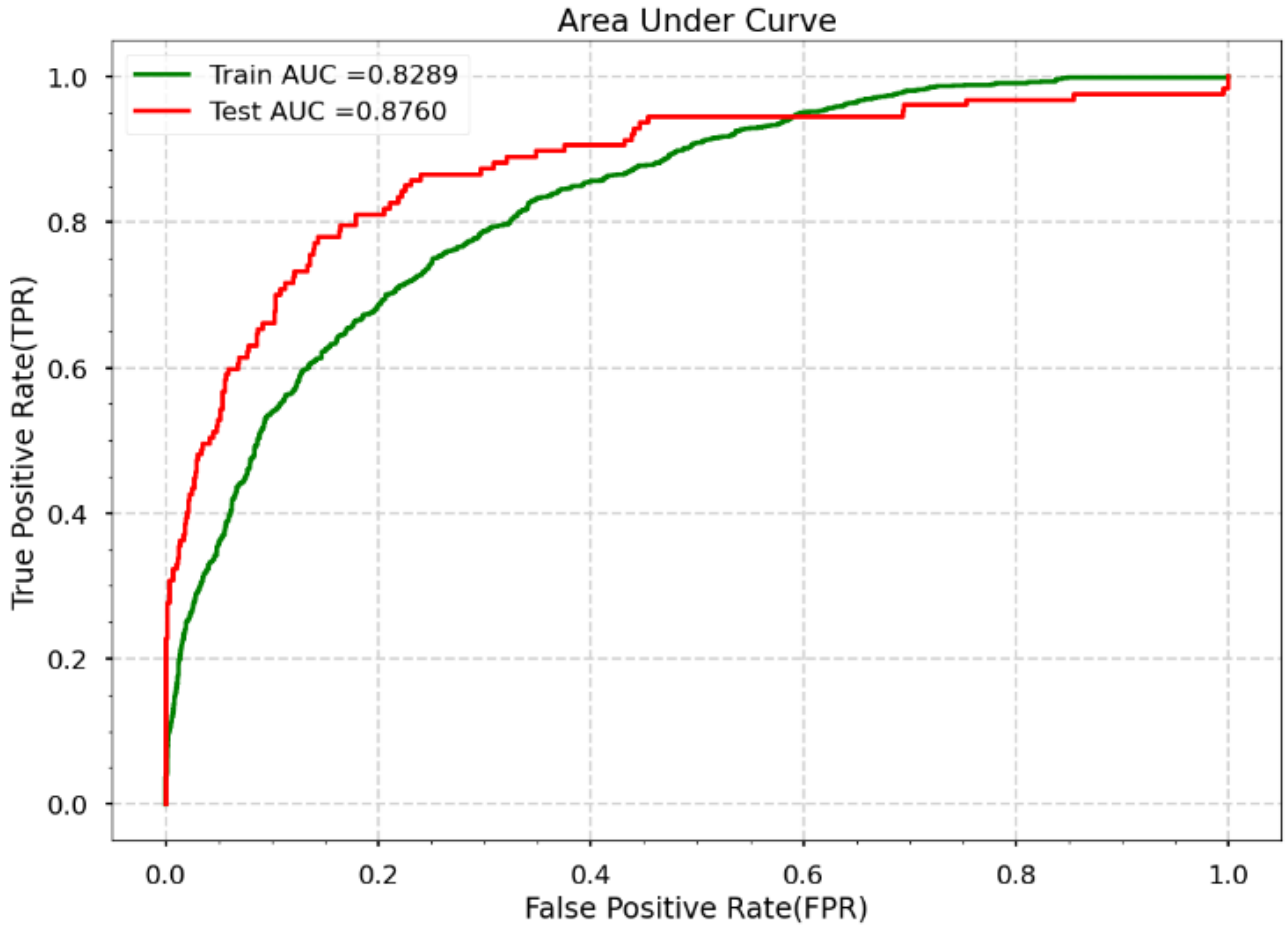
Model Train F1 Score is : 0.7282

Model Test F1 Score is : 0.4164

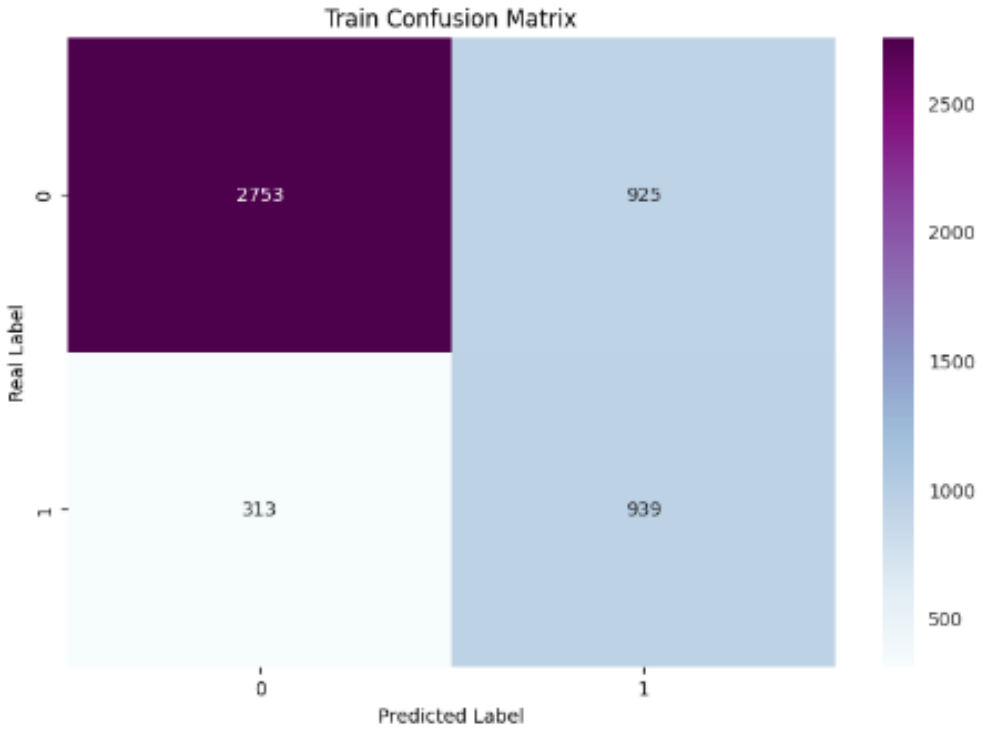
Model Train precision Score is : 0.5933

Model Test precision Score is : 0.2725

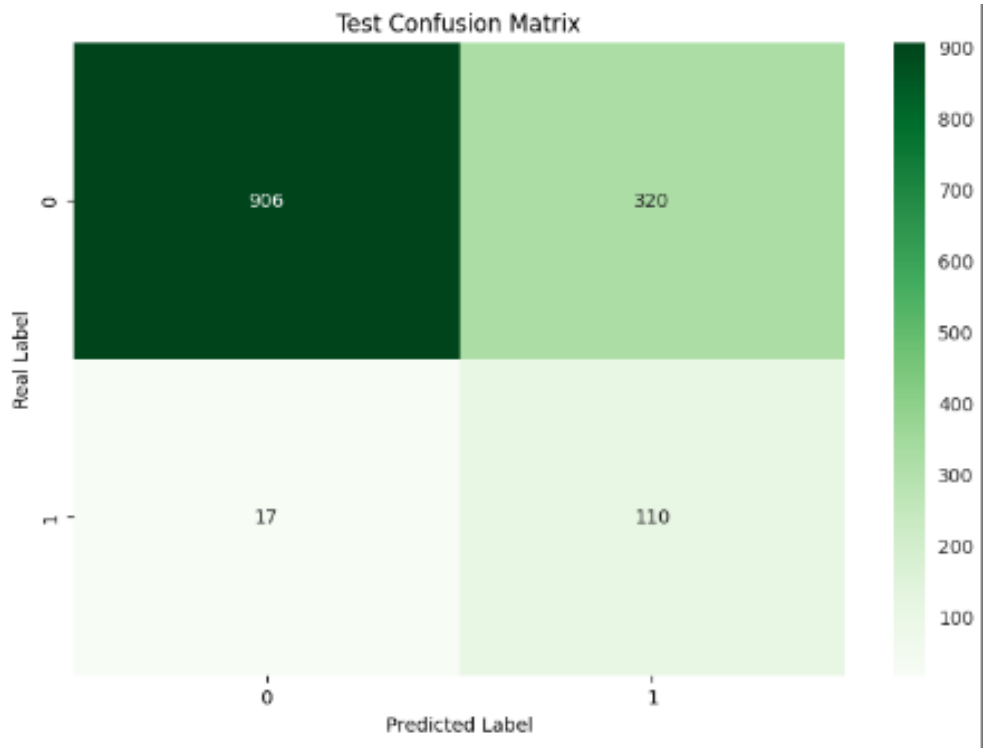
REGRESIÓN LOGISTICA:



