



Instituto Tecnológico
de Buenos Aires

TRATAMIENTO DE DATASET, ANÁLISIS Y GRÁFICOS

Federico Kaplun

—

Score obtenido

Submission and Description

Public Score ⓘ



predicciones_finales (1).csv

Complete (after deadline) · 2h ago

0.69529

- **01: Introducción**
- **02: Tratamiento del DataSet**
- **03: Selección de modelos**
- **04: Conclusiones**

La base contiene información sobre alojamientos publicados en Airbnb en la ciudad de **Amsterdam**.



La base de entrenamiento cuenta con 4.928 registros con 67 variables.
La base de testeo cuenta con 1.233 registros, con 66 variables. La variable a predecir es *review_scores_rating*.

Preparación de la base:

- **Se eliminaron columnas nulas y que no aportaban ('id', 'host_id', 'host_name', 'license')**
- **Se confirma que la base no contiene duplicados**
- **Se normalizan columnas numéricas no preparadas (Columnas con porcentajes, dinero, etc.)**
- **Se normalizan las columnas de fecha**
- **Se formatean las columnas booleanas**
- **Se establecen criterios útiles para las columnas de texto**

Para las variables *'name', 'description', 'neighborhood_overview', 'host_about', 'amenities', 'host_verifications'*:

Primero se establecen **dos supuestos**:

- Una buena review juzga la veracidad de lo ofertado
- Entre mas descripcion hay, mas cercana a la realidad será la expectativa

Se cambió el contenido de las columnas por un **conteo de las palabras que ellas contienen**.

```
import re

text_columns = ['name', 'description', 'neighborhood_overview', 'host_about', 'amenities', 'host_verifications']

for column in text_columns:
    df[column] = df[column].apply(lambda x: len(re.findall(r'\b\w+\b', x.lower())) if isinstance(x, str) else 0)
```

Más criterios para variables de texto:

- Para las variables categoricas ('room_type', 'property_type', 'neighbourhood_cleansed', 'host_neighbourhood', 'host_location', 'host_response_time') se realizo un *encoding*, cambiando cada categoría por un número entero
- Para las fechas ('host_since', 'calendar_last_scraped', 'first_review', 'last_review'), se estableció como base (1) la fecha más antigua, y el resto son números enteros partiendo desde ahí

| | host_since | calendar_last_scraped | first_review | last_review |
|---|------------|-----------------------|--------------|-------------|
| 0 | 795 | 4502 | 4350 | 4354 |
| 1 | 1879 | 4502 | 4249 | 4409 |
| 2 | 2911 | 4502 | 4298 | 4308 |
| 3 | 2116 | 4502 | 4258 | 4272 |
| 4 | 4074 | 4502 | 4333 | 4468 |

Entrenamiento de modelos:

Luego de todos estos cambios, obtenemos un DataFrame de xxx variables, aun con xxx registros.

Para entrenar a los modelos, se realizó una partición del dataset de entrenamiento de 80-20.

Se utilizó el método Random Search, ampliando los parámetros como se vio necesario

