BANK SUBSCRIPTION

Universidad Tecnológica Nacional

Autores:
Bottero Guido y Troitiño Imanol

Resumen:

Este trabajo se basa en un conjunto de datos que contiene las características de aquellos clientes bancarios que se suscribieron y de aquellos que no lo hicieron a una campaña de marketing. A partir de esta información, se emplearon técnicas de machine learning para predecir la probabilidad de que los clientes se suscriban a la campaña, en función de sus características. Se evaluaron dos modelos de clasificación: Regresión Logística y Support Vector Machine, probando diversos hiperparámetros. Además, se comparó el desempeño de ambos modelos al aplicar, previamente a su entrenamiento, una técnica de reducción de dimensionalidad. Los resultados mostraron que el modelo de Regresión Logística sin reducción de dimensionalidad fue el más adecuado para este problema, logrando una precisión (accuracy) de 0.9 y un área bajo la curva ROC (AUC) de 0.87.

Palabras clave: Machine learning, Data Science, classifier.

I. INTRODUCCIÓN Y OBJETIVOS

Dado el significativo esfuerzo económico que los bancos destinan a campañas de marketing y captación de clientes, surge la necesidad de optimizar este proceso para reducir los costos asociados. El presente estudio tiene como objetivo predecir, en función de las características individuales de cada cliente, la probabilidad de que se suscriban o no a una campaña de marketing específica del banco. El propósito es orientar las acciones de marketing hacia aquellos clientes con mayor probabilidad de suscripción, según los resultados de la predicción, con el fin de maximizar la eficiencia de las campañas.

Para lograr esta predicción, se emplearán técnicas de Machine Learning con el fin de desarrollar un modelo matemático adecuado. El modelo será entrenado utilizando la base de datos de clientes proporcionada por el banco, que incluye información sobre una cartera de 45.211 clientes y 17 características asociadas a cada uno.

II. DESCRIPCIÓN DEL DATASET

La base de datos de los clientes, a partir de ahora "Dataset", cuenta con las siguientes variables (o features):

Variables numéricas numéricas:
- *Age:* Edad del cliente
- *Balance:* Promedio anual de saldo en la cuenta.
- *Contact:* Tipo con contacto del cliente.
- *Last Contact Day:* Último día de contacto en el mes.
- *Last Contact Duration:* Duración del último contacto en segundos.
- *Campaign:* Cantidad de contactos durante esta campaña.
- *Pdays:* Cantidad de dias que pasaron del último contacto de una campaña anterior.
- *Previous:* Cantidad de contactos previos a esta compaña.
- *Poutcome:* Performance de la campaña de marketing anterior.

Categóricas:
- *Job:* Tipo de empleo del cliente
- *Martial* status: Estado civil
- *Education:* Educacion maxima alcanzada
- *Last Contact Month:* Ultimo mes de contacto en el año

Booleanas:
- *Credit:* Si tiene deuda de crédito o no.
- *Housing loan:* Si tiene seguro de hogar o no.
- *Personal loan:* Si tiene prestamos o no.

En el Dataset, hay features que tienen desde un 11% hasta un 17% de valores nulos. Además, la cantidad de muestras con al menos un valor nulo representan el 77% del Dataset.

III. ANALISIS EXPLORATORIO DE DATOS

Previo a la implementación de cualquier método de aprendizaje supervisado, es necesario realizar un pre-procesamiento sobre el Dataset y un Análisis Exploratorio de Datos (EDA) para poder limpiar el Dataset y comprender la naturaleza de cada una de sus features junto con su correlación.

A. Manejo de features categóricas y numéricas:

No se identificaron clases dentro de las features categóricas que tengan distinto nombre e igual significado.

Por otro lado, dentro de las numéricas se reemplazaron los valores -1 de Pdays por el máximo valor de la propia feature ya que es lo más semejante a que un cliente nunca haya sido contactado por una campaña de marketing.

B. *Manejo de valores nulos:*

Se reemplazaron los valores nulos de cada feature siguiendo los siguientes criterios, en función de si la feature era numérica o categórica:

- *Categóricas:* Se comparó cada feature categórica contra otra feature categórica o numérica discretizada (con deciles) para poder reemplazar los nulos con la moda resultante de la comparación.

- *Numéricas:* En función de la distribución de los datos de cada feature se decidió si es conveniente reemplazar los nulos con la media, mediana o con algún otro valor en específico de la propia variable numérica.