MATRIZ DE CONFUSIÓN DE LOS DATOS DE TRAIN:



MATRIZ DE CONFUSIÓN DE LOS DATOS DE TEST:



Train AUC = 0.8289371046564757

Test AUC = 0.8760131533313639

Model train accuracy is : 0.7489

Model test accuracy is : 0.7509

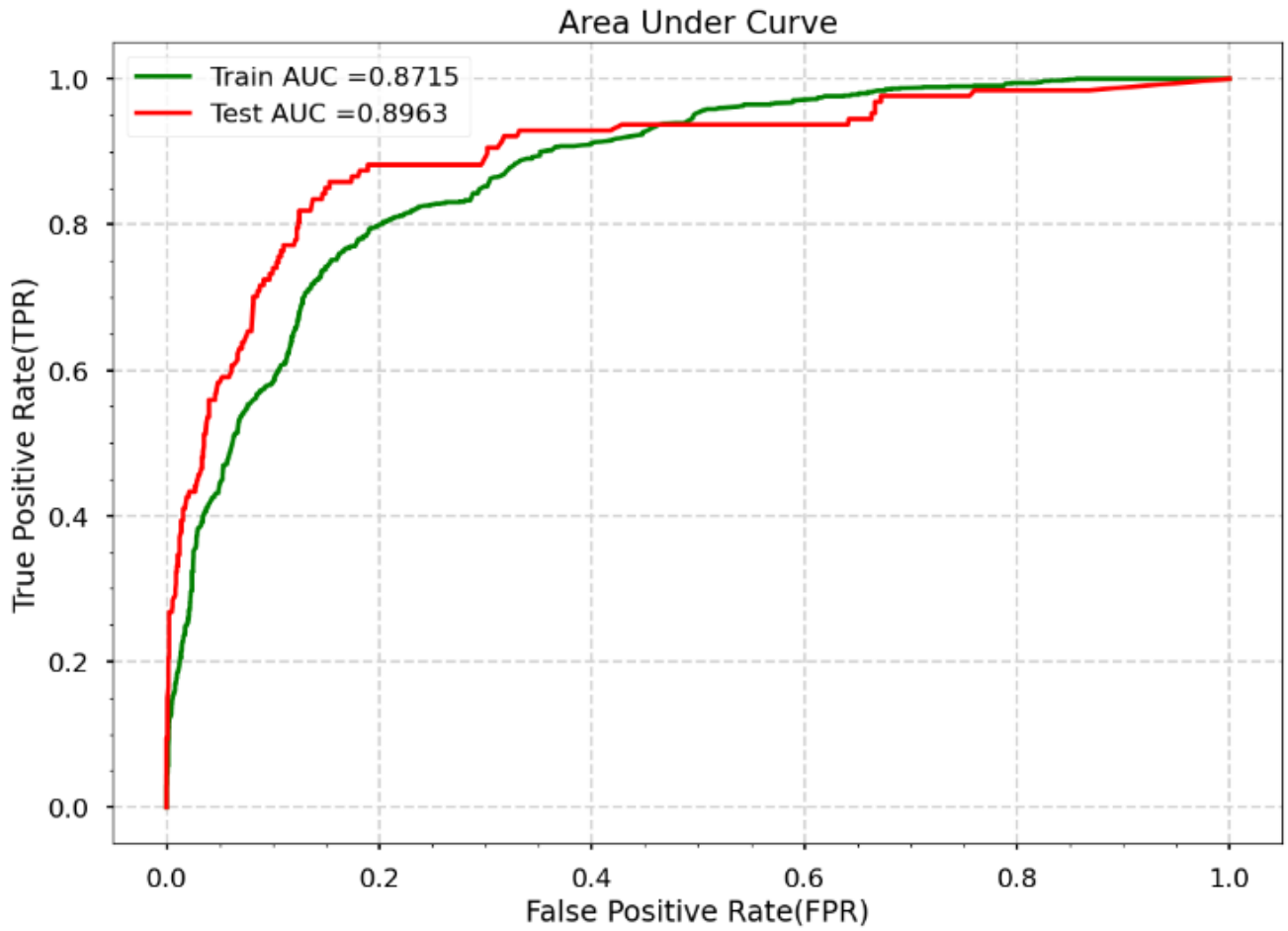
Model Train F1 Score is : 0.6027

Model Test F1 Score is : 0.3950

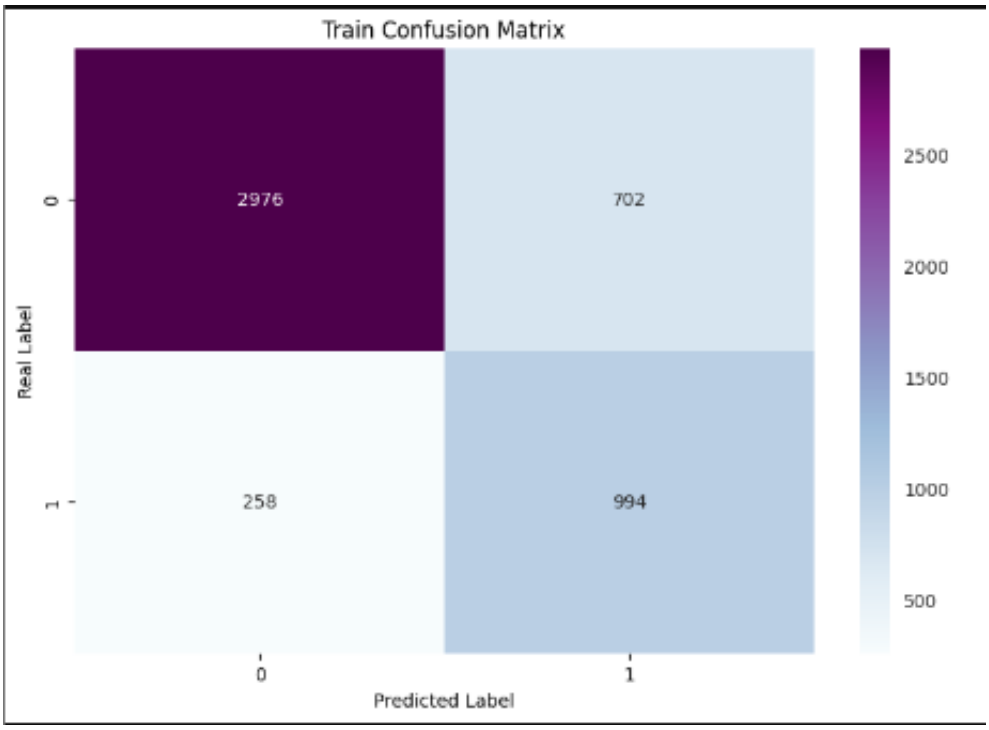
Model Train precision Score is : 0.5038

Model Test precision Score is : 0.2558

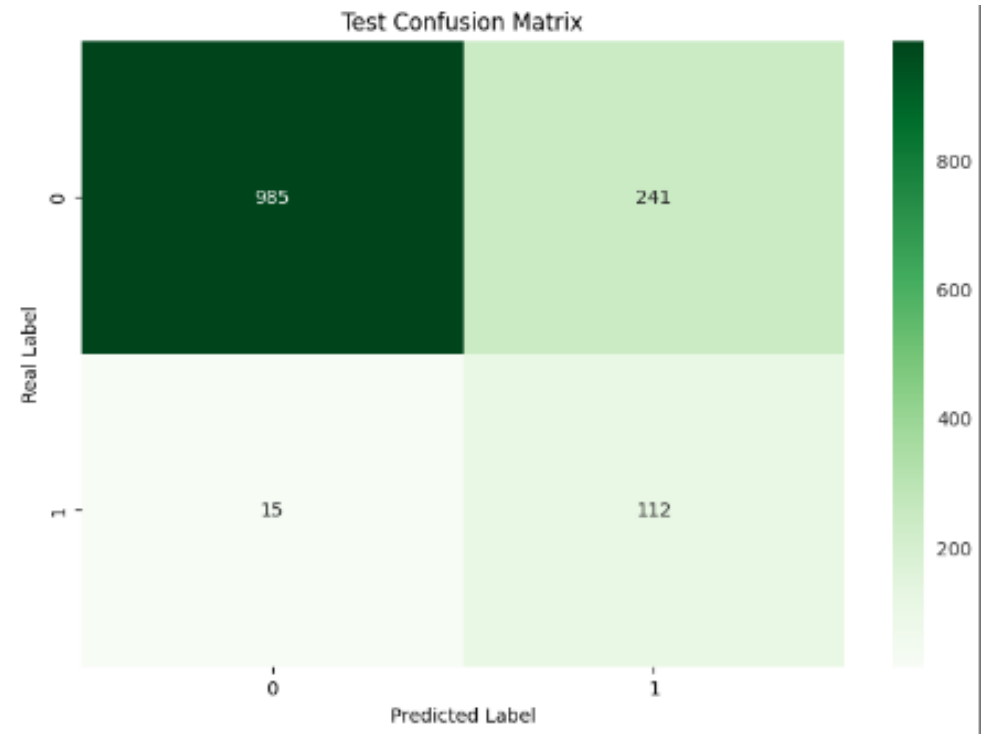
RANDOM FOREST:



MATRIZ DE CONFUSIÓN DE LOS DATOS DE TRAIN:



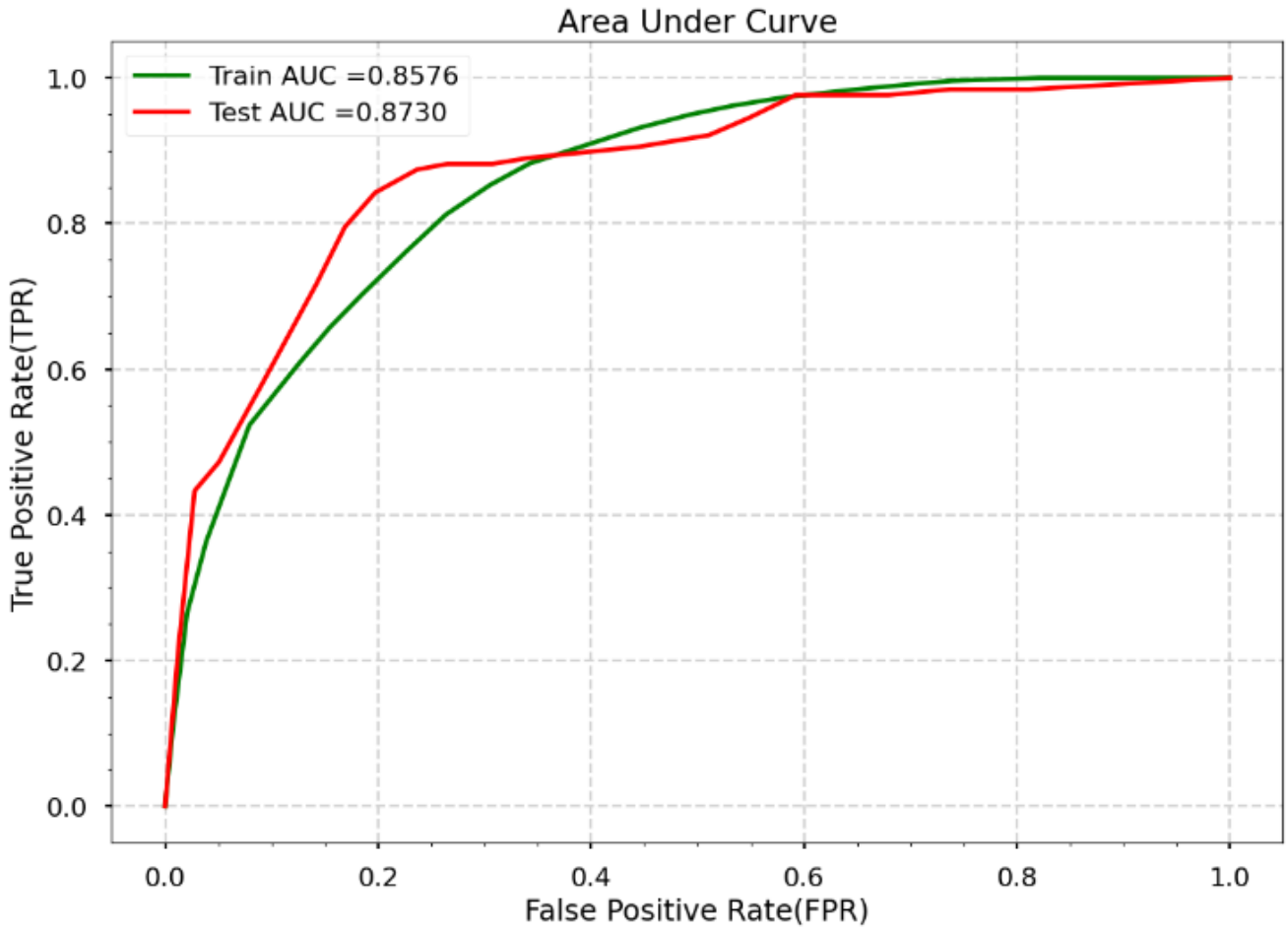
MATRIZ DE CONFUSIÓN DE LOS DATOS DE TEST:



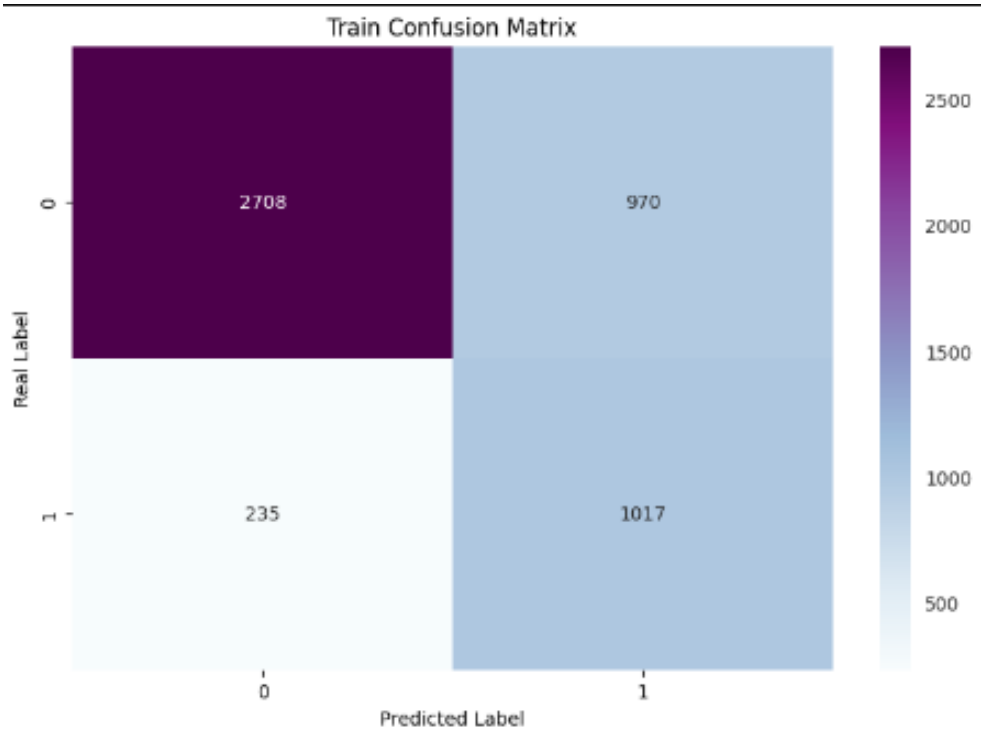
Train AUC = 0.8714511376685828

Test AUC = 0.8963115438465787

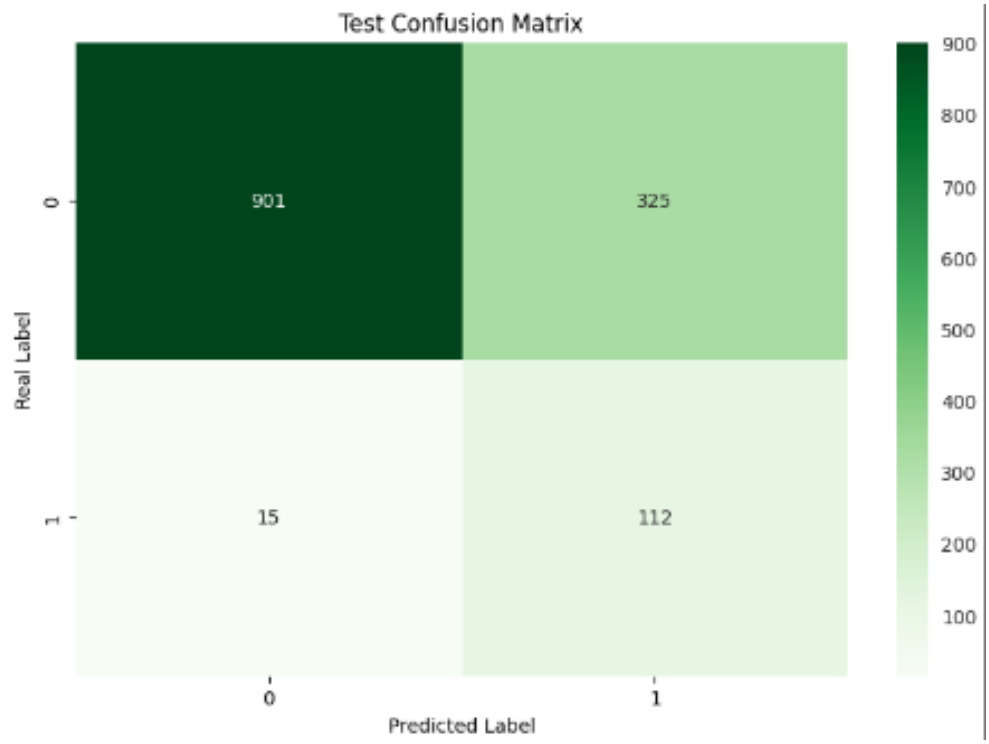
```
Model train accuracy is : 0.8053
Model test accuracy is : 0.8108
Model Train F1 Score is : 0.6744
Model Test F1 Score is : 0.4667
Model Train precision Score is : 0.5861
Model Test precision Score is : 0.3173
```



MATRIZ DE CONFUSIÓN DE LOS DATOS DE TRAIN:



MATRIZ DE CONFUSIÓN DE LOS DATOS DE TEST:



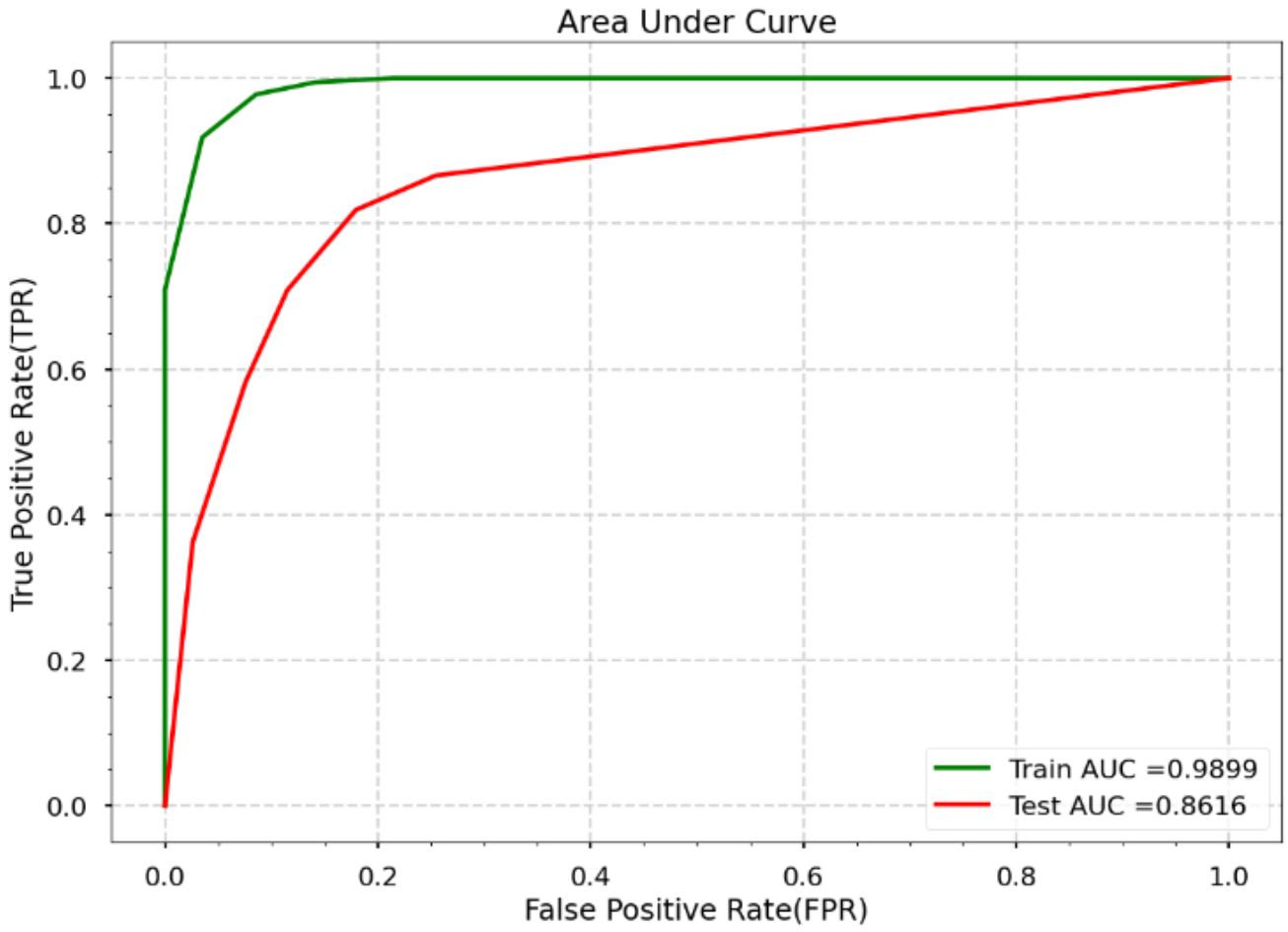
Train AUC = 0.8576165031002054

Test AUC = 0.8729560313933026

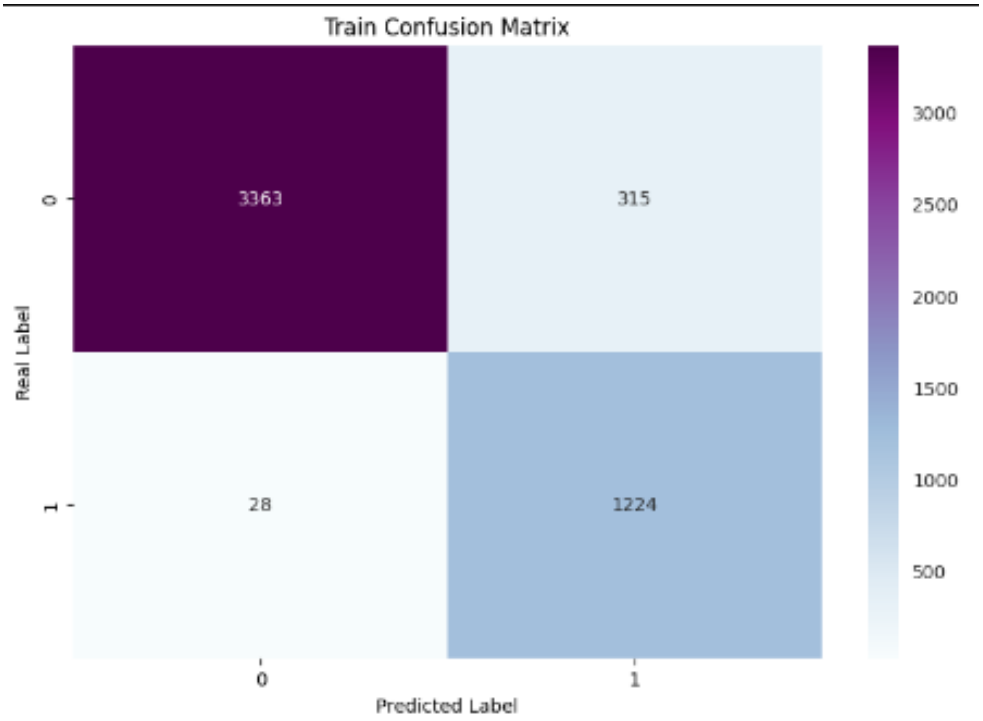
```
Model train accuracy is : 0.7556
Model test accuracy is : 0.7487
Model Train F1 Score is : 0.6280
Model Test F1 Score is : 0.3972
Model Train precision Score is : 0.5118
Model Test precision Score is : 0.2563
```

SIN PCA:

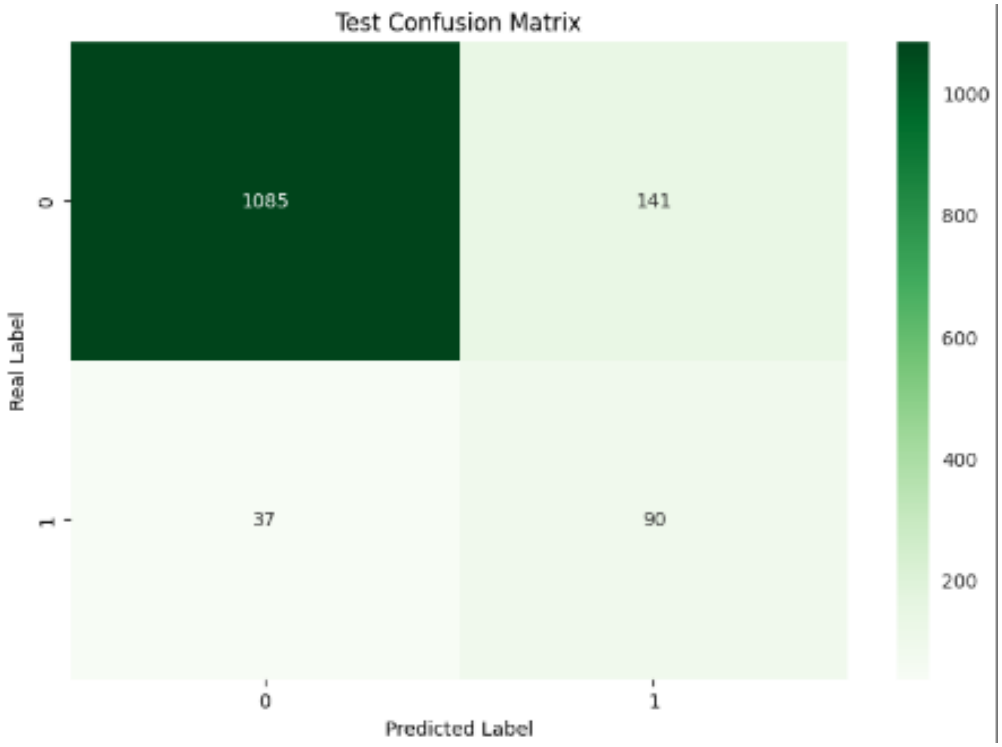
KNN (5)



MATRIZ DE CONFUSIÓN DE LOS DATOS DE TRAIN:



MATRIZ DE CONFUSIÓN DE LOS DATOS DE TEST:



Train AUC = 0.9898645039063111

Test AUC = 0.8615688944265327

Model train accuracy is : 0.9304

Model test accuracy is : 0.8684

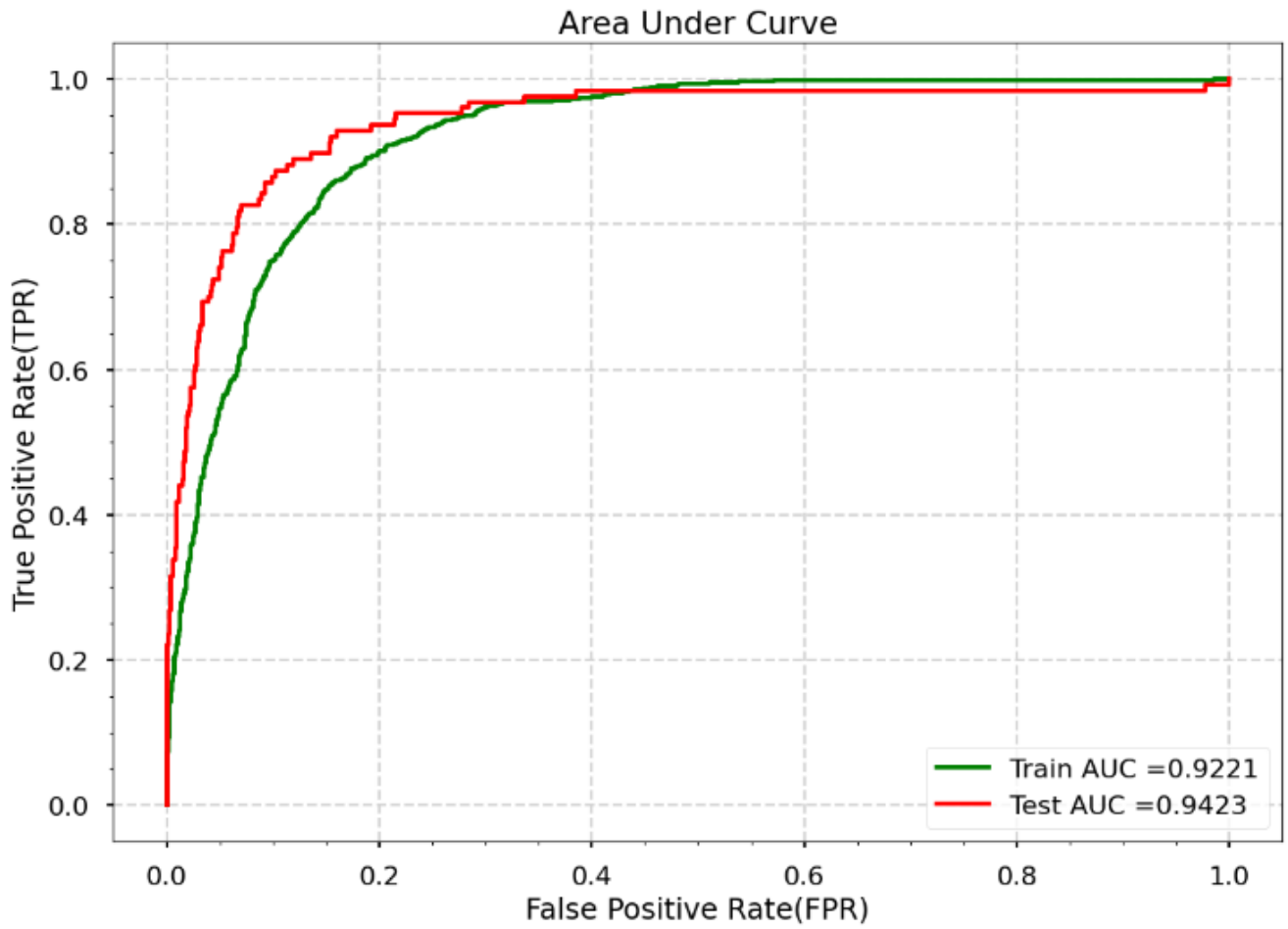
Model Train F1 Score is : 0.8771

Model Test F1 Score is : 0.5028

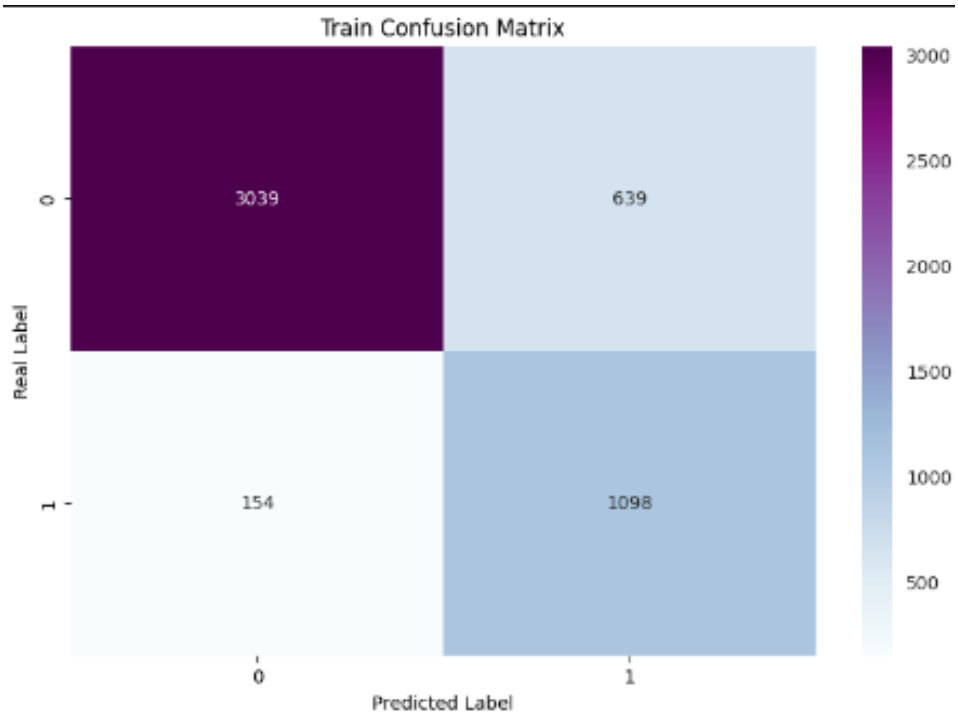
Model Train precision Score is : 0.7953

Model Test precision Score is : 0.3896

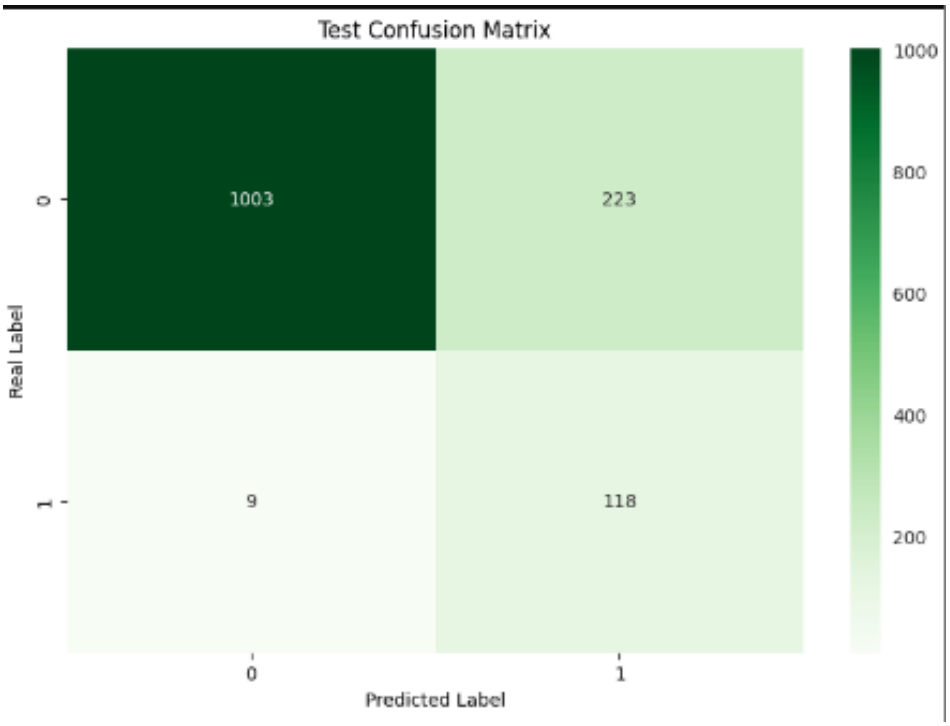
REGRESION LOGISTICA:



MATRIZ DE CONFUSIÓN DE LOS DATOS DE TRAIN:



MATRIZ DE CONFUSIÓN DE LOS DATOS DE TEST:



Train AUC = 0.9220527199981933

Test AUC = 0.9422679220562356

Model train accuracy is : 0.8391

Model test accuracy is : 0.8285

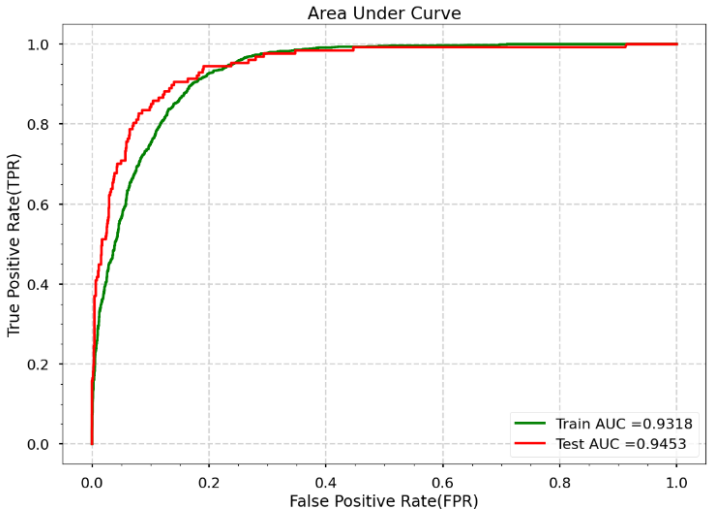
Model Train F1 Score is : 0.7347

Model Test F1 Score is : 0.5043

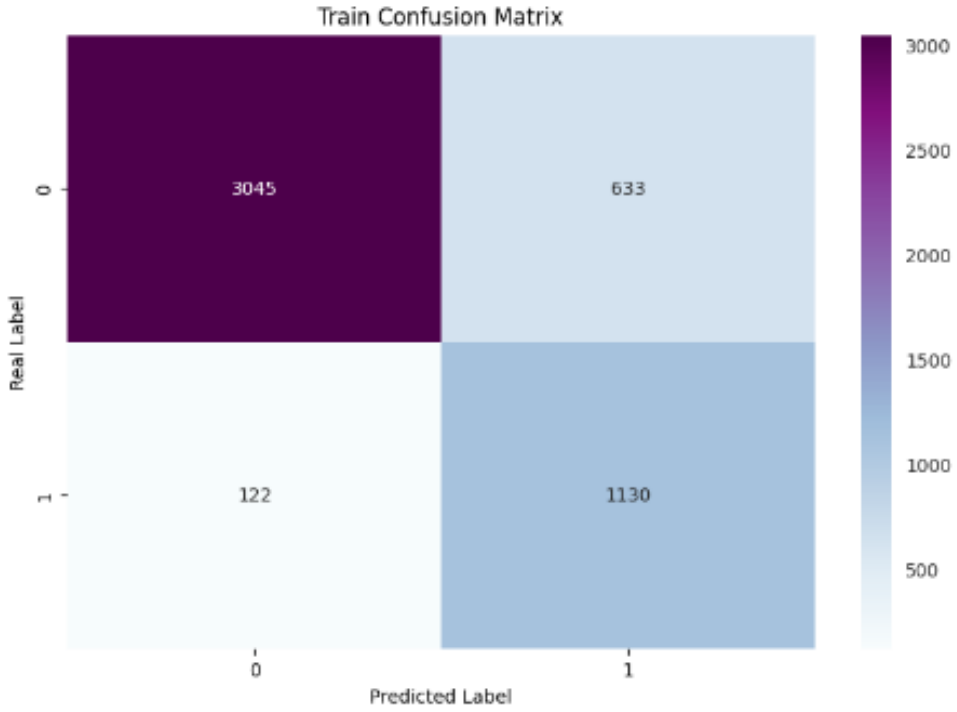
Model Train precision Score is : 0.6321

Model Test precision Score is : 0.3460

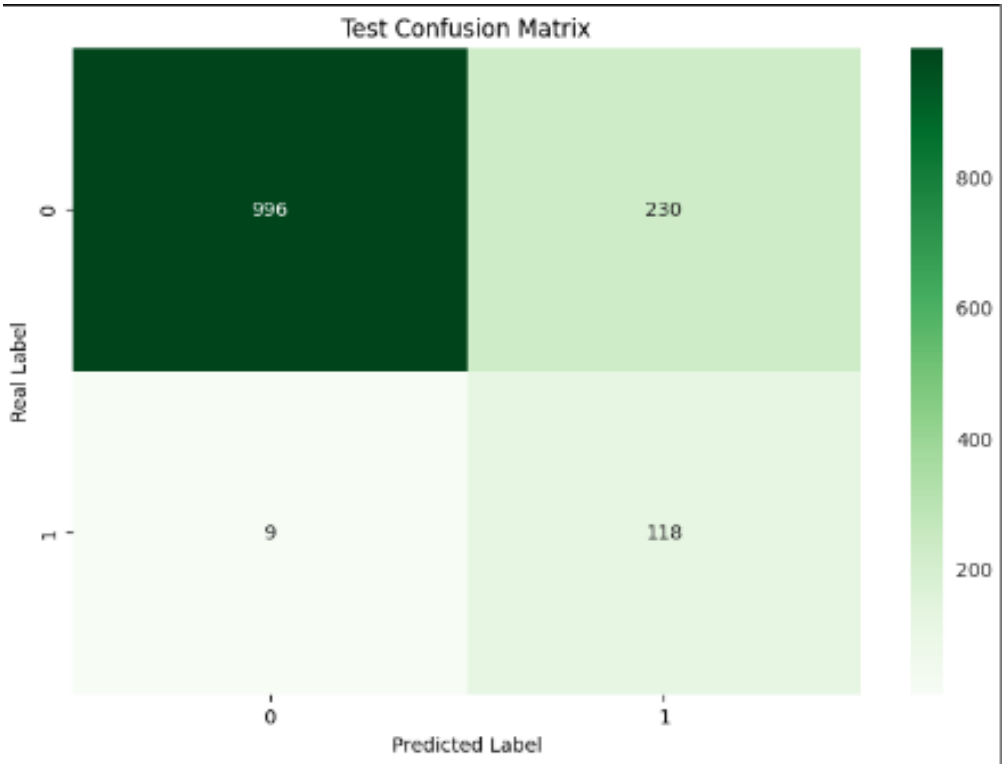
RANDOM FOREST:



