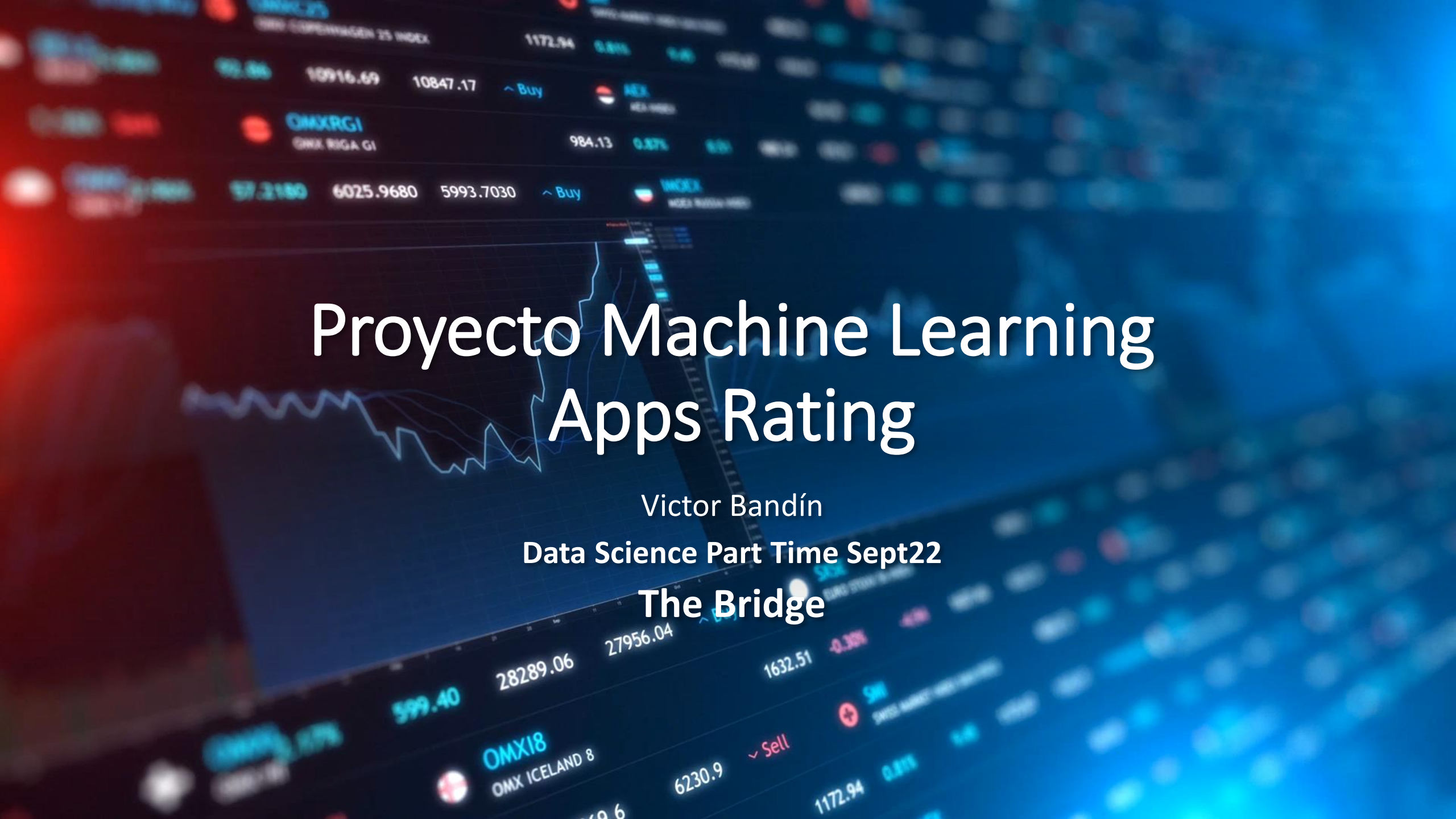


Proyecto Machine Learning Apps Rating

Victor Bandín

Data Science Part Time Sept22

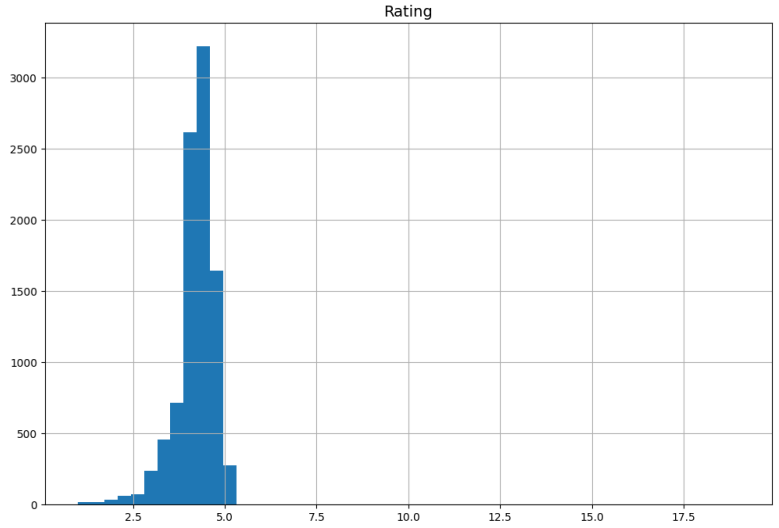
The Bridge

Temática y obtención de datos

- El objetivo de mi proyecto es predecir con la mayor precisión posible, a partir de un dataset publicado en Kaggle, el rating que los usuarios de Play Store de Google asignan a una determinada app a partir de sus características publicas: Número de descargas, ultima actualización, temática, etc.

- El objetivo de mi proyecto es predecir con la mayor precisión posible, a partir de un dataset publicado en Kaggle, el rating que los usuarios de Play Store de Google asignan a una determinada app a partir de sus características publicas: Número de descargas, ultima actualización, temática, etc.

Estructura del Dataset




```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 10841 entries, 0 to 10840
Data columns (total 13 columns):
 #   Column          Non-Null Count  Dtype
---  ------          --------------  -----
 0   App             10841 non-null  object
 1   Category        10841 non-null  object
 2   Rating          9367 non-null   float64
 3   Reviews         10841 non-null  object
 4   Size            10841 non-null  object
 5   Installs        10841 non-null  object
 6   Type            10840 non-null  object
 7   Price           10841 non-null  object
 8   Content Rating  10840 non-null  object
 9   Genres          10841 non-null  object
 10  Last Updated    10841 non-null  object
