Instituto Tecnológico y de Estudios Superiores de Monterrey

Reporte final

Inteligencia artificial avanzada para la ciencia de datos I (Gpo 101)

Equipo 1

Profesor:Dr. Jorge Adolfo Ramírez Uresti

Realizado por :

Joshua Daniel Hernández Coronado A01750800

13 de Septiembre de 2022

1. Introducción

El siguiente reporte constituye los resultados referenetes al análisis exploratorio de una base de datos extraida de la plataforma Kaggle, dicha base de datos posee información acerca del comportamiento de clientes de una institución bancaria los cuales son clasificiados por un target denominado .Attrition Flag"que determina la posibilidad que tine un cliente de abandonar el banco.

El dataset que mencionamos posee características específicas que permite una amplia manipulación entre ellas, se puede apreciar que no existen datos nulos, vacios ni duplicados. Por otro lado existen pocas variables categÓricas o cualitativas, lo que facilita el entendimiento y simplifica el trabajo de preprocesamiento, así mismo el dataset cuenta con una basta cantidad de datos lo que nos permite realizar un entrenamiento eficiente para clasificar.

El principal problema al que nos enfretamos en este caso es la clasificación de clientes que abandonaran la institución bancaria ya que el cálculo de .Attrition Flag.es determinado por un miembro del banco lo que hace que este proceso sea lento y tedioso para las instuciones financieras que buscan aumentar su nÚmero de cliente, pero que al mismo tiempo buscan estar al pendiente de sus clientes actuales, dicho de otro modo, la información obtenida sirve para generar perfiles e identificar patrones de comportamiento de clientes potenciales.

2. Preprocesamiento

Para el preprocesamiento comenzamos analizando la tipología de los datos contenidos en el dataset, para este análisis se utilizó el comando .info() que nos entrega una tabla con el tipo de valor de cada variable [1].

```
df.info()#analizamos la tipologia de los datos

<class 'pandas.core.frame.DataFrame'>
Int64Index: 10127 entries, 0 to 10126
Data columns (total 21 columns):
 #   Column                    Non-Null Count  Dtype
---  ------                    --------------  -----
 0   CLIENTNUM                 10127 non-null  int64
 1   Attrition_Flag            10127 non-null  object
 2   Customer_Age              10127 non-null  int64
 3   Gender                    10127 non-null  object
 4   Dependent_count           10127 non-null  int64
 5   Education_Level           10127 non-null  object
 6   Marital_Status            10127 non-null  object
 7   Income_Category           10127 non-null  object
 8   Card_Category             10127 non-null  object
 9   Months_on_book            10127 non-null  int64
 10  Total_Relationship_Count  10127 non-null  int64
 11  Months_Inactive_12_mon    10127 non-null  int64
 12  Contacts_Count_12_mon     10127 non-null  int64
 13  Credit_Limit              10127 non-null  float64
 14  Total_Revolving_Bal       10127 non-null  int64
 15  Avg_Open_To_Buy           10127 non-null  float64
 16  Total_Amt_Chng_Q4_Q1      10127 non-null  float64
 17  Total_Trans_Amt           10127 non-null  int64
 18  Total_Trans_Ct            10127 non-null  int64
 19  Total_Ct_Chng_Q4_Q1       10127 non-null  float64
 20  Avg_Utilization_Ratio     10127 non-null  float64
dtypes: float64(5), int64(10), object(6)
```

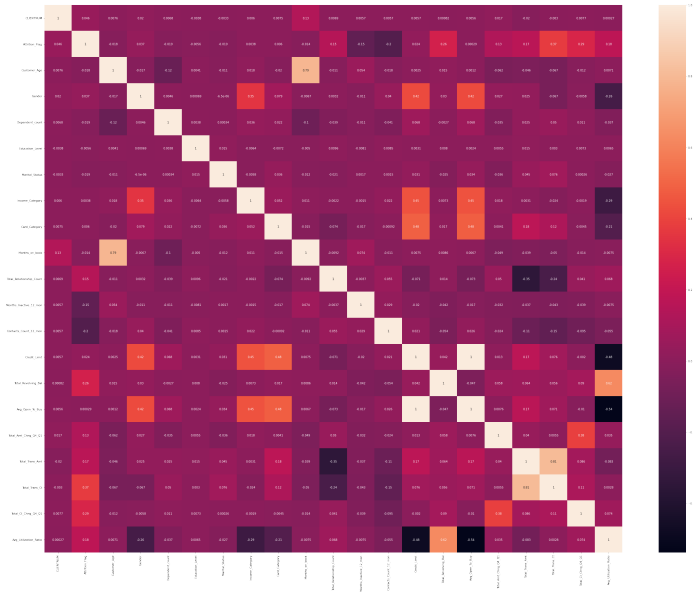
Este análisis se hace con la intención de determinar las variables que deben ser recategorizadas mediante el uso de *LabelEncoder* que nos permite pasar los valores categóricos a cualitatativos otorgándoles un número de indentificación, en este caso especifico se detectaron y se categorizaron las siguientes variables:

- Attrition_Flag
- Education_Level
- Marital_Status
- Income_Category
- Card_Category

En específico para la variable Income_Category se realizó una doble categorización debido a que esta variable se encontraba descrita mediante un rango de valores.

