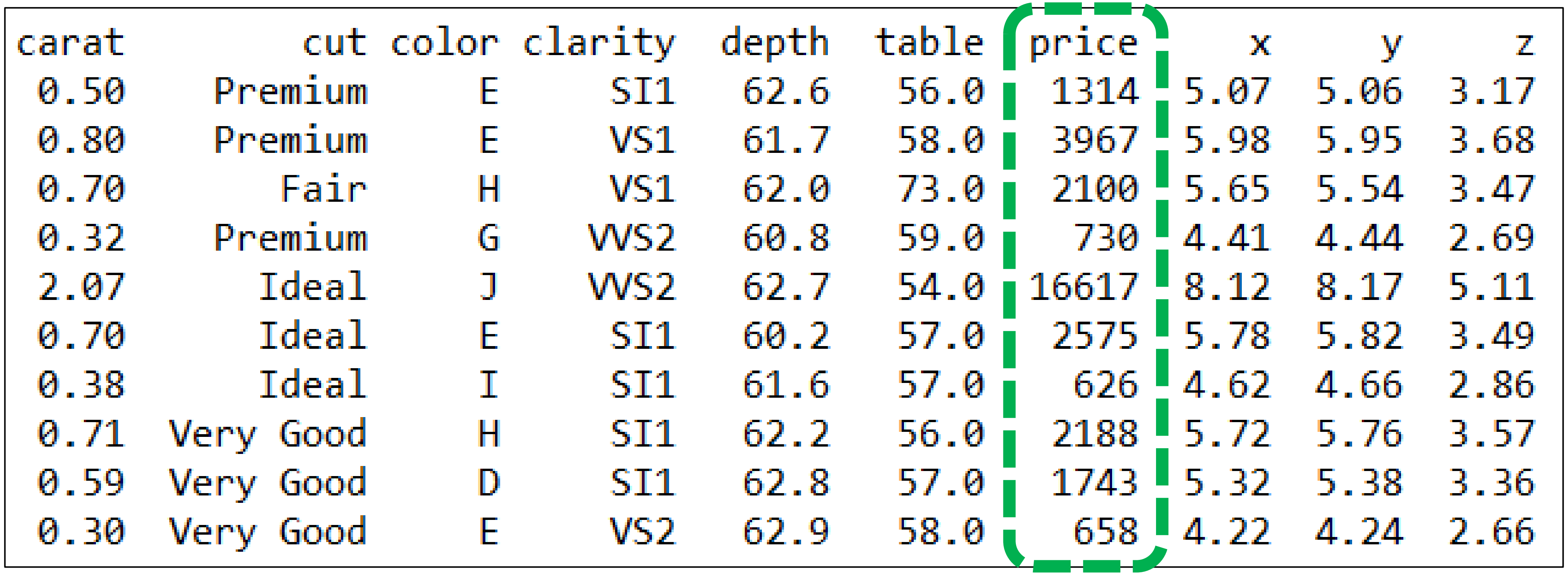
Diamond Price Estimation

Kutay Erkan

25.12.2018

The Dataset

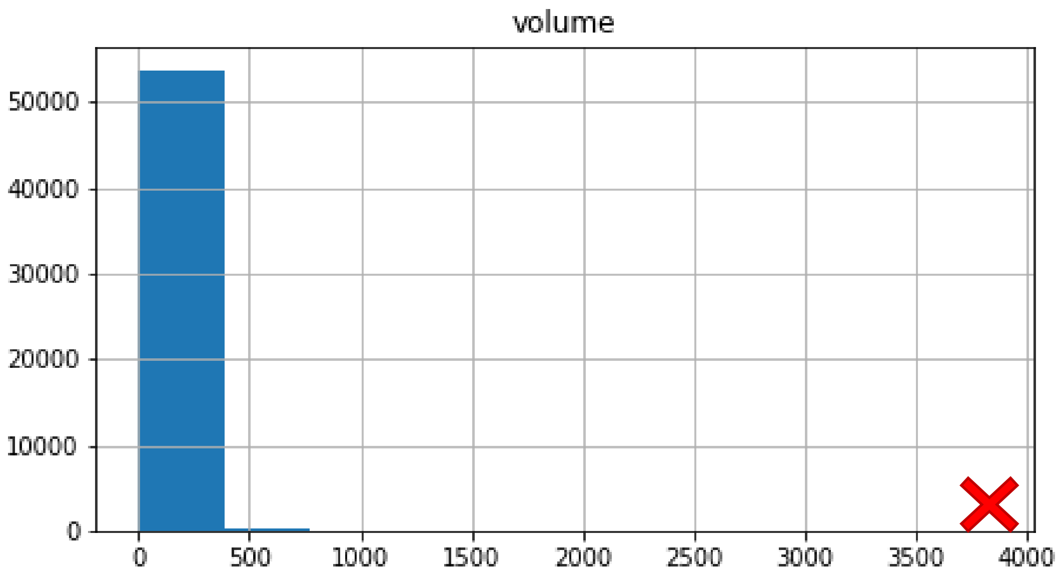
- 1 target
- 10 features
  - Color → D: best, J: Worst
  - Clarity → FL (Flawless), IF, VVS1, VVS2, VS1, VS2, SI1, SI2, I1, I2, I3