 11  Current Ver     10833 non-null  object
 12  Android Ver     10838 non-null  object
dtypes: float64(1), object(12)
memory usage: 1.1+ MB
```

```
data_report(X_train)
```

COL_N	Category	Reviews	Size	Installs	Type	Price	Content Rating	Current Ver	Android Ver	Main_Genre	Secondary_Genre	Days_Since_Last_Update
DATA_TYPE	object	int64	float64	int64	object	float64	object	object	object	object	int64	int64
MISSINGS (%)	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
UNIQUE_VALUES	33	4437	357	19	2	59	6	2074	28	47	2	1147
CARDIN (%)	0.5	67.72	5.45	0.29	0.03	0.9	0.09	31.65	0.43	0.72	0.03	17.51

Principales desafíos

El dataset utilizado contenía muchas lagunas de información (NaN) en el target.

Limpiar los datos para convertir las variables categóricas (todas menos el target) en numéricas cuando esto era posible/tenia sentido.

Identificar posibilidades para crear nuevas features relevantes que mejoraran la eficacia predictiva del modelo.

Probar diferentes opciones de preprocesado y comparar los resultados para elegir la mejor combinación de features, preprocesado, modelo e hiperparámetros para mejorar los resultados.

Modularizar el código mediante una serie de funciones en Python para ganar flexibilidad y eficacia a la hora cargar los datos, entrenar el modelo y valorar las predicciones, así como de alimentarlo con nuevos datos y guardar los modelos resultantes.

Problema de Machine Learning

Utilizaremos un algoritmo supervisado de regresión para predecir una variable continua (el Rating otorgado por los usuarios a la app).

