TRABAJO PRÁCTICO FINAL – Ciencia de Datos

Franco Lores y Martin Etchemendigaray

INTRODUCCION

En el siguiente trabajo se realizará el análisis de un set de datos de una empresa de telecomunicaciones, con el fin de encontrar un modelo matemático que pueda predecir si un cliente abandonará la compañía o no en el futuro.

Primero se realizará un preprocesamiento y un Análisis Exploratorio de Datos (EDA) donde podremos sacar unas primeras conclusiones, para luego generar un pipeline de Machine Learning y encontrar el mejor modelo clasificador.

DESARROLLO

El set de datos a analizar corresponde a una empresa de telecomunicaciones, y la variable a predecir será "Churn", una variable categórica binaria, lo que quiere decir que solo podrá tomar 2 valores, por lo que nos encontramos ante un problema de **clasificación**. Churn representa si un cliente abandonará la empresa o no.

Como la variable a predecir/etiqueta es información del dataset (cada X viene acompañada por una etiqueta Y), los modelos de aprendizaje a utilizar serán supervisados.

Previo a la implementación de los modelos de Machine Learning, se realizará un Análisis Exploratorio de Datos, en donde podremos ver cómo están compuestos principalmente nuestros datos y las relaciones entre las variables.

PREPROCESAMIENTO

Para el análisis, previamente se debió realizar una limpieza y ordenamiento del dataset. Originalmente, cuenta con 7043 filas y 22 columnas.

En un primer paso se eliminaron las variables Column ID, Gender y Unamed: 0 ya que no aportan información sensible a la hora de generar el modelo de predicción.

Una vez realizado esto, nuestro dataset quedo con la siguiente estructura de informacion:

Vemos que de las 18 variables restantes, la mayoria son variables categóricas, mientras que unas pocas son continuas.

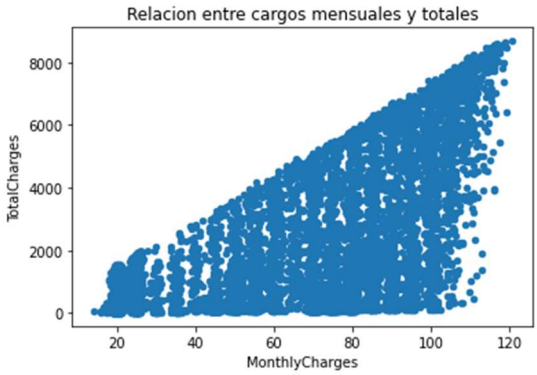
```
SeniorCitizen                                    [nan, 0.0, 1.0]
Partner                                           [Yes, No, nan]
Dependents                                        [No, nan, Yes]
tenure            [1.0, 34.0, 2.0, 43.516548463356976, 8.0, 22.0...
PhoneService                                      [No, Yes, nan]
MultipleLines                  [No phone service, No, nan, Yes]
InternetService                      [DSL, nan, Fiber optic, No]
OnlineSecurity            [No, Yes, nan, No internet service]
OnlineBackup              [Yes, No, nan, No internet service]
DeviceProtection               [No, Yes, No internet service]
TechSupport                    [No, Yes, No internet service]
StreamingTV                    [No, Yes, No internet service]
StreamingMovies                [No, Yes, No internet service]
Contract          [Month-to-month, One year, nan, Two year]
PaperlessBilling                                  [nan, No, Yes]
PaymentMethod     [Electronic check, Mailed check, Bank transfer...
MonthlyCharges    [29.85, 55.5735294117647, 53.85, 42.3, 75.825,...
TotalCharges      [29.85, 1889.5, 108.15, 1840.75, 151.65, 820.5...
Churn                                             [No, Yes]
```

Se encontraron valores nulos en gran parte de las features. Eliminarlos implicaría reducir en casi un 90% la cantidad de samples.

Es por eso que se tomaron una serie de decisiones para re-ordenar el dataset:

1. Cambiar, dentro de la variable Total Charges, los valores vacíos por nulos para evitar una inconsistencia en la información y tipo de dato que maneja cada Atributo.

2. Realizar un rellenado de los nulos de las variables Tenure, MonthlyCharges y TotalCharges Se detectó una relación entre ellas de la forma= TotalCharges = MonthlyCharges * Tenure. Es por esto que, en caso de tener 2 de 3, se puede aproximar el valor de la restante.



3. Con el resto de los nulos de las demás variables, se las llenó realizando un autollenado tomando el valor inmediato anterior.

LABEL ENCODER

Se realizó un label encoder para cambiar valores string dentro de los atributos por valores numéricos de tipo 0 y 1 para aquellos atributos binarios.

Junto a esto, se cambió el tipo de variable que maneja cada columna del dataset para lograr la consistencia de los datos con la estructura. La estructura final quedó dispuesta de la siguiente manera.

