Modelos predictivos para la Tasa de Churn - TotalEnergies

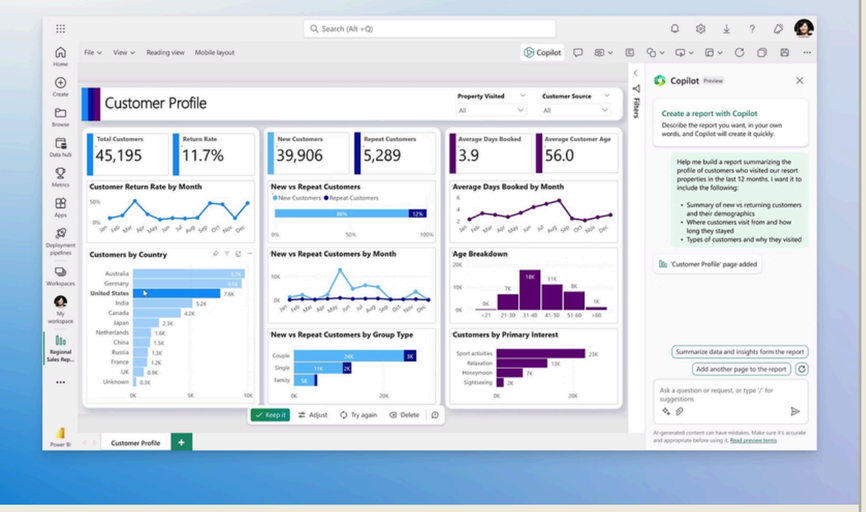
Iker Argüelles Eduarte

Doble Grado Ciencia e Ingeniería de Datos e Ingeniería en Servicios y Tecnologías de la Telecomunicación

Cátedra TotalEnergies de Analítica de Datos e Inteligencia Artificial

TotalEnergies

Universidad de Oviedo



01. Introducción

Mi experiencia en TotalEnergies se basó en una experiencia de 1 mes, durante todo julio, de la mano de la Universidad de Oviedo, y mediante el Grado de Datos.

La afronté como una manera de introducirme en el mundo de las empresas y el análisis de datos, pudiendo aprender a como se trabaja profesionalmente, y qué aplicaciones se usan.

02. Objetivo

Mi labor se basó en la documentación del trabajo y los modelos desarrollados por otros estudiantes de la EPI, que hacían prácticas allí a la vez que yo, o durante el año.

A su vez, pude ir nutriéndome de múltiples conocimientos sobre aprendizaje automático, y aplicaciones usadas en empresas.

Menciones espaciales al organizador de todo esto:

Luciano Sánchez Ramos

03. Metodología

Para poder llevar a cabo este trabajo, lo que hice fue recoger todos los archivos de mis compañeros, leerme información sobre lo que usaron, y luego ir exponieéndolo. Para ello guie de:

- **Scripts**
- **Power BI**
- **Documentación**
- **Apps externas**

Luego además, tuve que traducir todos los PowerPoint que hice, para poder ser presentados a grandes jerarquías de la empresa.

04. Conclusiones

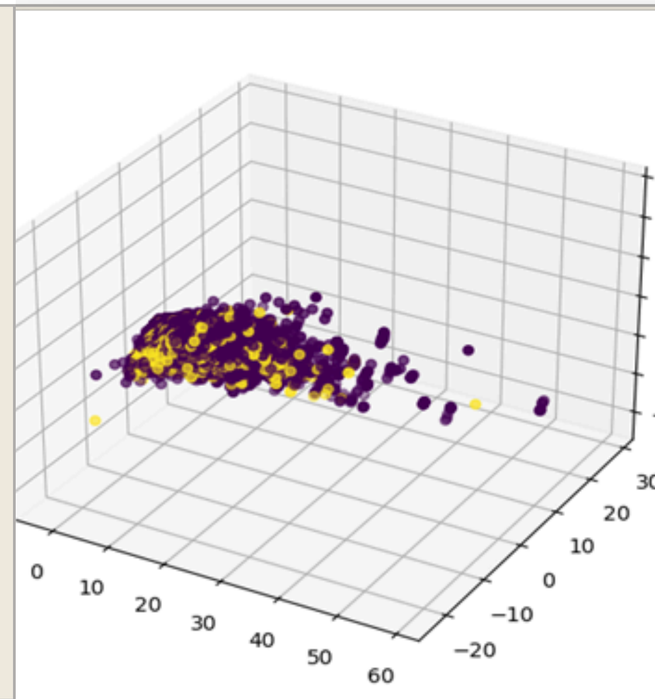
Es increíble ver cómo, a pesar del gran conocimiento de los ya trabajadores en este departamento, los internos siempre logran llegar algo más al asunto con nuevas técnicas, programas y conocimientos que al final son algo tan nuevo, que la vieja escuela puede no conocerlo.

