“Año del Fortalecimiento de la Soberanía Nacional”

Trabajo final de Machine Learning y Deep Learning con Python

(Nivel 2)

Autores:

Bayona Hernández, Juan Diego

Chunga Castilla, José Augusto

Coello Mejía, Gustavo Adolfo

Huamán Paredes, María Sofía

Asesor:

Ing. Pedro Rotta Saavedra

Piura 2022

CONTENIDO

INTRODUCCIÓN

El manejo y análisis de bases de datos son considerados fundamentales para una buena organización tanto industrial como estadístico. A medida que pasa el tiempo, las empresas comienzan a crecer en el mercado y la data con la que muchos de ellos trabajan se va convirtiendo cada vez más tediosa. Esto conlleva a problemas de ineficiencia e improductividad en los empleados al no saber cómo interpretar estas grandes datas. Hoy en día, los lenguajes de programación han facilitado el manejo de Big Data, gracias a que su adaptabilidady su capacidad de creación de funciones han otorgado a los trabajadores una herramienta capaz de mejorar la eficiencia en su producción, además de un buen manejo de la información a la hora de tomar decisiones en muchos casos.

Uno de los lenguajes de programación más conocido es Python. Este lenguaje nos permite realizar un análisis de Machine Learning que ayudan a optimizar el manejo de las bases de datos. En el presente informe, se desarrollará un sistema de programación el cual generará una mejora en el análisis de datos. La finalidad de dicho programa es el de facilitar al empleado la actualización de los datos, la filtración de dicha base y la de poder fabricar gráficas estadísticas que sean de fácil interpretaciónpara la toma de decisiones.

Las oportunidades de mejora con la programación son sustanciales, por lo cual este ejemplo nos ayuda a observar las ventajas competitivas que puede tener una empresa con personas que manejen el lenguaje de programación de Python. A medida que se va requiriendo mayores funciones para optimizar, el código puede ser mejorado en muchos casos. Lo cual se puede inferir que el uso de un lenguaje de programación brinda una mejora constante para cualquier empresa. Además, que facilita el trabajo de los trabajadores y ayuda a mejorar la productividad y la eficiencia.

ANÁLISIS DEL PROBLEMA

Para este trabajo tenemos como objetivo desarrollar un análisis de Machine Learning con respecto al data set “app.csv” en el cual se va desarrollando una serie de algoritmos que van entrenando adecuadamente nuestro programa para así tener una predicción más eficiente. El buen entrenamiento del programa va proporcionando al programa de una mayor autonomía.

El foco principal de nuestro programa es el siguiente. El programa lo que busca analizar es como las transacciones de los clientes han ido aumentando o disminuyendo en cada una de las aplicaciones de interfaz de pagos unificados. Esto nos ayuda a evaluar como el valor de seguridad y de la facilidad en el uso de la aplicación afectan positiva o negativamente en los ingresos y montos en los aplicativos. Las apps UPI (apps de interfaz de pagos unificados) se han convertido en herramientas esenciales para las personas. Y el consumidor lo que busca es en esa aplicación es tener la confianza y la seguridad de que no existe vulnerabilidad alguna en dichas aplicaciones. Estos factores de control por cada empresa juegan un papel muy importante en el uso de sus respectivos aplicativos. En la data analizada observamos como la afluencia y dispersión de los montos de los clientes ha ido cambiado a lo largo de los años.

Otro punto importante en el desarrollo de este informe es que las características de la data app.csv son las siguientes. Tiene 4 variables numéricas que son: volumen de transacciones por cliente, valor monetario por cliente, volumen de transacciones total y valor monetario total. Posee los meses y años de la data expuesta; además de la señalización de la app UPI que se esté tipificando en la data. La data presentada tiene una frecuencia mensual. Además, que tiene un detallado especifico de la moneda que se utiliza para cada una de las variables numéricas.

La importancia de la data a evaluar es que nos proporciona una data frecuente del comportamiento e impacto de las aplicaciones UPI en los clientes. Su frecuencia mensual nos proporciona una mejora en la predicción en el corto plazo. Además de una buena proyección del largo plazo debito a la cantidad de dato que existe. El análisis se vuelve más enriquecedor al tener una data mensual ya que los impactos pueden ser previstos e identificados en un plazo corto; en vez de tener una data trimestral o semestral que puede tener observaciones sesgadas de las apps debido a factores externos que pueden existir y ensucian la veracidad de la data.

ANÁLISIS DEL RESULTADO

Haciendo un análisis de trade off entre sesgo y varianza nos encontramos que los cuatros modelos de encuentran en el mismo parámetro entre 0 y 20. Por lo cual, para un análisis optimo podemos utilizar cualquiera de los 4 modelos. Se podría considerar un Random Forest debido a que una de las salidas del modelo es la importancia de variables. Pero también, debido a la similitud de los resultados por metodología otro modelo optimo sería el modelo KNN debido a que es bueno para detectar anomalías. Para este caso creemos conveniente utilizar el modelo Random Forest debido a que la importancia de cada variable es importante en identificar.