C. *Manejo de Outliers:*

No se encontraron valores atípicos dentro del Dataset.

D. *Matriz de correlación lineal:*

En la matriz se puede observar cuan grande es la correlación lineal entre dos variables numéricas. En el Dataset, la mayor correlación lineal se encuentra entre las features Pdays y Previous con un valor de -0.52.

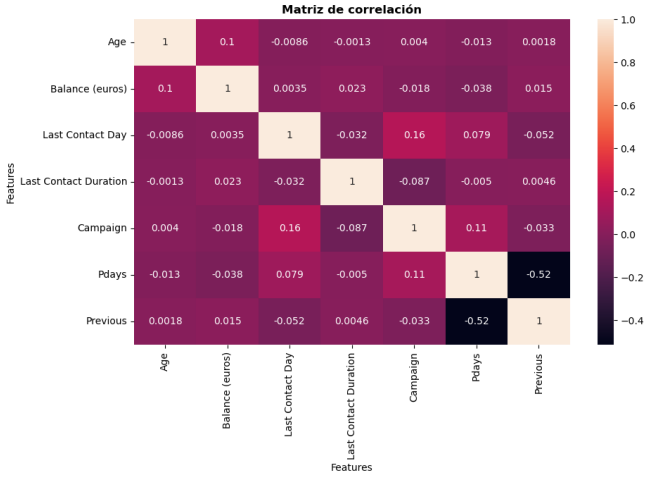

Figura 1: Matriz de correlación lineal

E. *Distribución de la variable a predecir*

En el Dataset, la variable Subscription, que se buscará predecir, tiene un 88.3% registros iguales a 0 y el otro 11.7% igual a 1.

IV. MATERIALES Y MÉTODOS

A. *Materiales*

Para llevar a cabo el entrenamiento de los modelos de Machine Learning se utilizó el programa "Jupyter Lab" dentro de "Anaconda Navigator" junto con las siguientes librerías: Pandas, Numpy, Matplotlib, Seaborn y Sklearn.

B. *Métodos de clasificación utilizados*

Los métodos de clasificación que se evaluaron para predecir la variable "Subscription" fueron los siguientes:

- **Regresión Logística:** Es un clasificador de regresión lineal precedido de una función de activación "sigmoid", lo que genera que el output sea binario y no continuo como una regresión normal.
  A cada muestra clasificada, le asigna una probabilidad de pertenecer a cada clase existente en el problema. Si la probabilidad es mayor a cierto umbral (0.5) entonces pertenece a una clase y viceversa.
  Posee un hiperparámetro C (costo) que penaliza la complejidad del modelo durante la optimización de la función de pérdida "Cross entropy". Esto evita el sobreajuste.

- **Support Vector Machine (SVM):** Es un clasificador lineal que busca un hiperplano separador que maximice el margen entre clases.
  El hiperplano separador queda definido por un subconjunto de muestras llamadas "support vectors".
  Posee 3 tipos de hiperparámetros. El hiperparámetro C (costo) penaliza a las muestras mal clasificadas durante la optimización de la función de pérdida.
  Por otro lado, existe un hiperparámetro asociado al kernel utilizado sobre el conjunto de datos. Cuando el conjunto de datos no es linealmente separable, se puede utilizar un kernel para encontrar un espacio dimensional en el que se puedan clasificar mejor los datos utilizando un clasificador lineal. Algunos de los kernels más utilizados son el lineal, gaussiano y polinómico.
  Finalmente, otro hiperparámetro se define en función del kernel seleccionado. Por ejemplo "d" es el grado que se debe definir en un kernel polinómico.

C. *Método de reducción de la dimensionalidad utilizado*

La reducción de dimensionalidad se basa en la hipótesis de que los datos de alta dimensionalidad contienen información redundante y que su estructura intrínseca puede representarse en un espacio de menor dimensión sin pérdida significativa de información.

Al reducir la dimensionalidad de un Dataset, se reduce la complejidad computacional a la hora de entrenar algoritmos de Machine Learing y se mitiga el problema de la "maldición de la dimensionalidad" donde el aumento de dimensiones dificulta identificar patrones y genera sobreajuste.

Una técnica utilizada para efectuar la reducción de la dimensionalidad es PCA (Principal Component Analysis).

