Diamond's Competition



DD BB Cleanning

- Observamos la correlación .

17.667809

Name: price/carat, dtype: float64

 $_{
m IF}$

- Otorgamos un valor a las columnas no numéricas.

Prueba de Random Forest

- Primero con Split 80% y 20%, Luego con todo X.
- Params : n_estimators = 1.000 n_jobs = -1

```
cut
click to scroll output; double click to hide
               11.663682
                                                             from sklearn.ensemble import RandomForestRegressor
 Premium
               11.807049
                                                             from sklearn.metrics import mean_squared_error
  Very Good
               12.446719
               13.893712
  Ideal
                                                             rf = RandomForestRegressor(
 Name: price/carat, dtype: float64
                                               In [49]:
                                                                          n estimators = 1000,
  color
                                                                          n jobs = -1
        9.017189
       10.422280
       11.510122
                                                          1 rf.fit(X, y)
                                               In [50]:
       12.992913
       13.141893
                                               Out[50]: RandomForestRegressor(n_estimators=1000, n_jobs=-1)
       14.133644
       14.300001
                                                             y pred = rf.predict(X)
                                               In [52]:
 Name: price/carat, dtype: float64
 clarity
                                               In [53]:
                                                             mean squared error(y, y pred)
           7.466673
 Ι1
  SI2
           9.212076
                                               Out[53]: 0.0011652131965208597
 SII
          11.440938
  VS2
          13.008773
          13.512145
  VS1
         17.459082
```

Instalación H2o y modelo

- Instalación de H2o y Java.
- Asignación X e y.
- Params : max_run_time sec = 2.000

```
In [20]: 1 X = list(data.drop(columns = ['price', 'id'] ).columns)
            1 automl = H2OAutoML(max_runtime_secs = 2500)
            automl.train(X,y,training_frame = h2train)
          AutoML progress:
            leader board = automl.leaderboard
            2 leader board.head()
                                                  model_id mean_residual_deviance
           StackedEnsemble_BestOfFamily_AutoML_20210228_180054
                              GBM_4_AutoML_20210228_180054
                              GBM_3_AutoML_20210228_180054
              StackedEnsemble_AllModels_AutoML_20210228_180054
                 GBM_grid__1_AutoML_20210228_180054_model_10
                              GBM_2_AutoML_20210228_180054
                  GBM_grid__1_AutoML_20210228_180054_model_3
                              GBM_1_AutoML_20210228_180054
                                                                      0.00801619 0.0895332 0.00801619 0.0644385 0.0101862
               XGBoost_grid__1_AutoML_20210228_180054_model_7
                                                                      0.00802058  0.0895577  0.00802058  0.0640437  0.0101733
               XGBoost_grid__1_AutoML_20210228_180054_model_1
                                                                      0.00822907 0.0907142 0.00822907 0.0641784 0.0102698
```

H20 MSE = 0.00771845