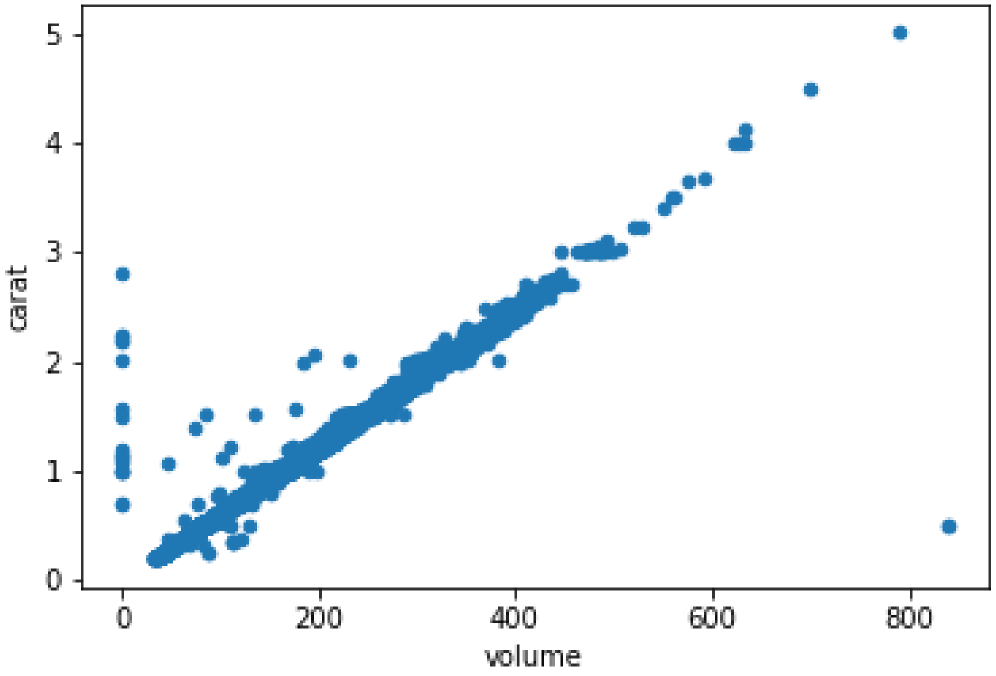


```
carat       cut color clarity  depth  table  price     x     y     z
 0.50   Premium     E     SI1   62.6   56.0   1314  5.07  5.06  3.17
 0.80   Premium     E     VS1   61.7   58.0   3967  5.98  5.95  3.68
 0.70      Fair     H     VS1   62.0   73.0   2100  5.65  5.54  3.47
 0.32   Premium     G    VVS2   60.8   59.0    730  4.41  4.44  2.69
 2.07     Ideal     J    VVS2   62.7   54.0  16617  8.12  8.17  5.11
 0.70     Ideal     E     SI1   60.2   57.0   2575  5.78  5.82  3.49
 0.38     Ideal     I     SI1   61.6   57.0    626  4.62  4.66  2.86
 0.71 Very Good     H     SI1   62.2   56.0   2188  5.72  5.76  3.57
 0.59 Very Good     D     SI1   62.8   57.0   1743  5.32  5.38  3.36
 0.30 Very Good     E     VS2   62.9   58.0    658  4.22  4.24  2.66
```

Feature Engineering

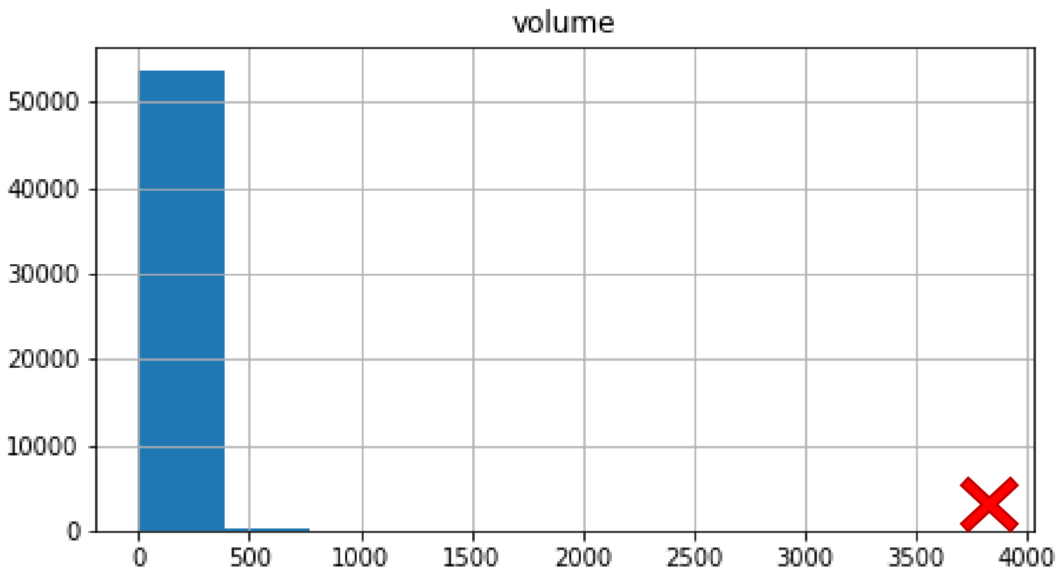
- df['volume'] = df['x'] * df['y'] * df['z']