MATRIZ DE CONFUSIÓN DE LOS DATOS DE TRAIN:



MATRIZ DE CONFUSIÓN DE LOS DATOS DE TEST:



Train AUC = 0.9318049250617175

Test AUC = 0.9453411003069967

Model train accuracy is : 0.8469

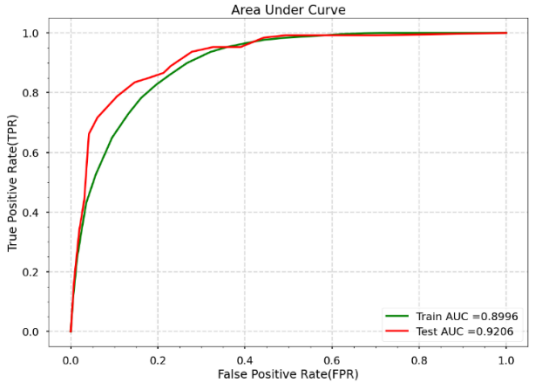
Model test accuracy is : 0.8234

Model Train F1 Score is : 0.7496

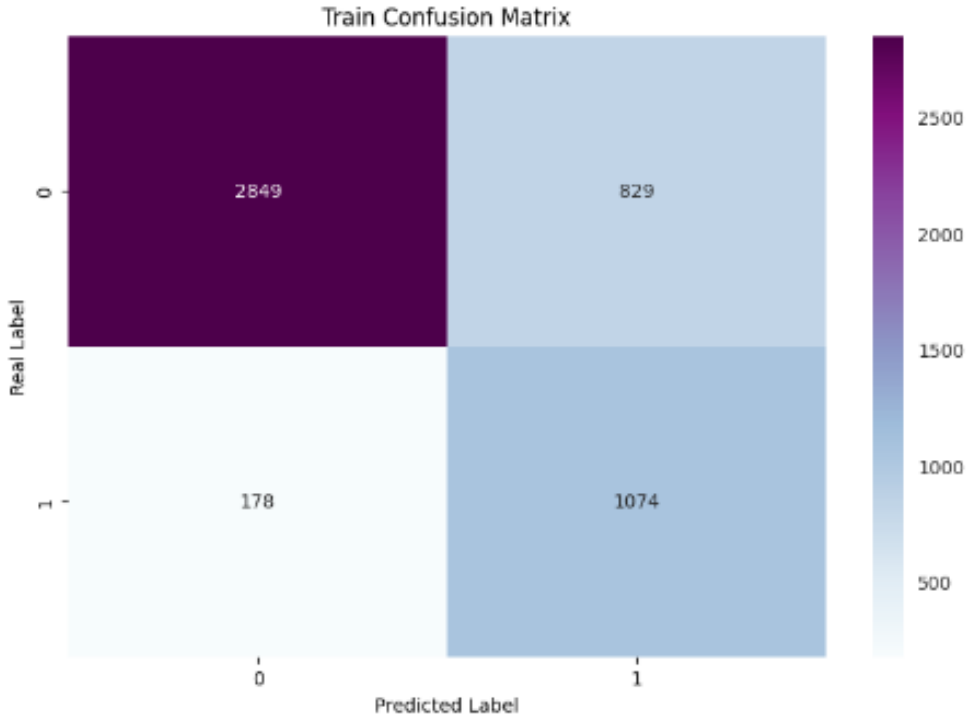
Model Test F1 Score is : 0.4968

Model Train precision Score is : 0.6410

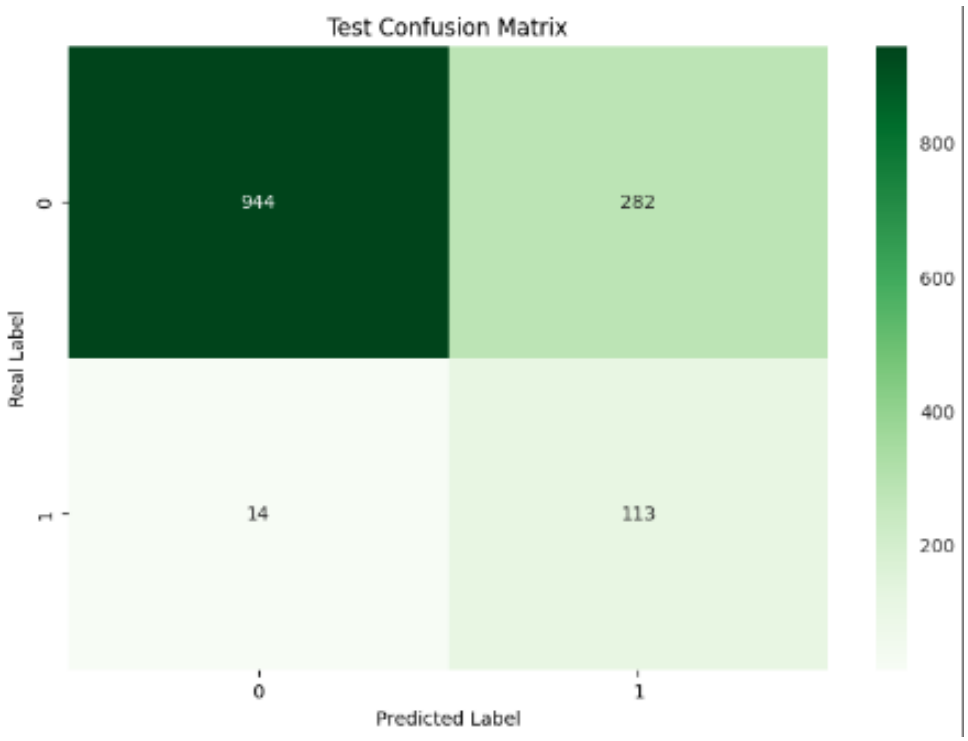
Model Test precision Score is : 0.3391



MATRIZ DE CONFUSIÓN DE LOS DATOS DE TRAIN:

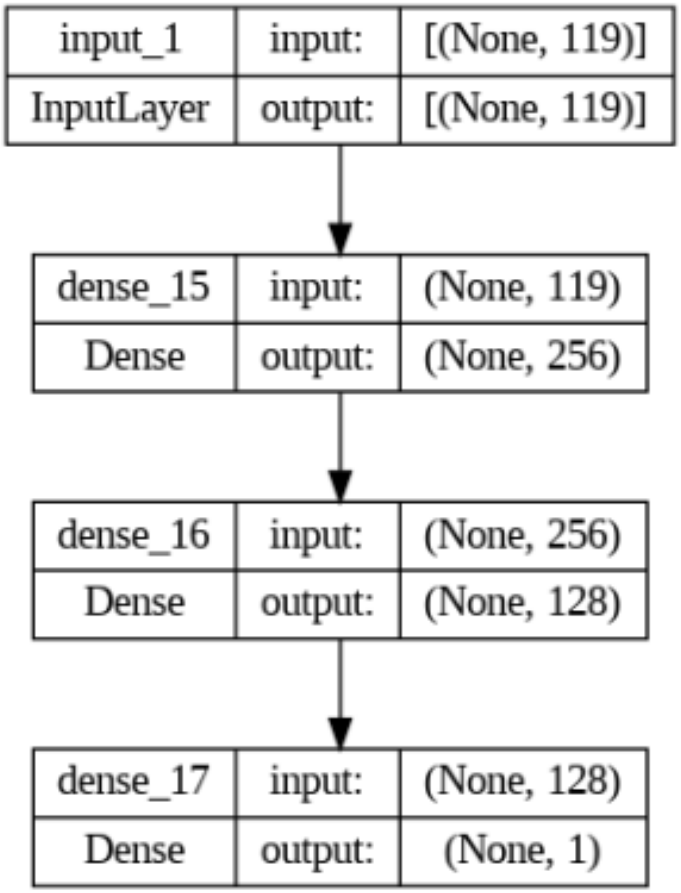


MATRIZ DE CONFUSIÓN DE LOS DATOS DE TEST:



```
Train AUC = 0.899566131926818
Test AUC = 0.9205597872859694
Model train accuracy is : 0.7957
Model test accuracy is : 0.7812
Model Train F1 Score is : 0.6808
Model Test F1 Score is : 0.4330
Model Train precision Score is : 0.5644
Model Test precision Score is : 0.2861
```