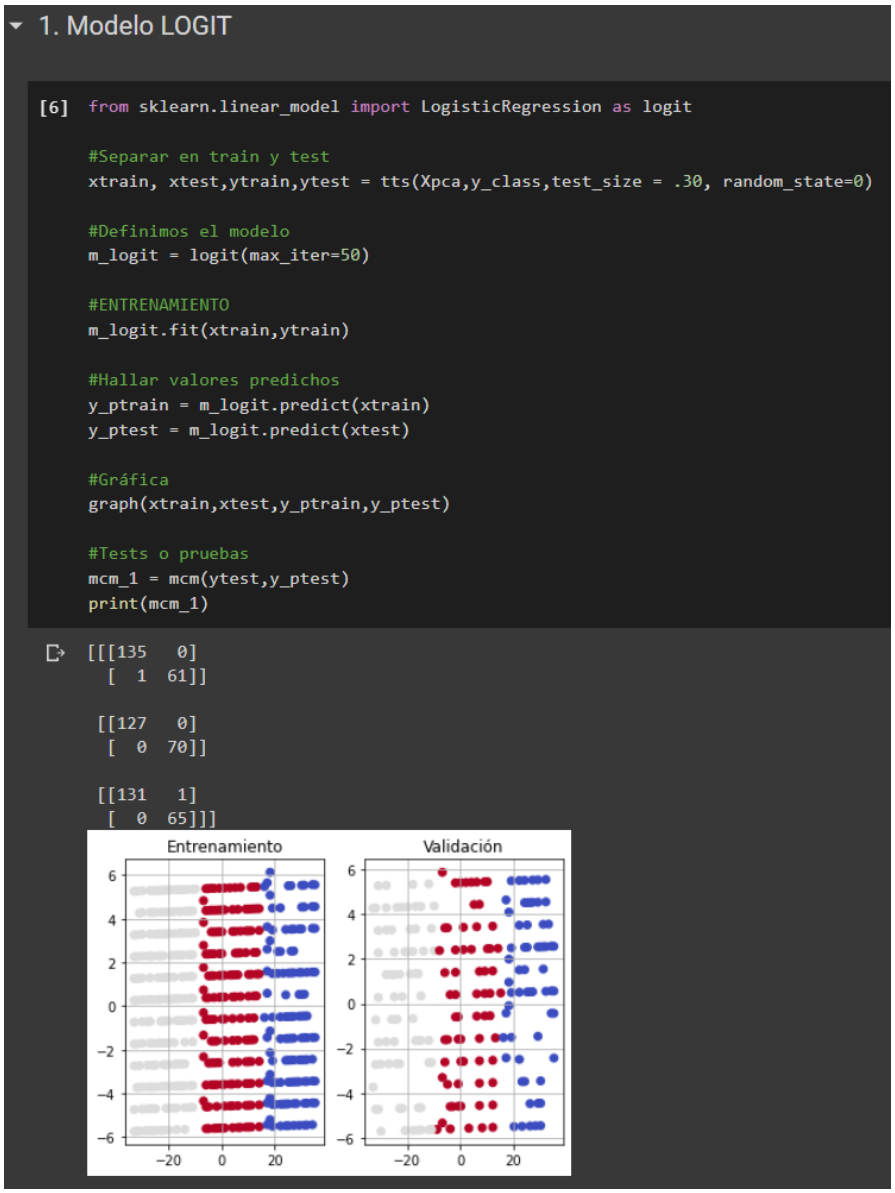

Figure 1: Modelo Regresión Logística

2. Random Forest

```
from sklearn.ensemble import RandomForestClassifier as rfc

##Separación de la base de datos
xtrain,xtest,ytrain,ytest = tts(Xpca,y_class,test_size =.30, random_state= 0)

#Definir modelo
RF = rfc(n_estimators=200)

#Entrenar
RF.fit(xtrain,ytrain)

#Predecir
y_ptrain = RF.predict(xtrain)
y_ptest = RF.predict(xtest)

#Gráfica
graph(xtrain,xtest,y_ptrain,y_ptest)

#Tests o pruebas
mcm_2 = mcm(ytest,y_ptest)
print(mcm_2)
```

```
[[[135   0]
  [  1  61]]

 [[126   1]
  [  0  70]]

 [[131   1]
  [  1  64]]]
```