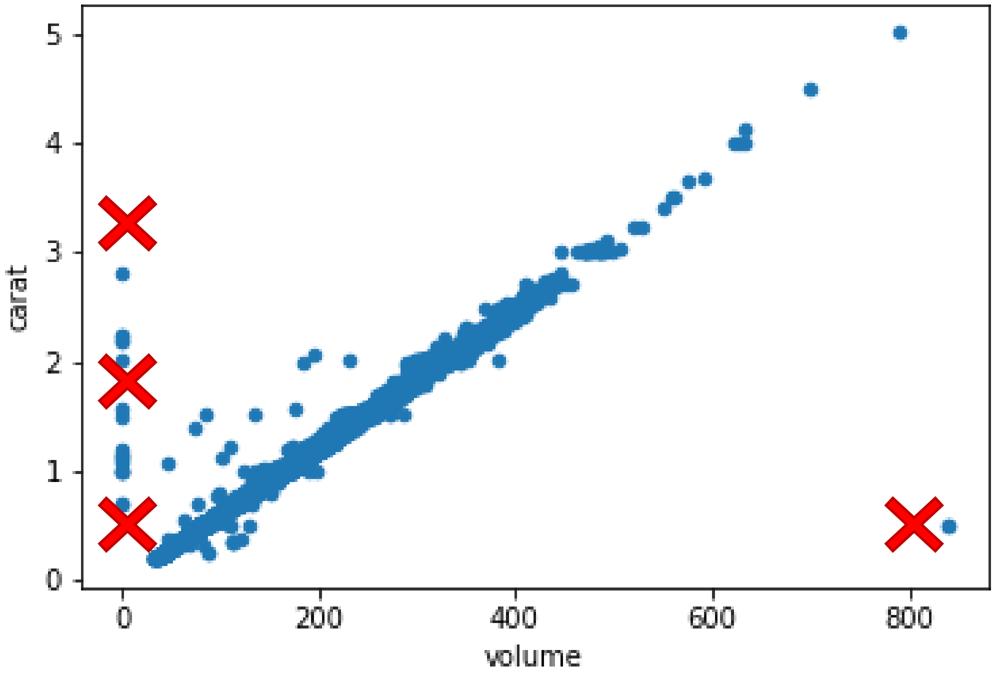




Feature Engineering

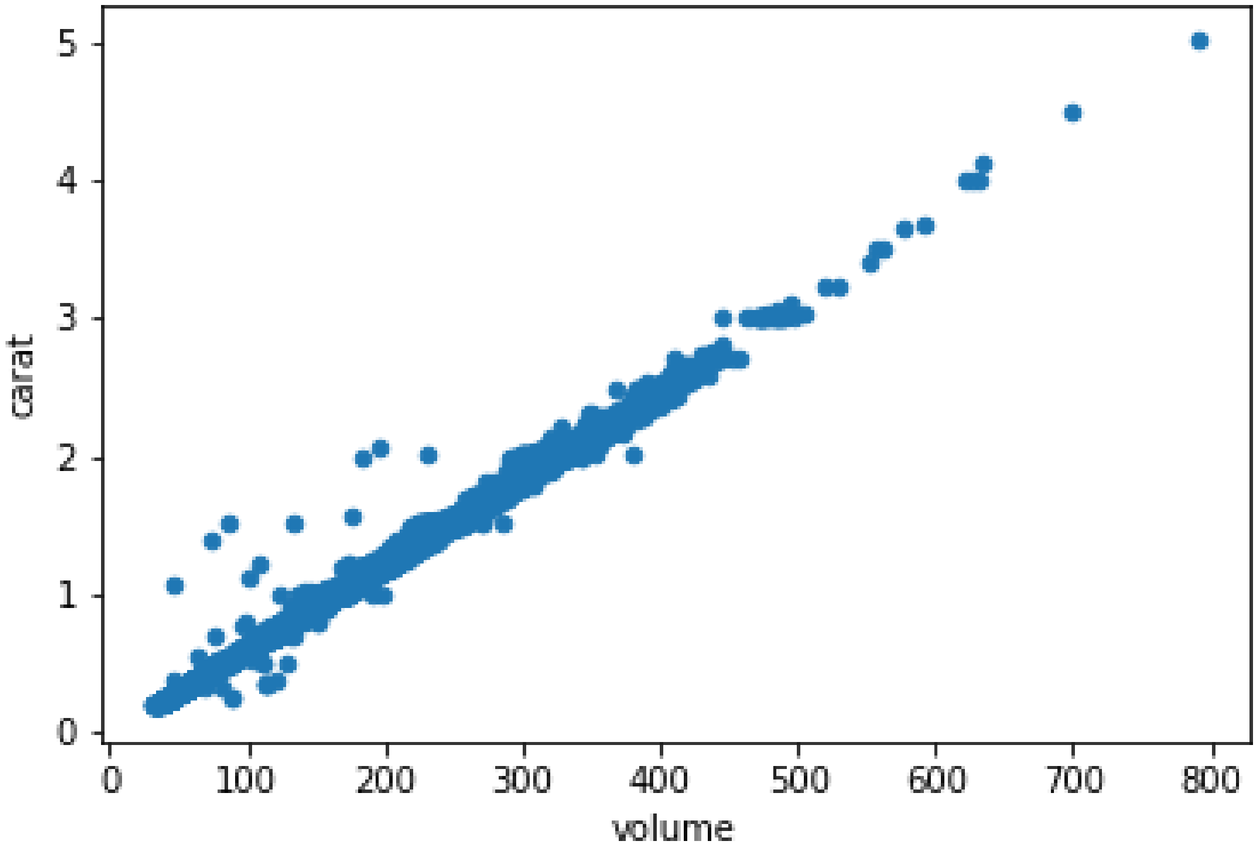
- df['volume'] = df['x'] * df['y'] * df['z']





