Diamonds Projeto

14/jan/2020

Robson Silva

Entendendo as variáveis

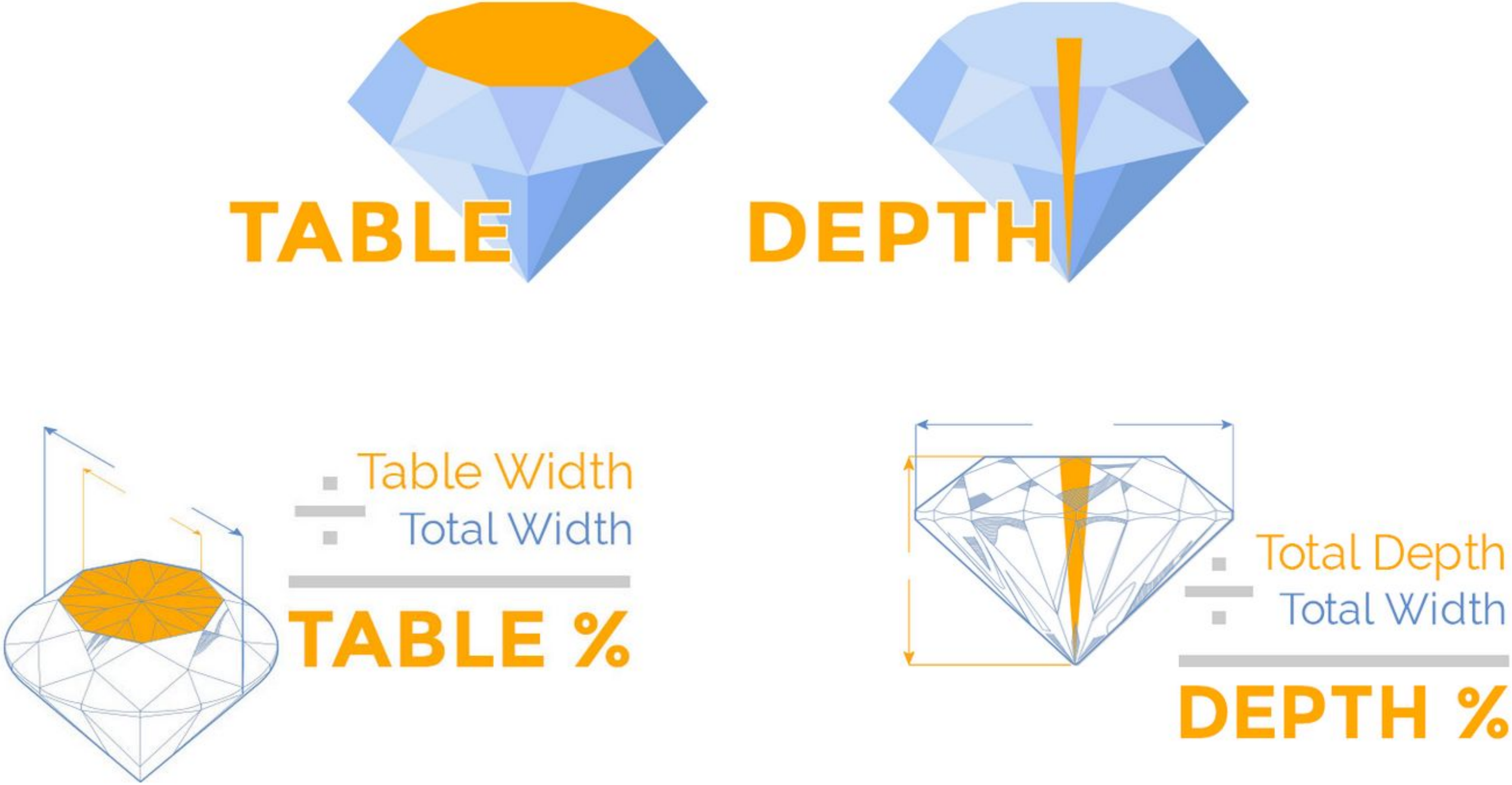
Column	Description
Price	Price in US dollars (326-18,823)
Carat	Weight of the diamond (0.2--5.01)
Cut	Quality of the cut (Fair, Good, Very Good, Premium, Ideal)
Color	Diamond colour, from J (worst) to D (best)
Clarity	A measurement of how clear the diamond is (I1 (worst), SI2, SI1, VS2, VS1, VVS2, VVS1, IF (best))
x	Length in mm (0--10.74)
y	Width in mm (0--58.9)
z	Depth in mm (0--31.8)
Depth	Total depth percentage = z / mean(x, y) = 2 * z / (x + y) (43--79)
Table	Width of top of diamond relative to widest point (43--95)

4C's

"Diamond carat weight is the measurement of how much a diamond weighs. A metric "carat" is defined as

200 milligrams."

https://4cs.gia.edu/en-us/diamond-carat-weight



https://www.brilliance.com/education/diamonds/depth-table

“

Essentially, anything between 50 and 69 percent is considered alright. However, the **most ideal TABLE PERCENTAGES are between 54 and 60 percent.** At this proportion, the table is large enough to allow light to enter the stone at the correct angles to reflect and refract off the smaller facets below.”

https://www.brilliance.com/education/diamonds/depth-table

Análises do Dataset

- Criação de novas features;
- Regressão Linear.