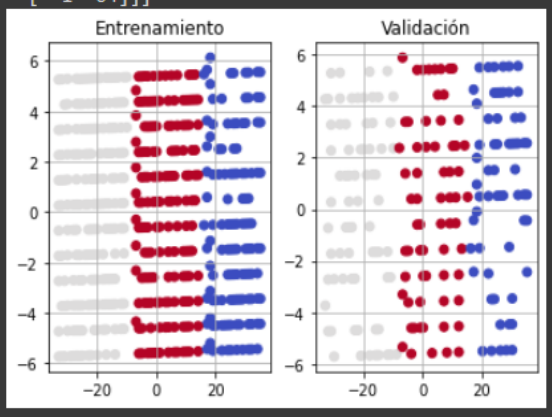

Figure 2: Modelo Random Forest

3. SVM

```
from sklearn.svm import SVC

##Separación de la base de datos
xtrain,xtest,ytrain,ytest = tts(Xpca,y_class,test_size =.30, random_state = 0)

#Definir modelo
svc = SVC(gamma = 'scale')

#Entrenar
svc.fit(xtrain,ytrain)

#Predecir
y_ptrain = svc.predict(xtrain)
y_ptest = svc.predict(xtest)

#Multilabel Confusion Matrix
mcm_3 = mcm(ytest,y_ptest)
print(mcm_3)

#Gráfica
graph(xtrain,xtest,y_ptrain,y_ptest)
```

```
[[[135   0]
  [  1  61]]

 [[127   0]
  [  0  70]]

 [[131   1]
  [  0  65]]]
```

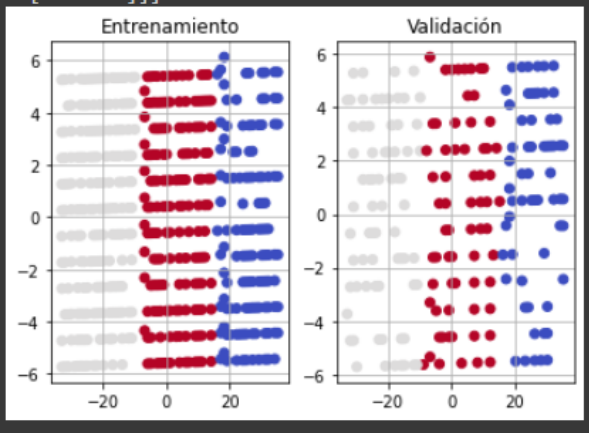

Figure 3: Modelo SVM

4. Modelo KNN

```
from sklearn.neighbors import KNeighborsClassifier as knn

#Separación
xtrain,xtest,ytrain,ytest = tts(Xpca,y_class,test_size =.3, random_state = 0)

#Modelación y Entrenamiento
modelo_knn = knn(n_neighbors=3)
modelo_knn.fit(xtrain, ytrain)
yptrain_3 = modelo_knn.predict(xtrain)
yptest_3 = modelo_knn.predict(xtest)

#VALIDACIÓN
mcm_4 = mcm(ytest,yptest_3)
print(mcm_4)

#Gráfico
graph(xtrain,xtest,yptrain_3,yptest_3)
```

```
[[[135   0]
  [  1  61]]

 [[126   1]
  [  0  70]]

 [[131   1]
  [  1  64]]]
```

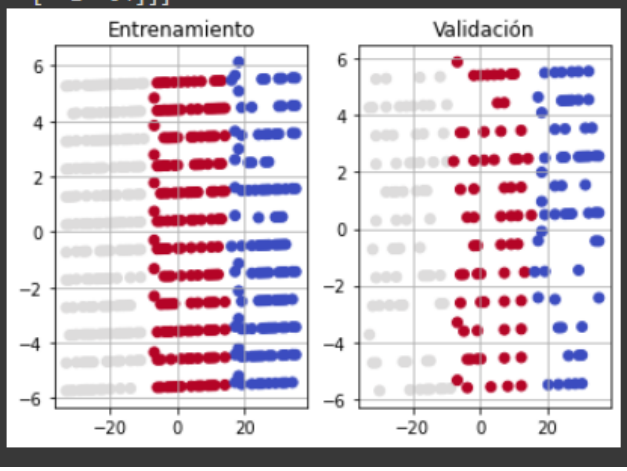

Figure 4: Modelo KNN

CONCLUSIONES

i. La facilidad que nos proporciona Python para trabajar con datos nos permite realizar un análisis a partir de gráficos generados. con Seaborn podemos hacer que la visualización sea la parte central de la exploración y comprensión de los datos, el módulo de pandas nos permite obtener un análisis estadístico y descriptivo. Y el módulo Matplotlib genera gráficos a partir de datos contenidos en listas o arrays. En el caso de Plotly nos ofrece visualizaciones complejas y sofisticadas de los datos en una interfaz gráfica. En conjunto todos estos comandos acompañados de Machine Learning nos ayuda a mejorar la compresión y el análisis de los datos.

ii. Machine Learning a través de algoritmos puede identificar patrones en datos masivos y proporciona un análisis predictivo, lo realizado en el curso nos ha permitido entender que está rama de la inteligencia artificial, a partir del análisis de datos con el fin de identificar patrones y apoyar en la toma decisiones con la mínima intervención humana (como reconocer si un mensaje de correo electrónico es spam o no) , esta disciplina permite en muchos ámbitos mejorar la productividad gracias al programa Python.

iii. La metodología de Deep Learning nos brinda una herramienta de mejora continua. La cual puede ser aprovechada para simplificar el análisis y la proyección de datos.

iv. La herramienta Python nos brinda una amplia gama de posibilidades. A través de una mejora continua, Python puede ser aprovechado a niveles exuberantes. La simplicidad para el manejo de Big data hace que Python sea beneficioso para aumentar la productividad tanto a nivel de empleado como a nivel de empresa.