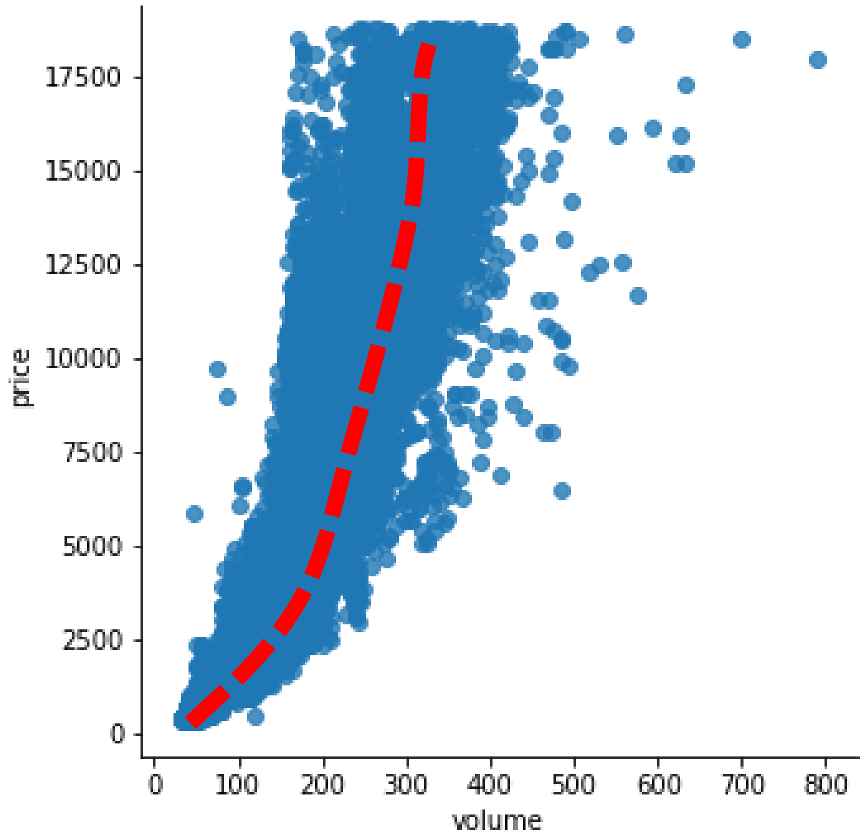
Feature Engineering

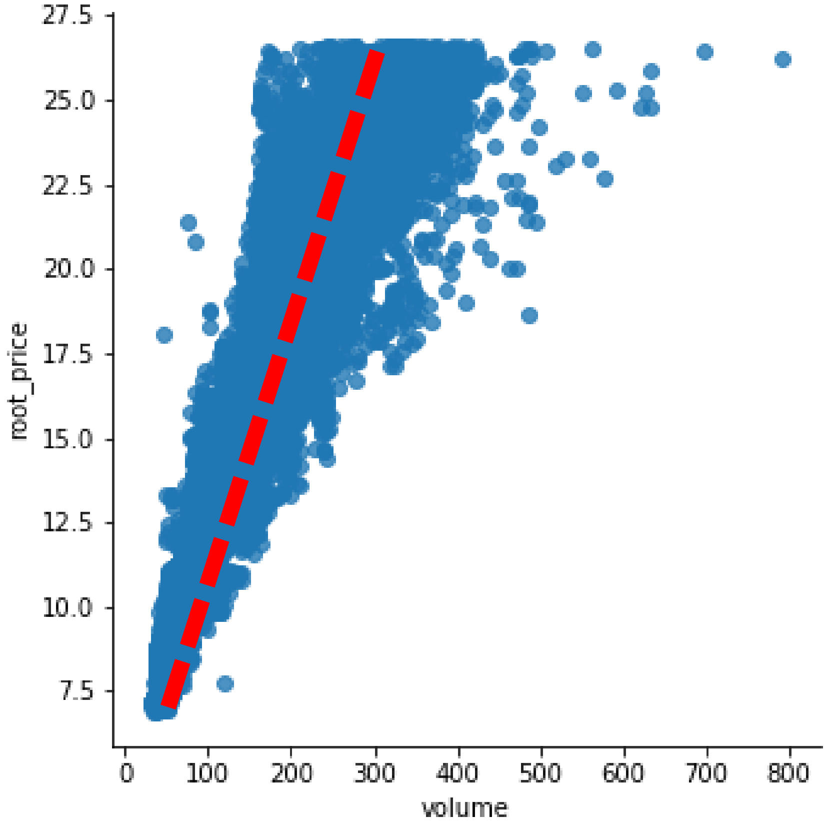
- df['volume'] = df['x'] * df['y'] * df['z']