Se hizo un análisis de la base de datos y no se encontraron valores vacio ni nulos, aunque, si fuese necesario, se podría optar por alguna técnica de rellenado de datos como es utilizar la media, moda o mediana depediendo de la naturaleza de los valores de cada variable.

También se realizó un análisis de histogramas con la intención de conocer la distribucion de los datos de cada variable del dataset. De igual forma, para conocer la importancia de cada variable en el modelo, se generó un mapa de calor con el cual se puede conocer la correlacion entre las variables.



Finalmente se separo el dataset original en dos sets de datos con la intencion de tener un set de entremamiento y otro de test para probar los resultados de nuestro modelo. para ello se utilizo la libreria sklearn con el cual se puede hacver dicha separacion de manera sencilla mediante la funcion train_test_split() que separo el dataset en 70 % para train y 30 % para test ademas de que se coloco un random_state = 45 para asegurarnos que existiera una correcta variabilidad de muestras, sumado a esto se escalo la informacion para ayudar a que tanto nuestro modelo como otros modelos fuesen mas sensibles y eficientes a la hora de leer los elementos de las variables.

3. Resultados

Para los resultados he optado por utilizar la sublibreria de metrics de la libreria sklearn la cual posee diversos mecanismos de evaluacion de modelos que en este caso vendra bien para evaluar el funcionamiento del arbol de decisiones2.

Para esto se generaron 3 pruebas con diferentes hyperparametros los cuales varian en random_state, Max_depth,max_leaf_nodes, auqnue tambien se experimento con otros arboles sin embargo se mantuvieron los que possen mejores scores de evaluacion.

Los modelos probados fueron los siguientes:

- clf_1 = DecisionTreeClassifier(criterion = .$^{\text{en}}$tropy", random_state = 40, splitter = "best")
- clf_2 = DecisionTreeClassifier(criterion = .$^{\text{en}}$tropy", random_state = 56, splitter = "best",max_depth= 20)
- clf_3 = DecisionTreeClassifier(criterion = .$^{\text{en}}$tropy", random_state = 40, splitter = "best",max_depth= 10,max_leaf_nodes = 50)

Para la metrica de precision score se obtuvieron los siguientes resultados:

- 0.9603766182816791

- 0.962992125984252
- 0.9614636045153756

Para la metrica de accuracy score se obtuvieron los siguientes resultados:

- 0.9364922671931556
- 0.9381375452451465
- 0.9443896018427114

Para la metrica de ROC se obtuvieron los siguientes resultados:

- 0.8806873589699084
- 0.887307686238619
- 0.8870220756473577

Para la metrica de matriz de confusion se obtuvieron los siguientes resultados respecto a valores estimados de forma correcta por el modelo:

- 398 y 2448 de 3039
- 405 y 2446 de 3039
- 400 y 2470 de 3039

El bias es la diferencia entre los valores que se lograron predecir mediante el modelo generedado y los valores reales en este caso se espera que nuestro bias sea bajo debido a que el algoritmo es flexible y se puede ajustar facilmente a la complejidad que puedan tener los datos, los resultados obtenidos pueden compararse mediante una resta entre los procentajes obtenidos de los scores

En lo que respecta a la varianza se esperaria un error relativamente alto ya que no es posible contar con con bias bajo y varianza baja a este proceso de estimacion de los dos errores se le conoce como compensasion bias-varianza ya que se debe encontrar un equilibrio entre una baja varianza y un bajo bias,ademas de que lo esperado en un modelo de arboles de decisiones es una predisposicion a cambios en la estimacion de la funcion objetivo cuando varian los datos de train esto es observable mediante cross validation:

4. Conclusiones

Nuestro modelo es bueno para realizar predicciones ya que tiene un accuracy score de .9438 lo que nos indica un buen funcionamiento del algoritmo, asi mismo el arbol de decision es un buen algortimo para presentar los resultados del modelo ya que permite que se vean las reglas,

En este caso el rezago de los datos se mantuvo al minimo debido a que gran parte del dataset estaba completo y no requeria algun tipo de procesamiento especifico.

Con base en los resultados obtenidos por el modelo, los cuales se muestran en la seccion de resultados se afirma que su rendimiento es bueno. Finalmente, se concluye que la combinación de sus hypeerparametros fue lo que permitio desarollar un podemo tan preciso y exacto en lo que respecta a clasificiar correctamente la posibilidad de abandono.

Referencias

[1] J. D. Hunter, "Matplotlib: A 2d graphics environment," *Computing in science & engineering*, vol. 9, no. 3, pp. 90–95, 2007.

[2] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg *et al.*, "Scikit-learn: Machine learning in python," *Journal of machine learning research*, vol. 12, no. Oct, pp. 2825–2830, 2011.