.UBAfiuba
FACULTAD DE INGENIERÍA

75.06 /95.58 Organización de Datos
Dr. Ing. Rodríguez - 2°C 2023

Informe Final - Grupo 29

Integrantes	Padrón
Isidro Héctor Borthaburu	108901
María Delfina Cano Ros Langrehr	109338
Martín Wainwright	108211

Introducción

A través de los distintos checkpoints fuimos explorando diferentes técnicas para mejorar la predicción y performance de nuestro modelo. En el primer checkpoint realizamos un análisis exploratorio de los datos donde fuimos observando las variables, sus categorías, si habían outliers, correlaciones, gráficos de las distribuciones, relación con el target entre otros. En este trabajo, eliminamos las variables que considerábamos irrelevantes para el análisis, imputamos las observaciones que tenían valores faltantes y realizamos distintas técnicas para el análisis y tratamiento de outliers.

Luego, en el segundo trabajo, realizamos las modificaciones necesarias al dataset para poder alimentar nuestro modelo de árbol de decisión. También realizamos la optimización de hiperparámetros, utilizando *Cross Validation*, y evaluamos el modelo en los conjuntos de entrenamiento y validación usando métricas como F1-score, precisión y recall mostrando la matriz de confusión.

Para el tercer checkpoint, como queríamos trabajar únicamente con números, cambiamos los meses de llegada por los valores 0 y 1 sobre si el hotel estaba en temporada alta o no, y también dejamos únicamente los nombres de los 10 países más frecuentes de proveniencia de los huéspedes y el resto los nombramos como "otros". Las técnicas que utilizamos en este checkpoint fueron: *KNN, SVM: Normalizado y Kernel-Radial, Random Forest, Cross Validation, XGBoost* y Ensambles Híbridos, como Stacking y Voting.

Por último, en la cuarta y última etapa del trabajo práctico implementamos el uso de redes neuronales, una de las mejores herramientas al momento de realizar predicciones. Para estos problemas usamos distintas funciones de activación, en regresión utilizamos Sigmoidea y Relu, mientras que en clasificación utilizamos sigmoidea 3 y softmax. También probamos diferentes casos cambiando la cantidad de neuronas tanto de entrada como neuronas ocultas. La búsqueda de hiperparametros no deja de ser importante para un mejor análisis, en este caso trabajamos con un Grid Search Cross Validation, para ver cual es la mejor combinación de hiperparametros para nuestra red neuronal.

Cuadro de Resultados

Realizamos un cuadro de los mejores modelos obtenidos en cada checkpoint. Como en el primero no subimos ninguna predicción a la competencia de Kaggle, decidimos incluir otro de los modelos que consideramos que era un buen predictor.

Modelo	CHPN	F1-Test	Presicion Test	Recall Test	Accuracy	Kaggle
Decision Tree Classifier	2	0.8379	0.8250	0.83484	0.83562	0.8259
XGBoost	3	0.85	0.85	0.85	0.8567	0.836
Random Forest	3	0.85	0.85	0.85	0.8505	0.837
modelo_7	4	0.83	0.83	0.83	0.8290	0.8136

En el segundo trabajo, nuestro mejor modelo fue el *Decision Tree Classifier* donde este es un árbol de decisión donde cada nodo es una pregunta y sus 2 subnodos son sus posibles respuestas. De esta forma, se van clasificando las observaciones según sus distintos atributos. También utiliza la entropía como hiper parámetro para medir la impureza.

En cambio, en el Checkpoint 3, nuestro mejor modelo fue el *Random Forest*. Este consiste en que varios árboles toman decisiones basándose en las características de los datos, y luego el modelo combina todas las decisiones y devuelve lo “más elegido” por todos los árboles, para así tomar una decisión con mayor precisión. Además, tuvimos otro muy buen modelo predictor que fue el *XGBoost* donde este consiste en construir distintos árboles de decisión donde cada nuevo árbol creado, corrige los errores de los árboles anteriores. De esta forma, se mejora constantemente las decisiones tomadas.

Para finalizar, en el último checkpoint implementamos una red neuronal donde este tenía un total de 7 capas donde estas fueron: 300, 250, 200, 150, 100, 50 , 25 y 1, y el total de épocas utilizados en este modelo fueron 30. También en este modelo utilizamos la métrica AUC ya que nos dimos cuenta que esta era mejor que Accuracy. Por último, utilizamos un batch size de tamaño 30.