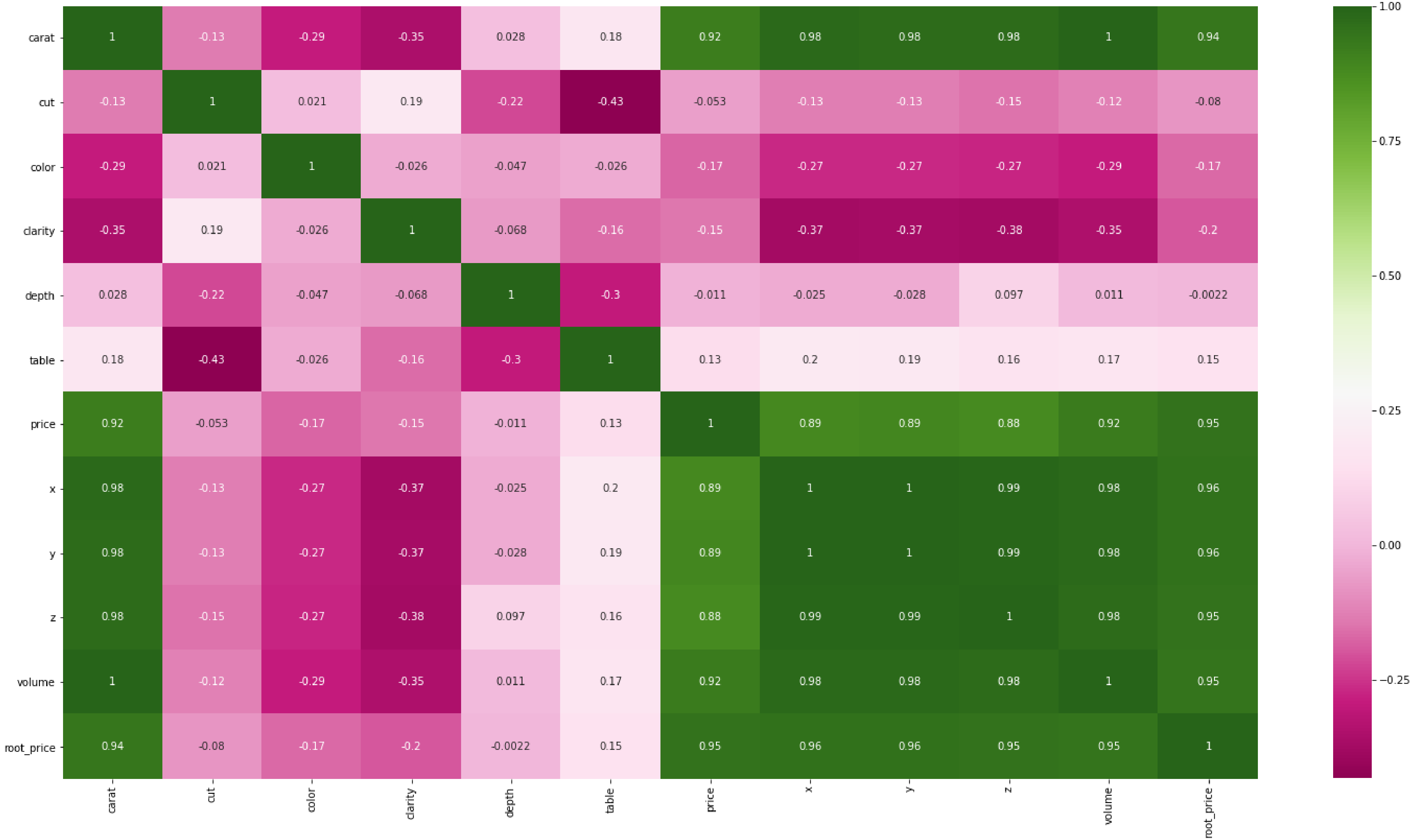
Feature Engineering

- df['root_price'] = np.cbrt(df['price'])