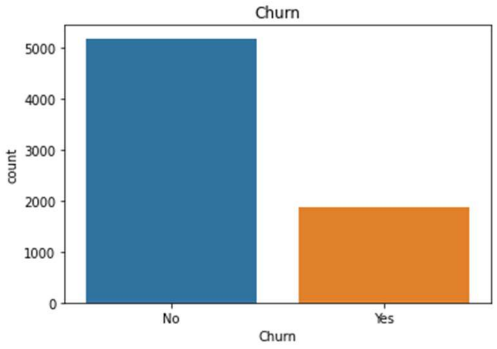
```
 #   Column            Non-Null Count  Dtype
---  ------            --------------  -----
 0   SeniorCitizen     6873 non-null   float64
 1   Partner           6873 non-null   float64
 2   Dependents        6873 non-null   float64
 3   tenure            6873 non-null   float64
 4   PhoneService      6873 non-null   float64
 5   MultipleLines     6873 non-null   float64
 6   InternetService   6873 non-null   object
 7   OnlineSecurity    6873 non-null   float64
 8   OnlineBackup      6873 non-null   float64
 9   DeviceProtection  6873 non-null   float64
 10  TechSupport       6873 non-null   float64
 11  StreamingTV       6873 non-null   float64
 12  StreamingMovies   6873 non-null   float64
 13  Contract          6873 non-null   object
 14  PaperlessBilling  6873 non-null   float64
 15  PaymentMethod     6873 non-null   object
 16  MonthlyCharges    6873 non-null   float64
 17  TotalCharges      6873 non-null   float64
 18  Churn             6873 non-null   float64
```

Además, para las variables categóricas no binarias, se generaron variables dummies.

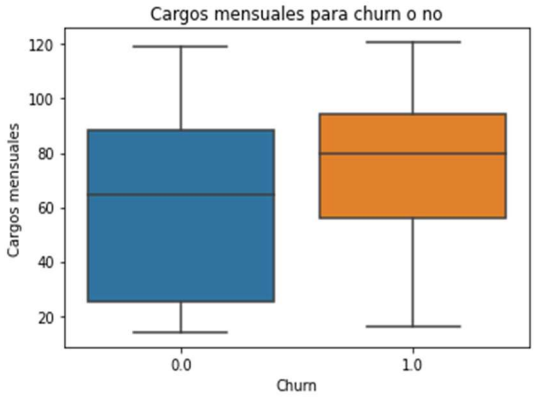
ANÁLISIS EXPLORATORIO DE DATOS

Dentro del EDA se encontraron relaciones que permiten encontrar algunas conclusiones alrededor de nuestra variable a predecir, Churn.

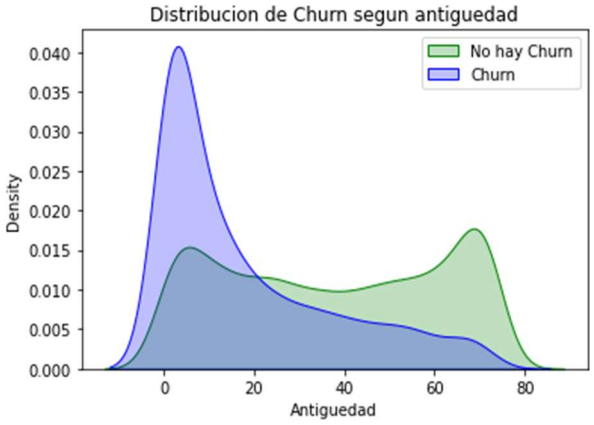
Lo primero que podemos visualizar es la proporción actual de Churn que encontramos en nuestro set de datos



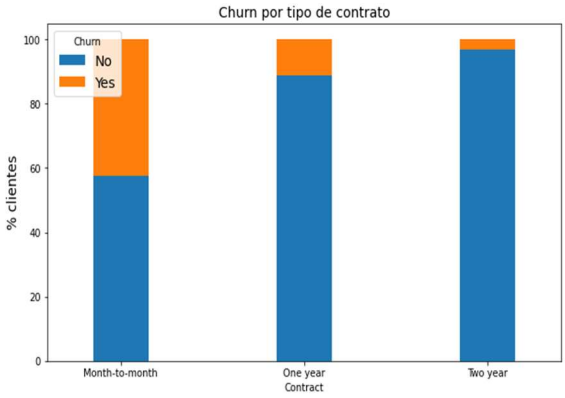
En el siguiente boxplot vemos como la tendencia a irse es de aquellos usuarios con mayor cargos mensuales, lo cual se puede interpretar como algo lógico, ante la posibilidad de un mejor precio en la competencia.



Abajo se puede visualizar cómo las personas que tienden a irse son aquellas con menor antigüedad, posiblemente debido a la baja fidelización con la compañía por parte de los clientes



En la siguiente imagen vemos cómo las personas con un contrato mensual poseen más usuarios que han dejado el servicio en comparación a las otras modalidades de contratación.



