

Checkpoint 3 - Grupo 28

Introducción

Exploramos los distintos modelos vistos en clase:

KNN: En este modelo se analizaron distintos casos. Primero usamos los datos sin escalar y graficamos la cantidad de aciertos versus la cantidad de vecinos. Llegamos a la conclusión que a medida que aumentan los vecinos, la cantidad de aciertos disminuye. Luego analizamos, el modelo con los datos sin escalar y con los datos escalados con MinMax y Norm entrenamos al modelo. Usando MinMax Scaler, nos dio una métrica de f1 score de 0.8385. En cambio normalizando los datos, nos dio un F1 Score de 0.8390.

SVM: En este modelo se estudiaron los tres diferentes casos de kernels: lineal, radial y polinómico. Optimizamos para cada caso, sus hiperparametros utilizando GridSearch. En el caso de kernel lineal, obtuvimos una métrica de F1 0.797, luego analizamos el caso de kernel polinómico y obtuvimos una métrica de F1 Score 0.843. Como vemos, hubo una mejora en los resultados. Por último, obtuvimos en el caso de Kernel radial una métrica 0.8464 el cual tuvo una mejora de los otros casos.

Random Forest: En este modelo se estudió tres casos. En el primer caso hicimos un modelo utilizando parámetros aleatorios. Con esto se mejoró gradualmente los resultados, pasando de una predicción de 0.84873 a 0.8537.

Luego optimizamos los hiperparametros de random forest con validación cruzada usando grid search y randomized, note que con grid search da mejores resultados por lo que pasamos de una predicción de 0.8537 a 0.87611.

Por último optimizamos por medio de validación cruzada con grid search todas las métricas. Se obtuvo menor predicción a la anterior 0.87611 a 0.87586.

XGBoost: los hiperparámetros fueron ajustados mediante GridSearchCV. Se aplicó validación cruzada estratificada. El modelo logró un F1-score promedio de 0.88, demostrando un buen equilibrio entre precisión y exhaustividad.

Voting: En este caso como en Stacking se usó los mejores modelos mas usando árboles que se utilizó en etapas anteriores. Con esto pudimos obtener lo mejor de esos modelos . Con esto vimos que con respecto a la predicción más alto de random forest que no mejoró tanto , con la predicción de RF de 0.87611 a una predicción de Voting de 0.87487. Lo cual nos dio resultados no superiores a RF.

Stacking: En este caso entrenamos con los distintos modelos trabajados en Voting , sin bien tiene buenas métricas los resultados de la predicción nos dieron menor a la predicción de RF , lo cual no es una mejoría de 0.87611 a 0.87499.

Construcción del modelo

Detallar como mínimo los siguientes puntos:

- Hiperparámetros optimizados para KNN:
 - Weights: distance
 - N_neighbors : 9
 - Metric: euclidean
 - Algorithm: kd_tree
- Hiperparámetros optimizados para SVN:
 - C: 50
 - Gamma: 0.01
 - Kernel: rbf
- hiperparámetros optimizados para RF:
 - criterion= 'entropy'
 - min_samples_leaf=1
 - min_samples_split=5
 - n_estimators=90
 -
- Hiperparámetros optimizados para XGBoost
 - max_depth=15

- learning_rate=0.1
 - n_estimators=100
 - subsample=0.1
 - colsample_bytree=0.8
 - gamma=0.1
 - objective="binary:logistic"
 - random_state=42
- Modelos usados para el ensamble tipo voting:
Los modelos que se usaron para el ensamble fueron:
 - Arbol de decision
 - XGBoost
 - Random Forest
 - Modelos usados para el ensamble tipo stacking y la meta modelo (meta learner) del ensamble stacking:
Los modelos que se utilizaron fueron los que dieron resultados buenos:
 - Arbol de decision
 - XGBoost
 - Random Forest