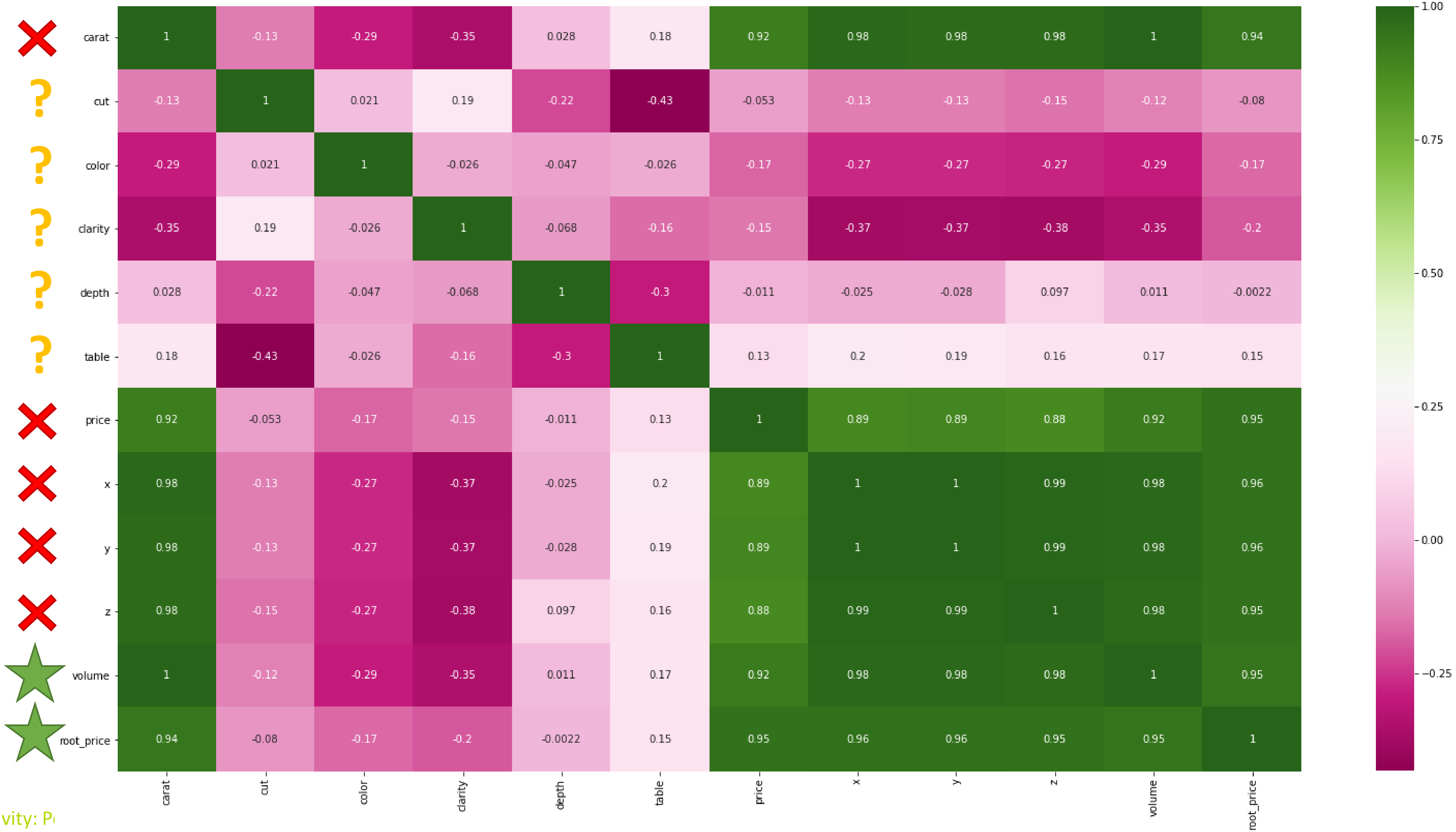


Exploratory Analysis



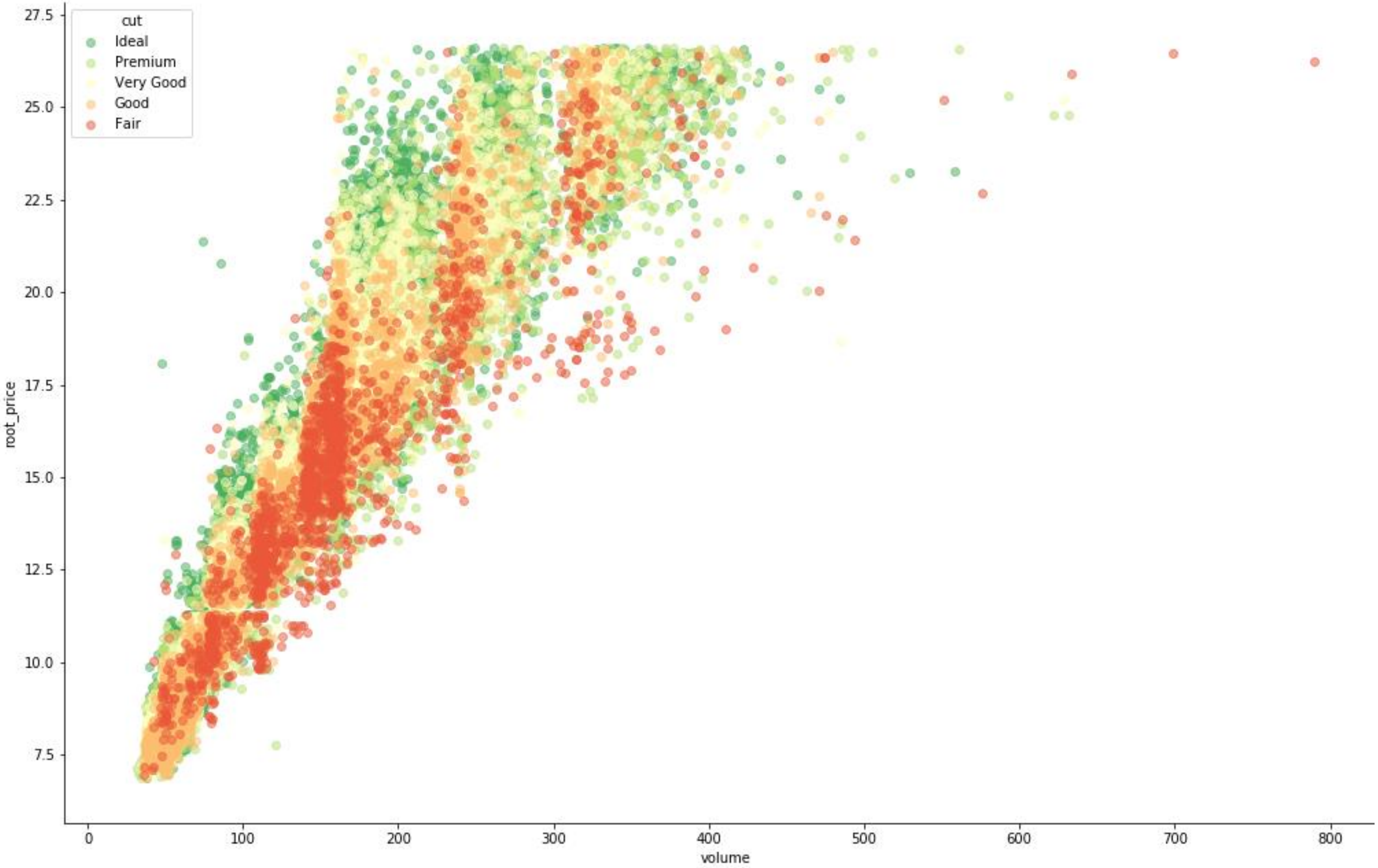
	carat	cut	color	clarity	depth	table	price	x	y	z	volume	root_price
carat	1	-0.13	-0.29	-0.35	0.028	0.18	0.92	0.98	0.98	0.98	1	0.94
cut	-0.13	1	0.021	0.19	-0.22	-0.43	-0.053	-0.13	-0.13	-0.15	-0.12	-0.08
color	-0.29	0.021	1	-0.026	-0.047	-0.026	-0.17	-0.27	-0.27	-0.27	-0.29	-0.17
clarity	-0.35	0.19	-0.026	1	-0.068	-0.16	-0.15	-0.37	-0.37	-0.38	-0.35	-0.2
depth	0.028	-0.22	-0.047	-0.068	1	-0.3	-0.011	-0.025	-0.028	0.097	0.011	-0.0022
table	0.18	-0.43	-0.026	-0.16	-0.3	1	0.13	0.2	0.19	0.16	0.17	0.15
price	0.92	-0.053	-0.17	-0.15	-0.011	0.13	1	0.89	0.89	0.88	0.92	0.95
x	0.98	-0.13	-0.27	-0.37	-0.025	0.2	0.89	1	1	0.99	0.98	0.96
y	0.98	-0.13	-0.27	-0.37	-0.028	0.19	0.89	1	1	0.99	0.98	0.96
z	0.98	-0.15	-0.27	-0.38	0.097	0.16	0.88	0.99	0.99	1	0.98	0.95
volume	1	-0.12	-0.29	-0.35	0.011	0.17	0.92	0.98	0.98	0.98	1	0.95
root_price	0.94	-0.08	-0.17	-0.2	-0.0022	0.15	0.95	0.96	0.96	0.95	0.95	1