MODELO DE MACHINE LEARNING

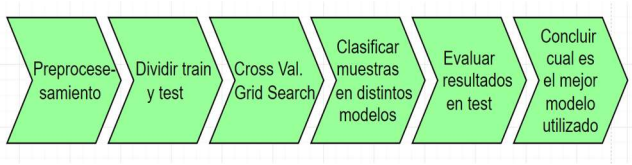
Como ya se mencionó previamente, nos encontramos ante un caso de aprendizaje

$$S = \{(x_1, y_1), (x_2, y_2), ..., (x_n, y_n)\}$$

$$\mathcal{X} \in \mathbb{R}^d \qquad y \in \{0, 1\} \qquad f(x) = y$$

supervisado de clasificación, donde la variable a predecir Churn es categórica binaria.

El pipeline para realizar la modelización será el siguiente:



Luego de dividir nuestro set de datos en Train y test (se utilizó un 30% de las muestras para test) y utilizar Standard Scaler para estandarizar los distintos rangos de valores en las variables de X, se generaron distintos modelos de clasificación.

KNN

El modelo de KNN toma la muestra de Xtest y encuentra el punto K más cercano en Xtrain y

$$\Pr(Y = j|X = x_0) = \frac{1}{K} \sum_{i \in \mathcal{N}_0} I(y_i = j). \qquad (2.12)$$

calcula la probabilidad condicional de la nueva muestra para clasificarla.

El accuracy para este modelo fue de 0.7759 y su matriz de confusión la siguiente:

Support Vector Machine

El SVM es un modelo de clasificación que busca el hiperplano separador que maximice el margen entre las clases.

$$\beta_0 + \beta_1 x_{i1} + \beta_2 x_{i2} + \cdots + \beta_p x_{ip}$$

Luego de realizar GS/CV encontramos que los mejores hiperparámetros para nuestro modelo son

- 'C'= 10,
- 'gamma' = 0.01,
- 'kernel' = 'rbf'

El accuracy del modelo fue de 0.8064 y la matriz de confusión la siguiente:

LOGISTIC REGRESSION

Este modelo se compone de una regresión lineal precedida de una función de activación sigmoide, lo que genera que el output sea binario y no continuo como en una regresión normal.

$$p(X) = \beta_0 + \beta_1 X.$$

$$p(X) = \frac{e^{\beta_0+\beta_1 X}}{1 + e^{\beta_0+\beta_1 X}}.$$

A cada muestra clasificada, le asigna una probabilidad de pertenecer a cada clase de nuestra label (Churn) Si la probabilidad es mayor a cierto umbral (0.5) entonces pertenece a una clase y viceversa.

Para este modelo el accuracy fue de 0.8094 y la matriz de confusión la siguiente:

CONCLUSIONES

A continuación, se pueden visualizar los resultados de evaluación para los distintos modelos implementados:

Modelo	Accuracy	Sensitivity	Specificity
KNN	0,7759	0,63	0,8939
SVM	0,8064	0,55076923	0,9409
Logistic Reg.	0,8094	0,61946903	0,9240

Si bien podemos ver que el modelo con mayor accuracy es el de Logistic regression, esto no necesariamente quiere decir que sea el mejor modelo para predecir ante nuevos datos. Por ejemplo, podemos ver gracias a la matriz de confusión que la specificity es superior en el modelo SVM.

Esto quiere decir que el modelo de Logistic Regression tiene un mayor número de falsos positivos, algo que creemos que es más importante que una pequeña diferencia en la accuracy.

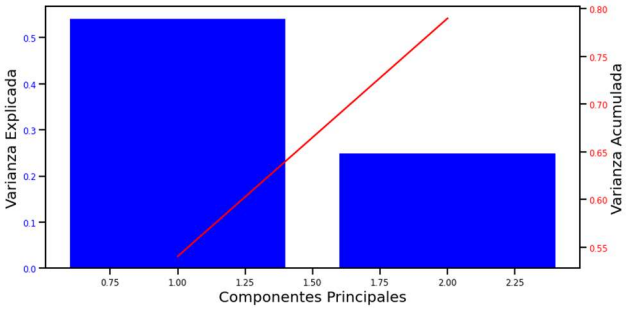
En un modelo de negocios, pensar que un cliente no va a abandonar la compañía y que finalmente si lo haga, es mucho más importante que pensar que se va a ir y que finalmente no lo haga.

Es por eso que concluimos que, para este caso en particular, es mejor ceder un poco de

accuracy y mejorar la specificity con el modelo SVM.

Como nuestro dataset se compone principalmente por variables categóricas (solo 3 son variables numéricas), realizar un PCA no tiene demasiada relevancia ya que como máximo podremos reducir la cantidad de variables en +/- 10% (de 25 a 23).

Si bien realizo el mismo, generando 2 variables que representaban el 80% de la covarianza de las 3 variables originales, los resultados del modelo no cambiaron.



REFERENCIAS

-Libro: An introduction to Statistical Learning. Gareth James, Daniela Witten, Trevor Hastie, Robert Tibshirani

-Libro: Pattern Recognition and Machine Learning. Christopher M. Bishop.

-Libro: Python Data Science Handbook.Jake VanderPlas

-Apuntes de clase del Ing. Martin Palazzo.