Inserindo novas colunas:

- Volume;
- Densidade;
- Carat x Volume;
- **3Cs:** Clarity x Cut x Color
- 3Cs x Vol

Dataset pronto para Regressão:

	carat	cut	color	clarity	depth	table	price	x	y	z	clarityNew	colorNew	cutNew	volume	density	carat*Vol	3Cs	3Csxvol
0	0.23	Ideal	E	SI2	61.5	55.0	326	3.95	3.98	2.43	2	6	5	10.001411	0.022997	2.300325	60	600.084672
1	0.21	Premium	E	SI1	59.8	61.0	326	3.89	3.84	2.31	3	6	4	9.033990	0.023246	1.897138	72	650.447276
2	0.23	Good	E	VS1	56.9	65.0	327	4.05	4.07	2.31	5	6	2	9.968566	0.023073	2.292770	60	598.113939
3	0.29	Premium	I	VS2	62.4	58.0	334	4.20	4.23	2.63	4	2	4	12.232621	0.023707	3.547460	32	391.443883
4	0.31	Good	J	SI2	63.3	58.0	335	4.34	4.35	2.75	2	1	2	13.591922	0.022808	4.213496	4	54.367689

Definindo uma função para Múltiplas Análises

```python
## Function for comparing different approaches
from sklearn.model_selection import train_test_split
def score_dataset(df):
    model = LinearRegression()
    y = df.price
    results = []
    for i in list(df.columns):
        if type(df[i][0]) != str:
            X = df[[i]]
            X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state = 0)
            model.fit(X_train, y_train)
            preds = model.predict(X_test)
            results.append( (i, mean_absolute_error(y_test,preds), model.score(X_train, y_train) ) )
    return results
```

Analisando a performance

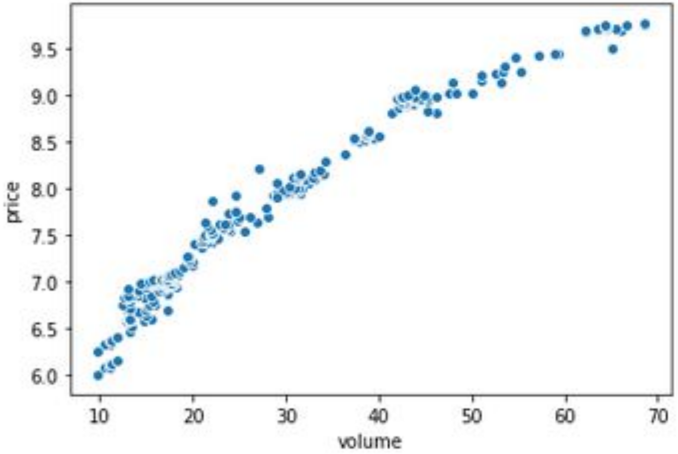
```
1 score_dataset(diamondsNew)
```

```
[('carat', 1010.6385120654585, 0.8496675874577887),
 ('depth', 3062.993107599667, 0.00015243252862717777),
 ('table', 3013.202731588257, 0.01649906126461831),
 ('price', 4.2722411210737817e-13, 1.0),
 ('x', 1371.3344701089952, 0.7869746343926203),
 ('y', 1369.231589508964, 0.7891432918567728),
 ('z', 1390.5284414225846, 0.7778587353338537),
 ('clarityNew', 2970.645934437729, 0.021697735375831306),
 ('colorNew', 2987.5658365267927, 0.030145110949522436),
 ('cutNew', 3048.1165441561043, 0.002819880901487881),
 ('volume', 1001.8440202571396, 0.8528797670926731),
 ('density', 2993.550579312727, 0.02004459763267541),
 ('carat*Vol', 1036.6571502159306, 0.7989266741237522),
 ('3Cs', 2921.741694038112, 0.037645135414547504),
 ('3Csxvol', 2474.637933069794, 0.2849633172127253)]
```

Aplicando a realidade

```
data = diamondsNew.query('clarityNew == 5 and colorNew == 6 and cutNew == 4')
```

```
sns.scatterplot(x = data.volume , y= np.log(data.price));
```



Mean Absolute Error: 0.1328
Score: 0.9580
r2_score ScikitLearn: 0.963

Regressão usando múltiplas variáveis

```python
X = diamondsNew[['clarityNew','colorNew','cutNew','volume']]
y = diamondsNew.price
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state = 0)
model = LinearRegression()
model.fit(X_train, y_train)
preds = model.predict(X_test)
print(f'Mean Absolute Error: {mean_absolute_error(y_test,preds)}')
print(f'Score: {model.score(X_train, y_train)}')
print(f'r2_score ScikitLearn: {r2_score(y_test, preds)}')
```