Y una lista de cosas que he llegado a concluir de todas estas prácticas es:

- Hay múltiples maneras de analizar las variables para cada base de datos,

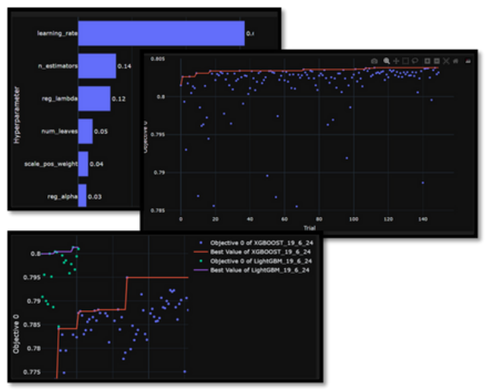
- Los modelos que a priori han sido mejores, han sido los basados en "boosting"

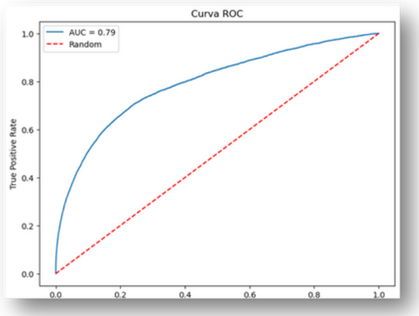
- Por último, hacer correcciones en las bases de datos anuales no es rentable porque con menos información se obtienen prácticamente los mismos resultados



05. Resultados y hallazgos

Básicamente, lo que se ha hecho es basarse en el resultado de las curvas ROC, centrándose en mejorar la AUC de los modelos. Para poder encontrar configuraciones de hiperparámetros se ha usado un programa externo llllamado “Optuna”, mediante el que se testean todos los modelos creados en Python, pudiendo ver las mejores configuraciones.





06. Trabajo futuro

Gracias a esta experiencia, cuento con una base para futuras prácticas que pueda hacer, y por supuesto, que me deja con mucho interés de cara a conocer nuevas empresas y entornos profesionales Creo que es algo a valorar por otros estudiantes, y más si estas haciendo el grado que mencioné, esto te abrirá muchas puertas en tu futuro laboral.

```python
# Se asigna cero a todas las variables con valor perdido
X_missing = X_full.copy()
X_missing[np.where(missing_samples)[0], missing_features] = 0
y_missing = y_full.copy()

# Estimacion del error con ceros en los valores perdidos
score = cross_val_score(estimator, X_missing, y_missing, cv=5).mean()
print("R2 del dataset con 75%% de valores perdidos, imputacion a 0= %.2f" % score)

# Se reemplazan los ceros por la media de la columna
estimator = Pipeline([("imputer", SimpleImputer(missing_values=0,
                                                strategy="mean")),
                      ("forest", RandomForestRegressor(random_state=0,
                                                       n_estimators=100))])
score = cross_val_score(estimator, X_missing, y_missing, cv=5).mean()
print("R2 del dataset con 75%% de valores perdidos, imputacion con la media = %.2f" % score)

# Se reemplazan los ceros por la mediana de la columna
estimator = Pipeline([("imputer", SimpleImputer(missing_values=0,
                                                strategy="median")),
                      ("forest", RandomForestRegressor(random_state=0,
                                                       n_estimators=100))])
score = cross_val_score(estimator, X_missing, y_missing, cv=5).mean()
print("R2 del dataset con 75%% de valores perdidos, imputacion con la mediana = %.2f" % score)
```