Informe Final - Grupo 23

Introducción

Modelos entrenados:

Árboles de decisión: un gran modelo, lleva poco tiempo para entrenarse y produce muy buenos resultados. Además gracias a la poda y la profundidad se puede evitar el overfit fácilmente.

KNN: tarda bastante en entrenarse y no fue muy efectivo con el problema asignado. No hubo manera de evitar el overfitting con este modelo.

SVM: fue el modelo que más tardó en entrenarse por diferencia, pero de todas maneras dio un resultado bastante positivo.

RandomForest: otro modelo simple, fácil de entrenar y que dio excelentes resultados. Como cada árbol se entrena en una muestra aleatoria del dataset se reduce notablemente el overfitting.

XG Boost: el modelo que mejores resultados nos dio en este problema. Sin embargo, es bastante más lento para entrenarse, comparado a RF, ya que el creado de árboles es de manera secuencial en este caso.

Redes Neuronales: pensábamos que era el modelo que mejor iba a rendir pero no fue así. Es posible que el problema a resolver no favorezca este tipo de modelo. Cabe recalcar que son rápidos para entrenar, pero si se agregan muchas capas a la red se termina sobreentrenando rápidamente, a pesar de tener regularizaciones y otras herramientas.

Stacking y Voting: en este caso también pensamos que iba a darle un gran beneficio a nuestras predicciones pero no fue así. Quizás nuestro problema fue que utilizamos pocos modelos a la hora de crear estos ensambles híbridos.

Pruebas y técnicas realizadas:

Buscando la mejora de nuestros modelos utilizamos diversas técnicas. Desde normalización y estandarización de datos, imputación hot deck para algunos valores faltantes, eliminación de outliers tanto univariados como multivariados, one hot encoding para poder utilizar variables categóricas, entre tantas técnicas. Además creamos nuevas variables y discretizamos valores tales como la variable country para proveer mejor información a nuestros modelos. Un detalle interesante a comentar fue la imputación de datos faltantes de agent_id, donde en función de el país y el hotel definimos cuál era el agente más probable de cada reserva. Algo similar

realizamos con los valores faltantes de country, donde encontramos correlación con deposit_type y market_segment
Para obtener la mejor combinación de variables para aportar a cada modelo primero tomamos las variables más importantes que habíamos encontrado durante el análisis exploratorio de datos y, luego, con el resto de las variables fuimos analizando si al agregarlas aportan nueva información útil para el mismo.

Cuadro de Resultados

Modelo	CHPN	F1-Test	Precision Test	Recall Test	Metrica X	Kaggle
modelo_1	1	0.8674	0.8528	0.8825	0.8768	0.8566
modelo_2	2	0.8800	0.8651	0.8953	0.9551	0.8666
modelo_3	3	0.8805	0.8734	0.8878	0.9482	0.8692
modelo_4	4	0.8628	0.8580	0.8676	0.8709	0.8545

Modelo 1: consiste en un modelo de árbol de decisión realizado en el chapter 2.
Params: criterion"entropy", max_depth = 23, min_samples_leaf = 8, ccp_alpha=0.00015.

Modelo 2: consiste en un modelo de Random Forest realizado en el chapter 3.
Params: 'n_estimators': 300, 'min_samples_leaf': 1, 'max_features': 'log2', 'max_depth': 30, 'class_weight': 'balanced'.

Modelo 3: consiste en un modelo de XG Boost realizado en el chapter 3.
Params: 'subsample': 0.6, 'reg_lambda': 0.3, 'reg_alpha': 0, 'n_estimators': 500, 'min_child_weight': 0, 'max_depth': 10, 'learning_rate': 0.05 'gamma': 0.3, 'colsample_bytree': 0.5

Modelo 4: consiste en una red neuronal realizada en el chapter 4.
Arquitectura: capa densa de 256 neuronas de entrada, 3 capas ocultas, una densa de 128, otra de 64 y una dropout de 0.2 y una neurona como capa de salida. También usa regularizaciones L1 , L2 y learning rate entre otras.

Conclusiones generales

Teniendo todo en cuenta, podemos afirmar firmemente que fue muy útil realizar un exhaustivo análisis exploratorio de datos por diversas razones, principalmente para tener una idea general del dataset sobre el que estamos trabajando y qué features podían llegar a tener más importancia sobre nuestro target. Además también ayudó considerablemente el preprocesamiento realizado, en diversas ocasiones probamos cómo darían los resultados de no haber hecho el preprocesamiento y siempre los resultados con el data frame original eran aproximadamente 2-4% menor.

Analizando los resultados de cada modelo, podemos concluir que los modelos que mejores resultados daban en el conjunto de test eran a su vez los que mejor resultado obtuvieron en la competencia de Kaggle, liderando con el modelo de XG Boost, sin embargo fue a su vez el modelo que más se demoraba en ser entrenado (de los 4 modelos nombrados). Algo para destacar es que siempre tuvimos en consideración la métrica de F1 score en conjunto de train para evitar el overfitting, sin embargo aquellos modelos que tenían mayor puntaje en conjunto de test también fueron los mayores en conjunto de train y en el Kaggle (Random Forest y XG Boost), por lo que podemos asumir que a pesar que los modelos si presentaban un sobre entrenamiento, generalizaron bien.

Siguiendo con este análisis, el modelo más sencillo y rápido sin dudas es el Árbol de decisión, donde entrenábamos y predecíamos casi instantáneamente sacrificando desempeño en las métricas. Sin embargo los árboles de decisión no fueron ni los más lentos ni los peores en términos de métricas, ya que KNN obtenía scores muy inferiores tardando considerablemente más y SVM obtiene métricas levemente inferiores pero tardando por lo menos 6 veces más. En nuestra opinión el mejor balance se encontraba en el Random Forest, donde obtuvimos muy buenos resultados muy similares a XG Boost tardando cierto tiempo menos. A pesar que las redes neuronales también eran muy rápidas comparadas a otros modelos, específicamente para este caso particular las redes neuronales no se ven muy beneficiadas, y quedó en evidencia con los valores de los scores.

Consideramos que el modelo podría llegar a ser utilizado de forma productiva si mejoramos levemente nuestro modelo para obtener scores más cercanos a 0,9 para tener mayor seguridad sobre las decisiones que se llevarán a cabo por nuestro modelo. Nuestro modelo podría ser utilizado por ejemplo en páginas web de reservas de hoteles, donde si hay una alta probabilidad de que un usuario cancele su

reserva podemos elevar el monto de depósito para mejorar las ganancias del hotel en aquellas reservas canceladas.

Analizando nuevamente todo el proyecto creemos que podría mejorarse por dos frentes. En primer lugar hay lugar a mejorar en el preprocesamiento de datos ya que hay varias decisiones que tomamos con la información que nos proveía el papel, sin embargo con un especialista en hotelería disponible es posible que se tomarían mejores decisiones a la hora de decidir valores atípicos, imputar datos faltantes o seleccionar features para nuestros modelos. Por otro lado, nuestros modelos se verían beneficiados de mayor capacidad de cómputo para encontrar mejores hiper-parámetros, ya que en su mayoría de veces encontrábamos mejores hiper-parámetros manualmente que a través del Random Search porque no realizamos la cantidad de iteraciones necesarias.

Tareas Realizadas

Integrante	Promedio Semanal (hs)
Juan Pablo Carosi Warburg	12
Mateo Daniel Vroonland	12