Diamonds train – datos CON precio

Diamond test – datos SIN precio, pero que se usan para calificar el modelo. Tiene un id

sample submission – csv que se hace con id, y\_pred

Se eligen dos de entrega y el private leaderboard se queda con el mejor de los dos.

Tareas

Basico:

categorías, one hot encoding, scaler, ---- todas las transformaciones se hacen a X (train y test!)

Ideas:

Gradient boosting

Incluir validacion cruzada

Incluir filtros

Outliers

eliminar columnas espureas → comprobar

procesamiento de columnas

add features: construir nuevas columnas

Matriz de modelado

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Modelo | Train error | Error CV | Error test (kaggle) | Parametros |
| RandomForestRegressor  w/o grid | 200 | 555 | 561 |  |
| RandomForestRegressor  with grid |  | 557.5 | 557.9 |  |
| Gradient Boosting  w/o grid | 712 | 733 | - |  |
| With grid (learn) | - | 655 | - | Imputer=mean  Nestimators=100  Learn\_rate=0.2 |
| With grid (tree) | - | 558 | - | 'regressor\_\_subsample': 1,  'regressor\_\_n\_estimators': 120,  'regressor\_\_min\_samples\_leaf': 1,  'regressor\_\_max\_depth': 8 |
| With grid (pushing to the limit) | - | 549 | 548 | 'regressor\_\_subsample': 1, (default!)  'regressor\_\_n\_estimators': [100,125, 150, 175, 200]  'regressor\_\_min\_samples\_leaf': 1 (default!)  'regressor\_\_max\_depth': 8,  'regressor\_\_learning\_rate': [0.15,0.2, 0.25, 0.3] |
| With grid (pushing to the limit) con rescale (sin projection) | - | 540 |  | ↑ |
| ~~With grid (pushing to the limit) + projection con menos dimensiones (19)~~ | ~~-~~ | ~~630~~ |  | ~~↑~~ |
| ~~With grid (pushing to the limit) + projection con menos dimensiones (14)~~ | ~~-~~ | ~~685~~ |  | ~~↑~~ |
| LightGBM | - |  |  |  |
| - Con grid | - | 540.9 | 538.5 | {'regressor\_\_boosting': 'dart', 'regressor\_\_learning\_rate': 0.05, 'regressor\_\_max\_depth': 16, 'regressor\_\_n\_estimators': 2000, 'regressor\_\_num\_leaves': 100}  cv=5!!! |
| Trying DART\_optimiz | - |  |  | 'regressor\_\_boosting': ['dart'],  'regressor\_\_learning\_rate': [0.001, 0.01, 0.05],  'regressor\_\_n\_estimators': [1000],  'regressor\_\_max\_depth': [ 6, 8, 16],  'regressor\_\_num\_leaves':[50,100,200] |
| DART con variables nuevas |  |  |  |  |
| NN | 501 | - | 585 |  |

Probar:

- Revisar

~~- Drop columns~~

- Outliers – Robust scaler?

~~- Unsupervised para hacer projection~~

- Unsupervised para hacer clusters

- Unsupervised para hacer projection + clusters

https://neptune.ai/blog/lightgbm-parameters-guide

drop\_seed: random seed to choose dropping models

Uniform\_dro:set this to true, if you want to use uniform

dropxgboost\_dart\_mode: set this to true, if you want to use xgboost dart

modeskip\_drop: the probability of skipping the dropout procedure during a boosting iteration max\_drop

drop\_rate: dropout rate: a fraction of previous trees to drop during the dropout

default gbdt vs dart = gbdt

improved gbdt vs dart vs goss (n\_estimators=500, max\_depth=16) = **dart**

testing num\_leaves (default 31) = **≈40-50**  → 547

n\_estimators - **optimo entorno 1000**

max\_depth

learning rate = **default 0.1**

std scaler vs robust scaler **indiferente**

std scaler doble (cat+num) 521.54 → 524 Target

Descartes

robust es peor

drop=first es peor