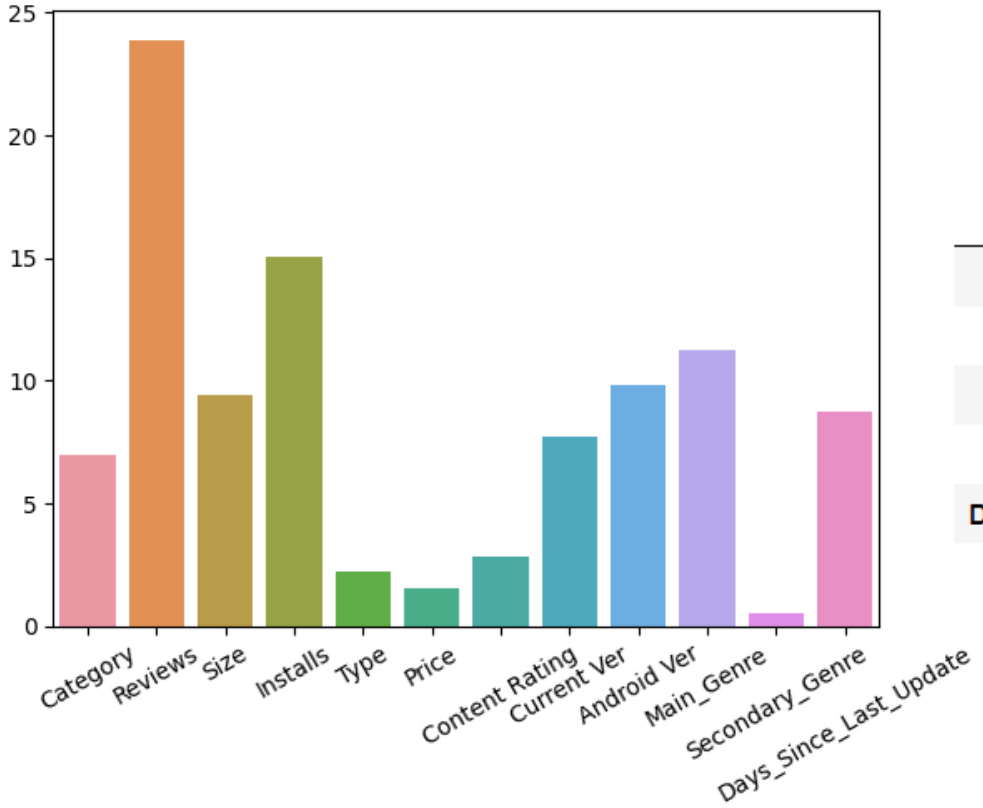
He probado:

- LinearRegression
- RandomForest
- XGBoost
- CatBoost

Feature engineering

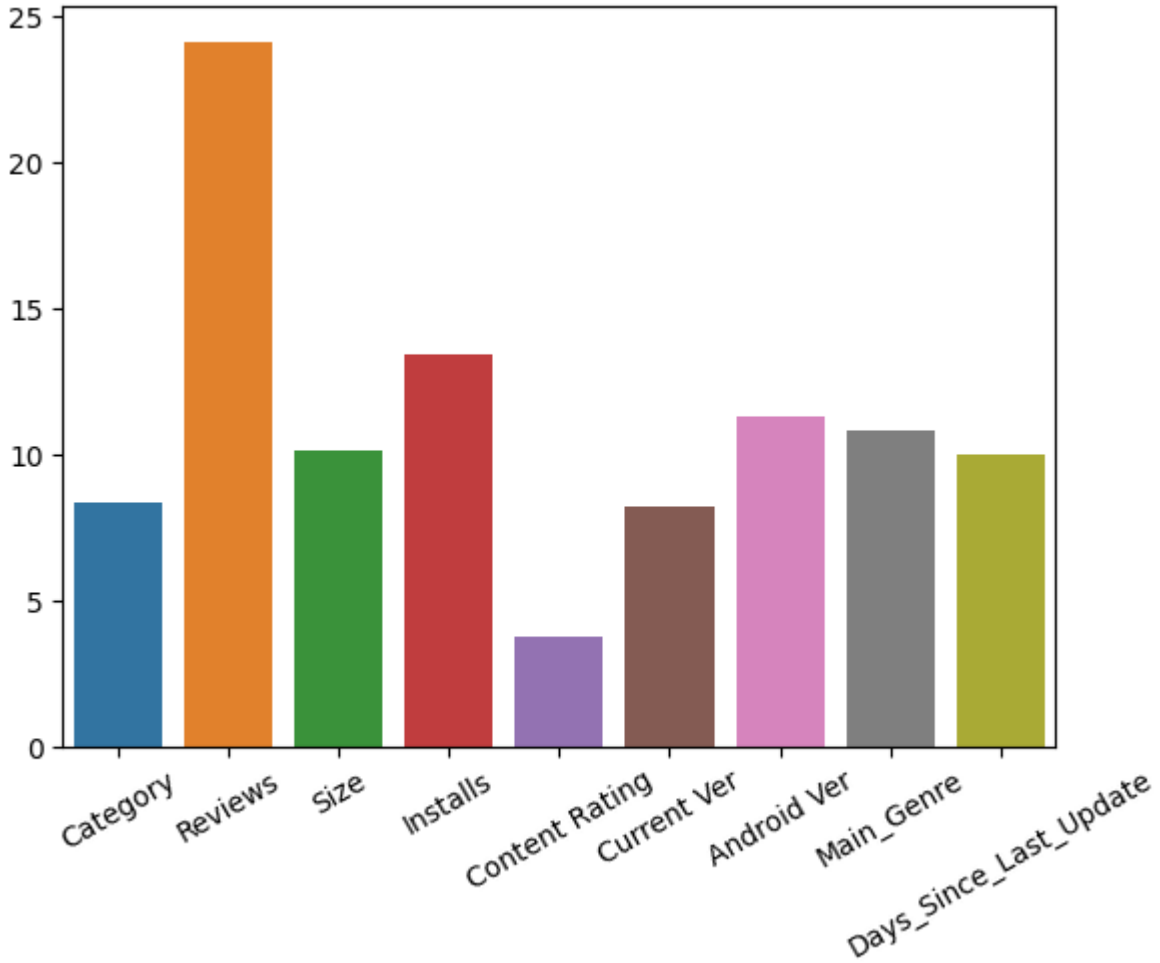
- Reducción de Features (Inicial)
- Conversión del tipo de variables y limpieza de datos.
- Pruebas de escalado de variables numéricas y transformación de categóricas (Ordinal y One Hot encoding).
- Creación de nuevas features.
- Reducción de Features (Final)

Selección de indicadores



	Rating	Reviews	Size	Installs	Days_Since_Last_Update
Rating	1.000000	0.068133	0.067508	0.051337	-0.142966
Reviews	0.068133	1.000000	0.103867	0.641605	-0.088182
Size	0.067508	0.103867	1.000000	0.044622	-0.193577
Installs	0.051337	0.641605	0.044622	1.000000	-0.104371
Days_Since_Last_Update	-0.142966	-0.088182	-0.193577	-0.104371	1.000000

Feature Importance Modelo Final



Conclusiones / Insights

- Crear una nueva feature como “Days_since_last_update ha mejorado significativamente los resultados del modelo.
- GridSearch me ha permitido identificar los mejores parámetros, si bien debido al tiempo que requería el entrenamiento he hecho pruebas segmentadas.
- He probado Optuna, pero un error me impedía guardar los resultados, si bien el RMSE de las pruebas era peor en todos los casos que el que obtuve previamente con GridSearch.
- Las pruebas de preprocesado con OneHotEncoder las he testado solo con CatBoost. XGBoost no funcionaba debido a que algunas de las features creadas no estaban presentes en el conjunto de test.
- En este proyecto, eliminar features “a mano” ha sido más eficaz que aplicar una PCA.
- Al tratarse de modelos basados en arboles, el escalado de variables no aportaba valor.
- El mejor modelo ha sido una combinación de limpieza de datos, reducción manual de features a partir del feature importance de los modelos previos, creación de una nueva feature y procesamiento directo de las categóricas restantes con CatBoost.


GRACIAS
POR SU ATENCION