Informe Final - Grupo 03

Introducción

A lo largo de los checkpoints, los datasets sufrieron cambios en los que o bien eliminamos valores nulos (faltantes), o bien los imputamos con el valor más frecuente.
De entre las técnicas usadas podemos mencionar las que nos dieron mejor resultado: el **Random Forest**, **Ensambles Híbridos**, o bien las mismas **Redes Neuronales**.

Cuadro de Resultados

Modelo	CHPN	F1-Test	Precision	Recall	Accuracy	Kaggle	Hiperparámetros
Mejor Árbol de Decisión	2	0.8601539	0.8347310	0.8871739	0.8536935	0.84687	**criterion=**"gini" **max_depth=**15 **min_samples_split=**8
Mejor Random Forest	3	0.8824841	0.8940763	0.8711887	0.8823286	0.88087	**criterion=**"entropy" **min_samples_leaf=**1 **min_samples_split=**3 **n_estimators=**100 **random_state=**1
Ensamble Híbrido (Stacking)	3	0.8870505	0.8912113	0.8829284	0.8859671	0.87612	**estimators=**[RF, XGB] **final_estimator=***Logistic RegressionCV()* **cv=***KFold(**n_splits=**5)*
Mejor Red Neuronal	4	0.8563228	0.8559791	0.8566770	0.8542132	0.8515	**epochs=**500 **batch_size=**100 **capas=**4 (Dense) **neuronas=**(**1000, 100, 50, 1**) **optimizer=**Adagrad

El modelo que nosotros consideramos que es es nuestro mejor estimador de todo el trabajo práctico fue una iteración de **Random Forest**.

Conclusiones generales

De todos los modelos entrenados, parece ser que (a excepción de las redes neuronales), aquellas que funcionaban con cierto grado de aleatoriedad (como es el caso de *Random Forest*) dieron mejores resultados tanto en el test local como en Kaggle. Otros, como el clasificador *K-Nearest Neighbors* (KNN) no sólo tardaba mucho en calcular, sino que padecía de una varianza terriblemente alta (motivación para hacer PCA, por ejemplo).
Y es que justamente alguna vez se consideró hacer Análisis de Componentes Principales (PCA), pero entrado ya el desarrollo del TP1, se nos complicó optimizar por ese lado.

Algunos puntos clave a destacar también son:

- **El análisis exploratorio fue fundamental.**. Si bien son técnicas casi todas nucleadas en el CHP1 (y un poquito en el CHP2), lo cierto es que sin eso nos habríamos perdido muchos casos que corregir. ¡Había hasta una columna con *94% de valores faltantes*!

- Las tareas de preprocesamiento demostraron mejorar considerablemente la *performance* de nuestros modelos, las mejoras que realizamos en el primer checkpoint permitieron optimizar los modelos de los siguientes.

- El modelo de **ensamble híbrido** de tipo **Stacking** fue el que obtuvo la puntuación más alta en F1-score en los tests locales. La misma fue de **0.8870505**, aunque sospechamos de no muy buen balance en sesgo/varianza.

- Nuestra puntuación más alta en Kaggle la conseguimos con un modelo de **Random Forest**. La misma fue de **0.88087**. De nuevo, posible riesgo de *overfitting*.

- Los modelos de boosting como **XGboost** y el modelo **Random Forest** fueron considerablemente sencillos de crear y entrenar: fueron de lejos los más veloces en su entrenamiento; Y a su vez fueron los que mejor desempeño tuvieron.

- Consideramos que nuestro modelo, si bien quizás no esté optimizado al nivel profesional, funciona de forma muy efectiva a la hora de predecir las cancelaciones. Con algunas mejoras podría ser tenido en cuenta en un problema de la vida real.

- Quizás se podría mejorar nuestro modelo con un trabajo aún más exhaustivo de optimización de hiperparametros o utilizando alguna técnica más avanzada que nosotros desconocemos por nuestra falta de experiencia en el campo de la ciencia de datos. Pero algo que se nos ocurre es haber usado PCA para reducir la carga de datos al entrenar modelos.

Tareas Realizadas

Integrante	Promedio Semanal (hs)
Franco Lighterman Reismann	4.5
Marcos García Neira	2
Martín Andrés Maddalena	1