PCA tiene un hiperparámetro "p" que define las dimensiones a las que se quiere reducir el Dataset original. El algoritmo de PCA sigue los siguientes pasos:

1. Se calcula la matriz de covarianza σ sobre las features del Dataset original.

2. Se realiza la descomposición espectral de σ para obtener sus autovectores y autovalores asociados. Los autovectores con mayor autovalor son los que representan la mayor variabilidad del Dataset.

3. Se ordenan los autovalores de mayor a menor.

4. Se construye una nueva matriz Z utilizando los primeros "p" autovalores y autovectores asociados. A estos últimos se los llama "componentes principales"

.

El resultado de PCA es la matriz Z, dada por un conjunto de datos de dimensión "p" menor a la dimensión "d" del Dataset original.

D. Grid Search y Cross Validation

Como se detalló anteriormente, a la hora de entrenar un modelo existen múltiples hiperparámetros que se pueden utilizar. Para encontrar la mejor combinación de hiperparámetros se utilizaron las técnicas de "Grid Search" y "Cross Validation".

- **Grid Search:** Es una grilla donde se mapean las distintas combinaciones de hiperparámetros que se desean probar durante el entrenamiento del modelo.

- **Cross Validation:** Se separa el conjunto de datos train del Dataset en múltiples subconjuntos train y validation. La media, de la métrica a evaluar, entre todos los subconjuntos define el puntaje final del modelo entrenado.

E. Pipeline

En el presente informe se desarrollaron dos pipelines con el propósito de encontrar el mejor modelo para predecir el valor de la variable "Subscription".

Pipeline 1: Modelo sin PCA

Realiza una primera limpieza sobre el Dataset original con las modificaciones descriptas en "Manejo de valores nulos" y "Manejo de features categóricas y numéricas". Luego, realiza un "StandarScaler "sobre las variables numéricas y un "OneHotEncoder" sobre las variables categóricas.

- **StandarScaler:** Transforma la feature numérica de tal forma que tenga media 0 y desvío estándar 1. Esto hace que la escala de todas las features sean semejantes y suaviza el efecto de los outliers.

- **OneHotEncoder:** Transforma las features categóricas en dummies. Esto es, crear una feature binaria adicional por cada clase que contenga la feature categórica original. De esta forma, los modelos a entrenar pueden "leer" las features categóricas.

Finalmente, el pipeline 1 realiza un Grid Search y Cross Validation para encontrar el mejor modelo de entre los siguientes:

- **Regresión logística: Se** prueba con los hiperparámetros C igual 100, 10 y 1.

- **Support Vector Machine:** Se prueba con el Kernel gaussiano (rbf), y con los hiperparámetros, C igual 5, 50 y 500 y gamma igual a 0.1, 0.01, 0.001.

Pipeline 2: Modelo con PCA

El flujo del pipeline 2 es idéntico al 1, solo que luego de aplicar el StandarScaler y el OneHotEncoder se aplica PCA para reducir la dimensionalidad del Dataset con un valor de "p" igual a 10.

Finalmente, el pipeline 2 realiza un Grid Search y Cross Validation para encontrar el mejor modelo de entre los siguientes:

- **Regresión logística: Se** prueba con los hiperparámetros C igual 0.1, 1 y 10.

- **Support Vector Machine:** Se prueba con el Kernel lineal (linear), y con los hiperparámetros, C igual 1, 5 y 10 y gamma igual a 0.1 y 0.01.

V. EXPERIMENTOS Y RESULTADOS

Se procede a realizar el análisis de los resultados del modelo sin PCA (pipeline 1) y luego con PCA (pipeline 2). En la primera instancia se obtuvieron los siguientes resultados:

Modelo	Sin PCA
Accuracy train	0.9017
Accuracy test	0.8999
AUC	0.8762
Sensitivity	0.3230
Specificity	0.9763

Tanto el accuracy como el AUC son cercanos a 1, con lo cual en principio parece ser que el modelo hace una buena predicción de la variable "Subscription".

El accuracy en train y en test es muy similar y con valores cercanos a 1, por lo que se podría decir que no hay grandes errores por Bias ni Variance.

El specificity tiene un valor muy cercano a 1, lo que quiere decir que el modelo clasifica muy bien cuando un usuario no se subscribe. Contrariamente, el sensitivity tiene un valor muy bajo, y esto quiere decir, que el modelo no está logrando detectar de forma acertada a los usuarios que se subscriben. Parece ser a priori, por la naturaleza del Dataset utilizado, ya que la gran mayoría (88%) de clientes del Dataset tienen "Subscription" igual 0, y esto dificulta el entrenamiento del modelo a la hora de detectar la clase "Subscription" igual 1.