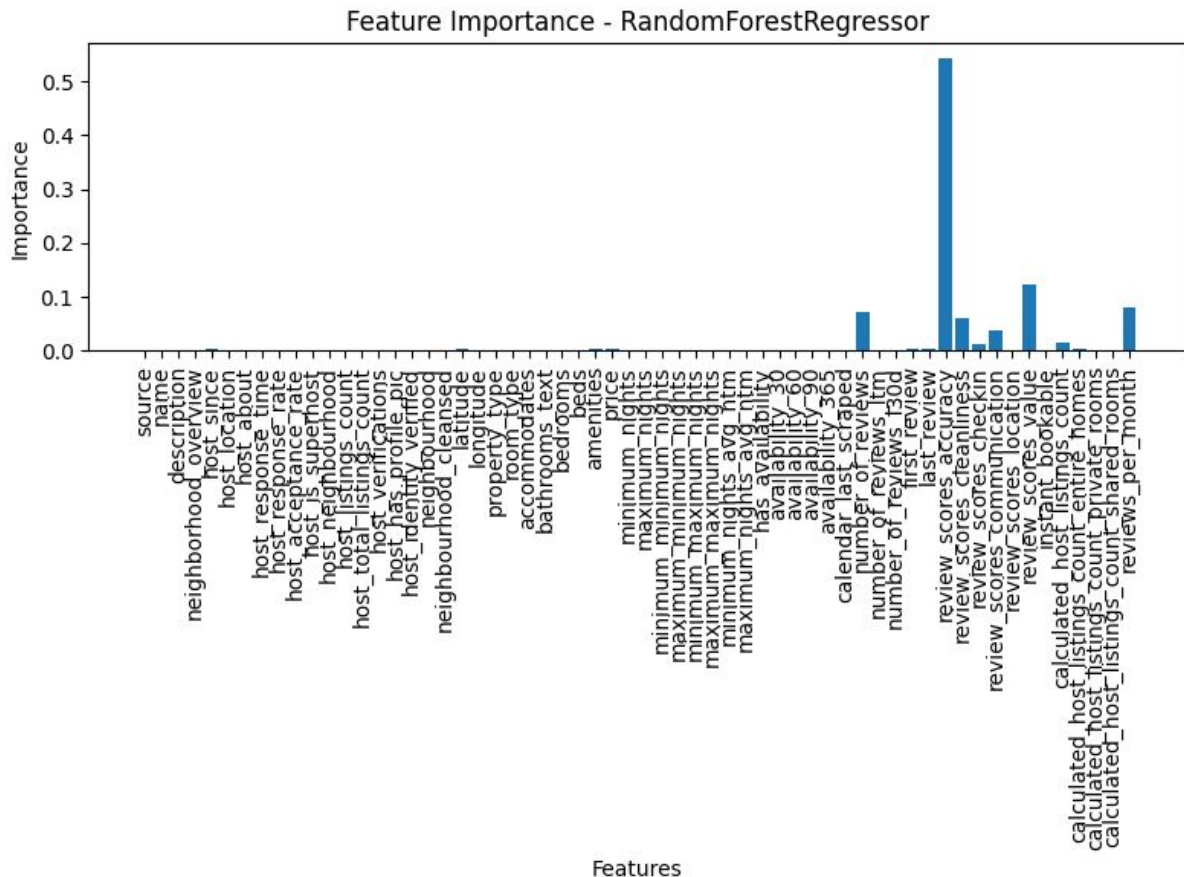
Modelos utilizados:

- XGBRegressor
- Linear Regression
- Extra Trees Regressor
- **Random Forest Regressor**
- CatBoost Regressor
- LGBMRegressor
- Decision Tree Regressor
- AdaBoost Regressor
- SVR (Support Vector Regression)



Modelo elegido: Random Forest Regressor

De todos los modelos probados, consistentemente devolvió el r^2 más alto. Para mejorarlo aún más, se realizó un análisis de *feature importance*, lo que reveló que variables fueron las más utilizadas para el modelo:



Selección de modelo:

Aunque el *feature analysis* no haya dado frutos (mejor r^2 sin *feature analysis* 0.8182, mejor r^2 con *feature analysis*, 0.7603), el Random Forest Regression sigue siendo el mejor modelo.

Random Forest Regression

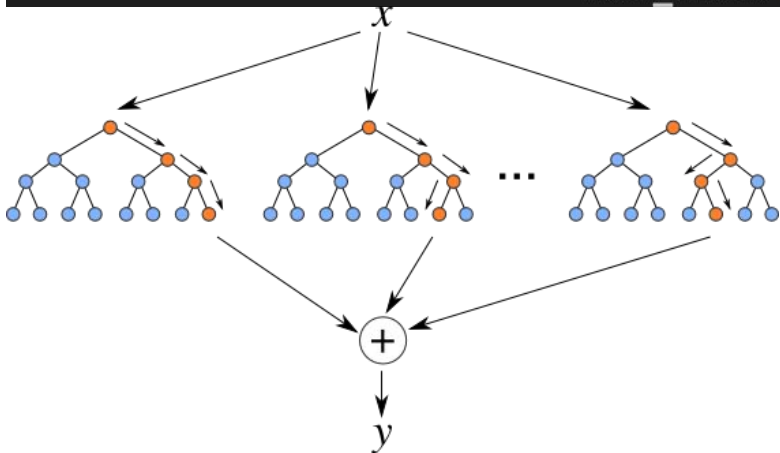
Se crean múltiples árboles de decisión, y cada uno realiza una predicción individual. Luego las predicciones se promedian para obtener la predicción final

Ventajas:

- Precisión
- Escalabilidad
- Robustez

Mejores hiperparametros:

```
model = RandomForestRegressor(bootstrap=True, ccp_alpha=0.0, criterion='squared_error',  
                             max_depth=None, max_features=1.0, max_leaf_nodes=None,  
                             max_samples=None, min_impurity_decrease=0.0, min_samples_leaf=1,  
                             min_samples_split=2, min_weight_fraction_leaf=0.0, n_estimators=100,  
                             n_jobs=None, oob_score=False, random_state=None, verbose=0,  
                             warm_start=False)
```





Instituto Tecnológico
de Buenos Aires

¡Gracias!