Conclusiones generales

Si nos pareció útil realizar un análisis exploratorio de los datos ya que de esta forma pudimos observar y analizar el dataset, y las distintas formas que se comportaba. También pudimos observar qué variables eran cualitativas y

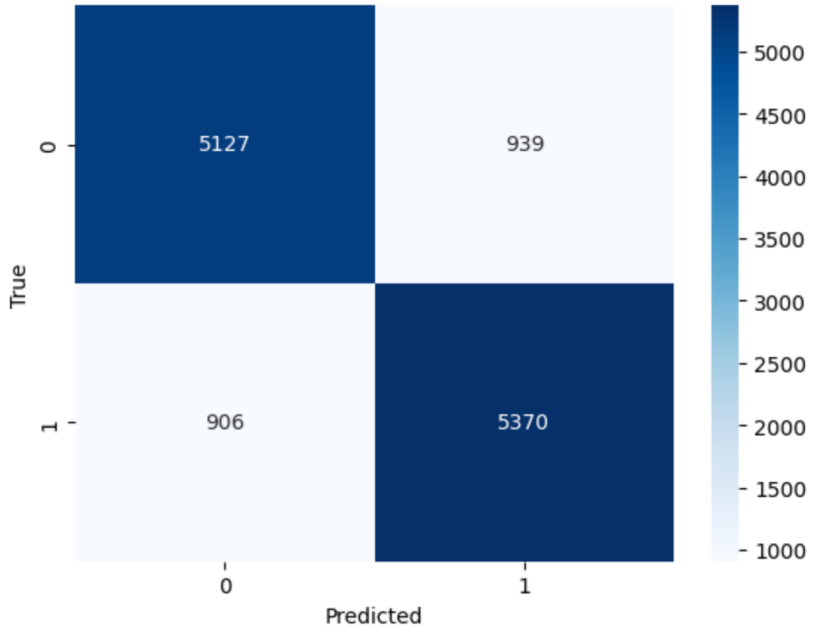
cuantitativas, como se correlacionaron distintas variables entre sí, sus respectivos outliers univariados y multivariados, la relación de las variables con el target, si habían variables irrelevantes para el análisis, entre otras.

Las tareas de preprocesamiento que realizamos creemos que ayudaron a mejorar la performance de los modelos ya que, como explicamos anteriormente, pudimos detectar outliers, variables irrelevantes que no servían para el análisis, etc. que si no los hubiéramos observado, no los hubiéramos tratado y, de esta forma, mejorar el rendimiento de nuestro modelo.

El modelo que tuvo el mejor desempeño en los datos de test fue *Random Forest* donde para observar este valor utilizamos la predicción de Kaggle. Sin embargo, el modelo más rápido de realizar fue el Decision Tree Classifier utilizando Cross Validation y, además, era el más sencillo de entrenar e implementar. Por ende, este modelo sí es útil para realizar predicciones y obtener un buen desempeño.

Nosotros creemos que es posible usar nuestro modelo de forma productiva por su gran eficacia y ser un buen clasificador de observaciones, tanto para clasificación como para regresión. Igualmente, nuestros resultados podrían ser mejores si el pre-procesamiento de datos fuera más exhaustivo y tal vez teniendo una máquina con mejor capacidad o utilizando nuestras computadoras como servidor y no trabajando en la nube con *Google Colab* ya que muchas veces se volvía muy tedioso ejecutar códigos y muy extenso el tiempo de ejecución.

Por último, como mencionamos anteriormente, nuestro mejor modelo predictor fue el *Random Forest* donde se obtuvo una muy buena clasificación de los datos. Sin embargo, este no fue el mejor predictor sobre los datos de entrenamiento, en cambio, el modelo que mejor se ajustó a estos datos fue el *XGBoost* donde obtuvimos una accuracy del 0.8567, un poco mejor que la de *Random Forest*. Esto se puede deber a que el *XGBoost* pueda estar un poco sobre ajustado a los datos de entrenamiento en comparación con el otro modelo y así obtener una menor predicción en los datos de test utilizando Kaggle. Finalmente, el siguiente gráfico es nuestra matriz de confusión del mejor modelo y se puede observar como este es un gran clasificador de datos,



Matriz de Confusión utilizando el modelo Random Forest

Tareas Realizadas

Integrante	Promedio Semanal (hs)
Isidro Borthaburu	9
Maria Delfina Cano Ros Langhrer	9
Martin Wainwright	9