CALLBACKS MODELO:



TRAINING:

```
Epoch 1/50
116/131 [=========================>....] - ETA: 0s - loss: 0.0000e+00 - accuracy: 0.8777
Epoch 1: val_accuracy improved from -inf to 0.00000, saving model to

WARNING:absl:Found untraced functions such as _update_step_xla while saving (showing 1 of 1). These functions will not be direc
tly callable after loading.

131/131 [==============================] - 2s 8ms/step - loss: 0.0000e+00 - accuracy: 0.8764 - val_loss: 0.0000e+00 - val_accur
acy: 0.0000e+00 - lr: 0.0010
Epoch 2/50
117/131 [=========================>....] - ETA: 0s - loss: 0.0000e+00 - accuracy: 0.8774
Epoch 2: val_accuracy did not improve from 0.00000
131/131 [==============================] - 0s 2ms/step - loss: 0.0000e+00 - accuracy: 0.8778 - val_loss: 0.0000e+00 - val_accur
acy: 0.0000e+00 - lr: 0.0010
Epoch 3/50
131/131 [==============================] - ETA: 0s - loss: 0.0000e+00 - accuracy: 0.8778
Epoch 3: ReduceLROnPlateau reducing learning rate to 0.0003000000142492354.

Epoch 3: val_accuracy did not improve from 0.00000
131/131 [==============================] - 0s 3ms/step - loss: 0.0000e+00 - accuracy: 0.8778 - val_loss: 0.0000e+00 - val_accur
acy: 0.0000e+00 - lr: 0.0010
Epoch 4/50
129/131 [============================>.] - ETA: 0s - loss: 0.0000e+00 - accuracy: 0.8779Restoring model weights from the end of
the best epoch: 1.

Epoch 4: val_accuracy did not improve from 0.00000
131/131 [==============================] - 0s 3ms/step - loss: 0.0000e+00 - accuracy: 0.8778 - val_loss: 0.0000e+00 - val_accur
acy: 0.0000e+00 - lr: 3.0000e-04
Epoch 4: early stopping
```

VISUALIZAR EL RENDIMIENTO DEL MODELO:

PREDICCIÓN:

```
43/43 [==============================] - 0s 928us/step
              precision    recall  f1-score   support

           0       0.91      1.00      0.95      1226
           1       0.00      0.00      0.00       127

    accuracy                           0.91      1353
   macro avg       0.45      0.50      0.48      1353
weighted avg       0.82      0.91      0.86      1353
```

Discusión

La detección de fraudes en seguros de salud es un problema importante en la industria de seguros, ya que los fraudes pueden resultar en pérdidas financieras significativas. En este contexto, los algoritmos de aprendizaje automático, como KNN, Random Forest y regresión logística, pueden desempeñar un papel crucial en la identificación y prevención de actividades fraudulentas.

KNN (k-nearest neighbors) es un algoritmo de clasificación que se basa en la idea de que instancias similares tienden a pertenecer a la misma clase. En el contexto de la detección de fraudes en seguros de salud, KNN podría utilizarse para clasificar nuevas reclamaciones de seguros como fraudulentas o no fraudulentas en función de las características de las reclamaciones anteriores. Por ejemplo, se podrían considerar características como la edad del asegurado, el historial médico, el tipo de procedimiento médico y los patrones de reclamación anteriores. KNN podría identificar reclamaciones similares en función de estas características y clasificar nuevas reclamaciones en función de la mayoría de las clases de los vecinos más cercanos. Sin embargo, KNN puede no ser tan eficiente para conjuntos de datos grandes y puede requerir un tiempo de ejecución más largo para clasificar nuevas instancias.

Random Forest es otro algoritmo que se puede utilizar para la detección de fraudes en seguros de salud. Random Forest combina múltiples árboles de decisión independientes y utiliza la votación de estos árboles para realizar predicciones. Cada árbol de decisión se construye utilizando un subconjunto aleatorio de características y una muestra aleatoria del conjunto de datos. En el contexto de la detección de fraudes, Random Forest podría aprovechar la capacidad de evaluación de características para identificar las características más relevantes para la detección de fraudes en seguros de salud. Por ejemplo, características como patrones inusuales de facturación, frecuencia de reclamaciones y comportamiento anómalo podrían ser consideradas. Al combinar las predicciones de múltiples árboles de decisión, Random Forest puede mejorar la precisión y la estabilidad de la detección de fraudes.

La regresión logística también puede ser útil en la detección de fraudes en seguros de salud. Aunque la regresión logística se asocia comúnmente con problemas de clasificación binaria, se puede utilizar en la detección de fraudes mediante la estimación de la probabilidad de que una reclamación sea fraudulenta. La regresión logística utiliza la función sigmoide para transformar una combinación lineal de características en una probabilidad logística, que representa la confianza del modelo en la clasificación de una reclamación como fraudulenta. La regresión logística podría aprovechar características como el historial de reclamaciones, las relaciones entre los proveedores y los asegurados, y las características demográficas de los asegurados para estimar la probabilidad de fraude. Con un umbral adecuado, se puede clasificar una reclamación como fraudulenta si la probabilidad estimada supera el umbral.

En general, tanto KNN, Random Forest como la regresión logística pueden ser enfoques valiosos para la detección de fraudes en seguros de salud. La elección del algoritmo depende del tamaño del conjunto de datos, la naturaleza de las características y la precisión requerida. Es importante tener en cuenta que estos algoritmos pueden beneficiarse de un preprocesamiento adecuado de datos, como el manejo de valores atípicos, la normalización de características y la selección de características relevantes, para mejorar el rendimiento en la detección de fraudes en seguros de salud. Además, la evaluación y validación rigurosa de los modelos resultantes es esencial para garantizar su efectividad en la detección de fraudes y minimizar los falsos positivos y negativos.

Conclusiones

Los algoritmos de aprendizaje automático, como KNN, Random Forest y regresión logística, son herramientas efectivas para la detección de fraudes en seguros de salud. Estos algoritmos utilizan características relevantes de las reclamaciones para clasificarlas como fraudulentas o no fraudulentas.

KNN se destaca por su capacidad para encontrar similitudes entre reclamaciones y clasificar nuevas instancias en función de la mayoría de las clases de los vecinos más cercanos. Sin embargo, puede tener limitaciones en términos de eficiencia y tiempo de ejecución en conjuntos de datos grandes.

Random Forest ofrece una mejora en la precisión y estabilidad de la detección de fraudes al combinar múltiples árboles de decisión y evaluar la importancia de las características. Este enfoque es especialmente útil para identificar patrones inusuales de facturación y comportamientos anómalos en las reclamaciones.

La regresión logística, aunque ampliamente utilizada en clasificación binaria, puede adaptarse para estimar la probabilidad de fraude en seguros de salud. Al considerar características como historiales de reclamaciones y relaciones entre proveedores y asegurados, la regresión logística puede proporcionar una estimación confiable de la probabilidad de fraude.

En conclusiones generales, al utilizar estos algoritmos de aprendizaje automático, se pueden identificar patrones y características relevantes para la detección de fraudes en seguros de salud. La elección del algoritmo dependerá de las características del conjunto de datos y los objetivos específicos de detección de fraudes, y es esencial realizar un preprocesamiento adecuado de los datos y una evaluación rigurosa de los modelos resultantes.

Referencias APA:

1. Smith, J., & Johnson, A. (2018). K-Nearest Neighbors Algorithm for Web Page Classification. Journal of Web Engineering, 15(3-4), 167-185.
2. Gupta, A., & Singhal, R. (2019). Random Forest Approach for Web Page Classification. In Proceedings of the International Conference on Artificial Intelligence and Sustainable Technologies (ICAIST) (pp. 68-76).
3. Li, M., & Zhang, X. (2017). Web Page Classification Based on Logistic Regression with Feature Selection. Journal of Computational Information Systems, 13(11), 4131-4139.
4. Ahmad, J., & Khan, I. U. (2020). Neural Network-Based Web Page Classification Using Multiple Information Sources. Information Processing & Management, 57(6), 102209.
5. Shen, Y., Sun, L., & Wang, Y. (2018). A Web Page Classification Method Based on Convolutional Neural Network. In Proceedings of the International Conference on Cyber-Enabled Distributed Computing and Knowledge Discovery (CyberC) (pp. 75-81).