En la segunda instancia, aplicando PCA, se prueba modelar con distinta cantidad de componentes principales, donde 10 es la

cantidad que mantiene un buen accuracy, similar al modelo sin PCA, pero utilizando solo el 20% de la cantidad de columnas que usó el modelo sin PCA (tiene 50 sin contar la variable target).

Modelo	Sin PCA	Con PCA [n=10]
Accuracy train	0.9017	0.8950
Accuracy test	0.8999	0.8897
AUC	0.8762	0.8394
Sensitivity	0.3230	0.1841
Specificity	0.9763	0.9830

A pesar de que la diferencia entre el accuracy en train y test del modelo con PCA es mayor a la del modelo sin PCA, esta sigue siendo poco significativa. Por ende, se podría decir que este modelo tampoco tiene grandes errores por Bias y Variance.

El área AUC bajo la curva ROC del modelo con PCA es siginifcativamente menor a la de la del modelo sin PCA. Esto quiere decir, que el modelo con PCA clasifica peor a medida que la función del clasificador encontrado se mueve en la dirección de alguna de las clases de la variable “Subscription”.

En línea con el punto anterior, se puede observar que si bien el modelo con PCA tiene un specificity levemente superior que el modelo sin PCA, el sensitivity es considerablemente peor. Puntualmente, es practicamente un 44% menor al modelo sin PCA, y esto quiere decir, que el modelo con PCA es peor a la hora de predicer si un cliente se subscribirá.

Por ultimo, se grafican las curvas ROC para ambos modelos en donde se puede observar graficamente las diferencias en la performance.

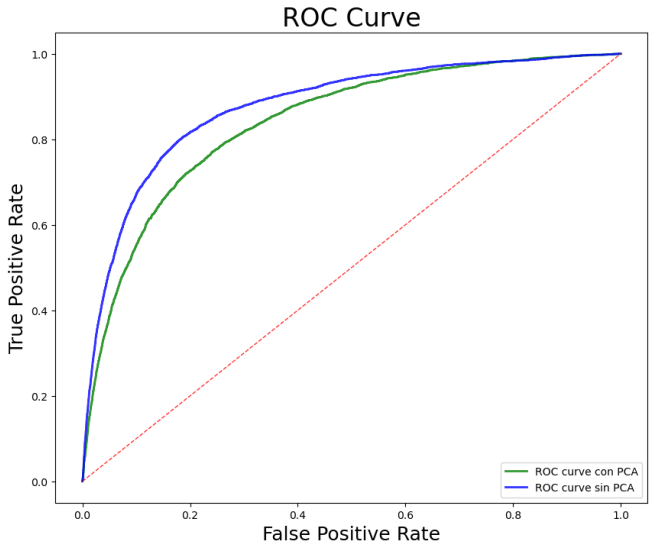

Figura 2: Curvas ROC

VI. Discusión y conclusiones

Luego de haber entrenado y analizado las métricas de los modelos de clasificación con y sin reducción de la dimensionalidad (PCA), se concluye en que el mejor modelo que se ha encontrado para realizar la predicción de la variable categórica "Subscription" es el de Regresión Logística con hiperparámetro C igual a 10 y sin aplicar PCA. Dicha conclusión surge por los siguientes motivos:

1. El accuracy del modelo sin PCA alcanzó un valor en test superior al modelo con PCA (0.899 vs 0.889).

2. El modelo con PCA alcanzó un accuracy similar al modelo sin PCA, utilizando comparativamente solo el 20% de la cantidad de features, lo que lo vuelve un modelo más "liviano" para llevar a producción.

   A pesar de ello, también tuvo un Sensitivity aproximadamente un 44% menor al del modelo sin PCA (0.184 vs 0.323), lo que lo vuelve considerablemente peor a la hora de identificar clientes que se subscriben. Es por ello, que concluimos en que la ventaja del modelo con PCA dada por ser un modelo más liviano, no compensa el hecho de que genera peores predicciones que el modelo sin PCA.

Bibliografía

[1] An Introduction to Statistical Learning. Gareth James • Daniela Witten • Trevor Hastie • Robert Tibshirani

[2] Probabilistic Machine Learning. *Kevin P. Murphy*

[3] Python Data Science Handbook. *Jake VanderPlas*

[4] Documentacion librería scikit-learn. https://scikit-learn.org/stable/supervised_learning.html