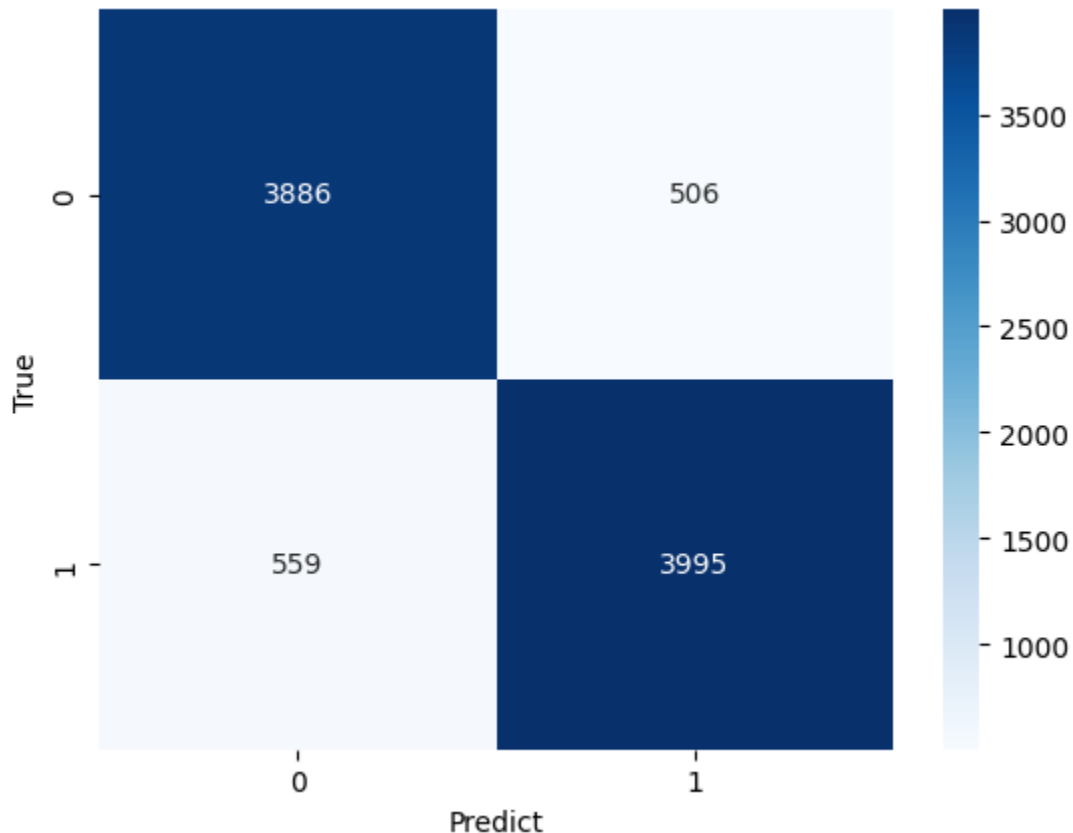
La meta modelo que se usó fue Random Forest con hiperparametros optimizados.

Cuadro de Resultados

Modelo	F1-Test	Precision Test	Recall Test	Accuracy	Kaggle
KNN	0.838568	0.8205183122	0.85743061	0.835084	0.53155
SVM	0.84644	0.8433675157	0.84955014	0.844876	0.33522
Random Forest	0.882385	0.887580537	0.8772507	0.880952	0.87611
XGBoost	0.88	0.88	0.88	0.88	0.86796
Voting	0.884365	0.886218302	0.882520	0.8825173	0.87487
Stacking	0.885583	0.882301656	0.888888	0.883076	0.87499

Matriz de Confusion

Nuestro mejor modelo según todo lo analizado anteriormente es Random Forest Cross Validation. Su matriz de confusión es:



- Precisión: Esta métrica se trata de los true positives sobre todos los casos positivos. En nuestro caso, tenemos un 0,8576 de métrica y quiere decir que una reserva es cancelada con un 85,76%
- Recall: Permite identificar la capacidad de nuestro modelo de encontrar correctamente los true positive. En nuestro caso, dice que el 87,72% de las reservas son efectivamente canceladas.
- F1-Score: Esta métrica combina la precisión y el recall, para obtener un valor mucho más objetivo. Es una medida ponderada entre el recall y la precisión. En nuestro caso, toma el valor de 0,8824
- Accuracy: Permite determinar el ratio de predicciones correctas, tanto como true positive y true negative. Esto quiere decir que en nuestro modelo, hay un 88,09% de predecir correctamente si la reserva es cancelada o no.

Nuestra métrica a considerar es f1-score ya que representa un ponderado de recall y precisión.

Tareas Realizadas

Integrante	Tarea
Jurgens, Cecilia Ines	KNN, SVM
Schipani, Martin Sebastian	XGBoost, Informe
Soto, Marilyn Nicole	Random Forest, Stacking, Voting