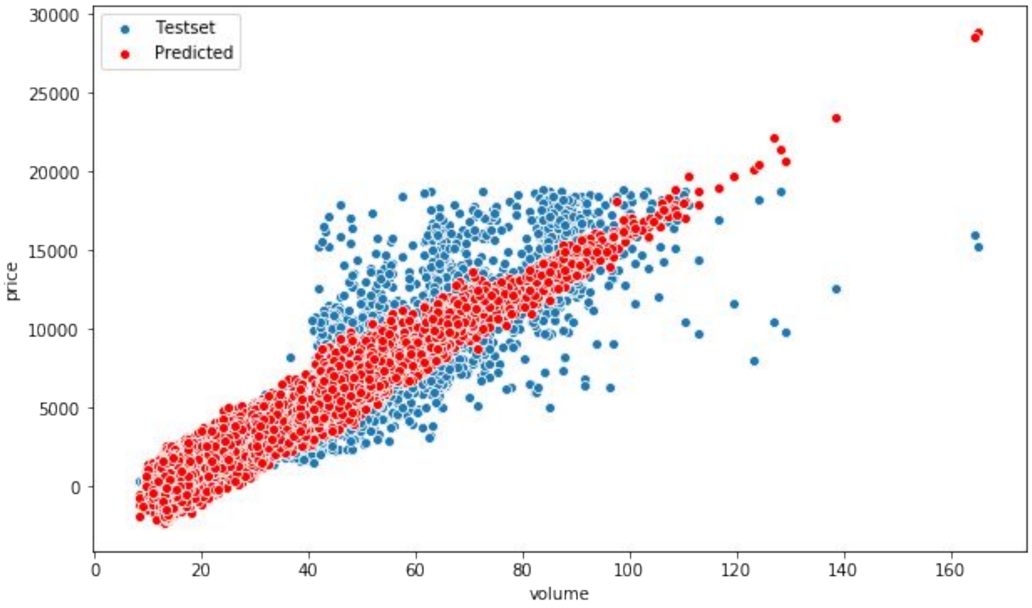
$$R^2_{adj} = 1 - (1 - R^2)\frac{n-1}{n-p-1}$$

N = is the number of points of data sample.
p = is the number of independent regressors.

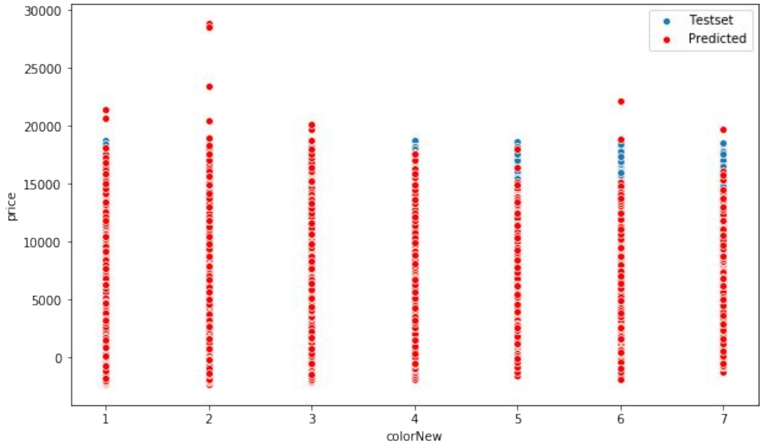
Mean Absolute Error: 856.4662
Score: 0.9050
r2_score Scikit Learn: 0.9052284

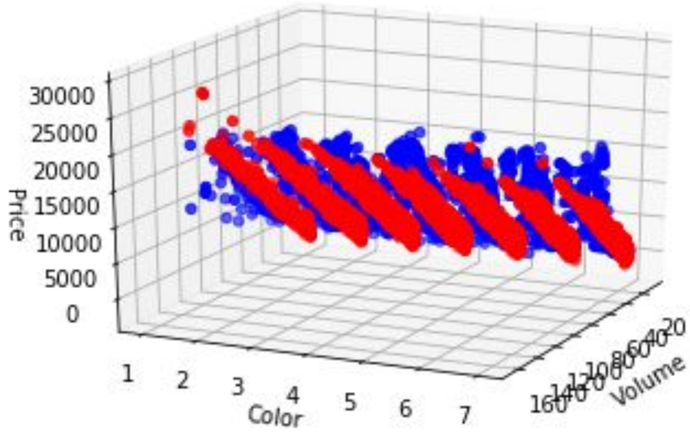
R2 adjusted: 0.9051933

Analisando os gráficos



Analisando os gráficos





Obrigado!

SlidesCarnival

Free templates for all your presentation needs

For PowerPoint and Google Slides

100% free for personal or commercial use

Ready to use, professional and customizable

Blow your audience away with attractive visuals