Exploratory Analysis



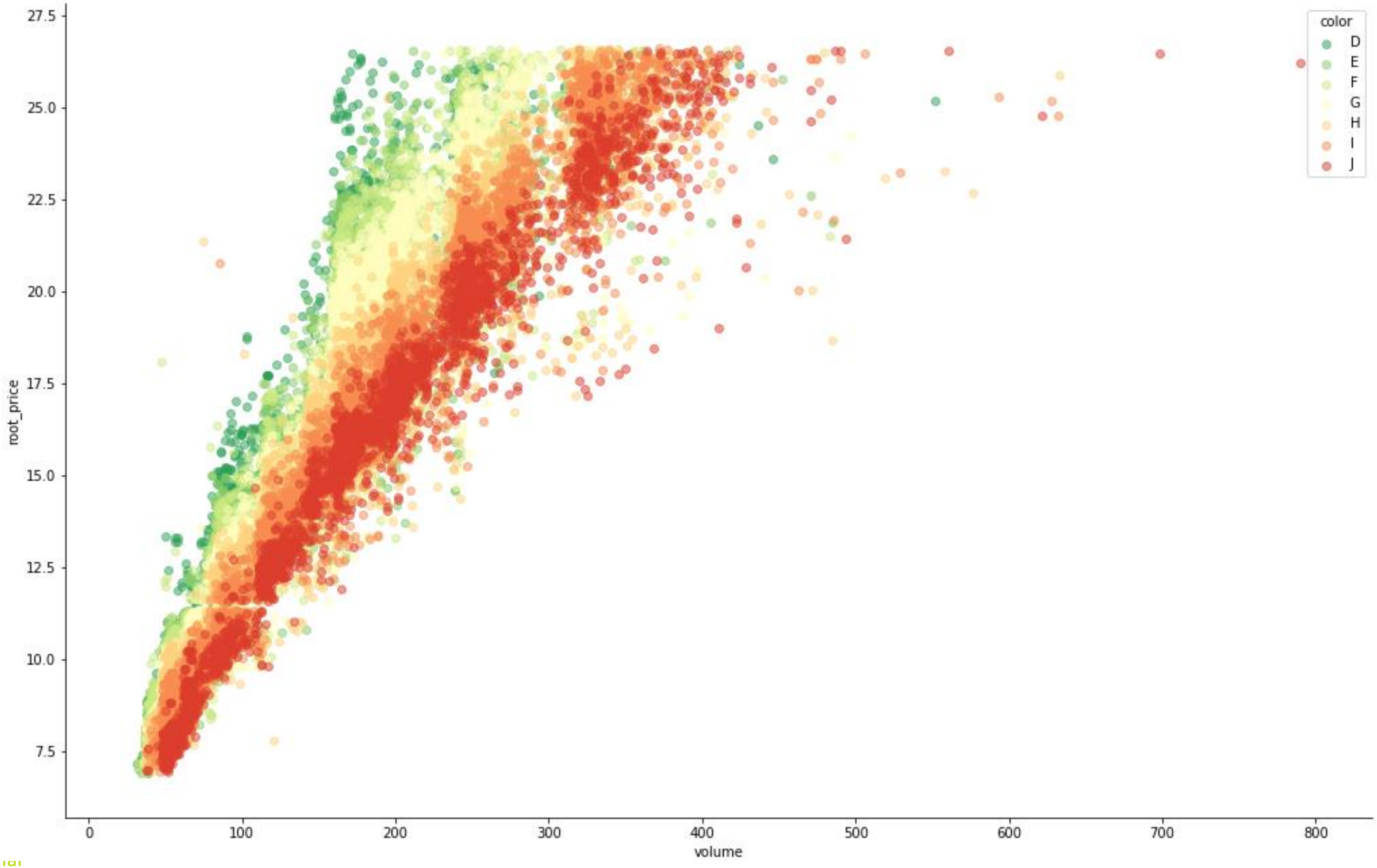
		carat	cut	color	clarity	depth	table	price	x	y	z	volume	root_price
✗	carat	1	-0.13	-0.29	-0.35	0.028	0.18	0.92	0.98	0.98	0.98	1	0.94
?	cut	-0.13	1	0.021	0.19	-0.22	-0.43	-0.053	-0.13	-0.13	-0.15	-0.12	-0.08
?	color	-0.29	0.021	1	-0.026	-0.047	-0.026	-0.17	-0.27	-0.27	-0.27	-0.29	-0.17
?	clarity	-0.35	0.19	-0.026	1	-0.068	-0.16	-0.15	-0.37	-0.37	-0.38	-0.35	-0.2
?	depth	0.028	-0.22	-0.047	-0.068	1	-0.3	-0.011	-0.025	-0.028	0.097	0.011	-0.0022
?	table	0.18	-0.43	-0.026	-0.16	-0.3	1	0.13	0.2	0.19	0.16	0.17	0.15
✗	price	0.92	-0.053	-0.17	-0.15	-0.011	0.13	1	0.89	0.89	0.88	0.92	0.95
✗	x	0.98	-0.13	-0.27	-0.37	-0.025	0.2	0.89	1	1	0.99	0.98	0.96
✗	y	0.98	-0.13	-0.27	-0.37	-0.028	0.19	0.89	1	1	0.99	0.98	0.96
✗	z	0.98	-0.15	-0.27	-0.38	0.097	0.16	0.88	0.99	0.99	1	0.98	0.95
★	volume	1	-0.12	-0.29	-0.35	0.011	0.17	0.92	0.98	0.98	0.98	1	0.95
★	root_price	0.94	-0.08	-0.17	-0.2	-0.0022	0.15	0.95	0.96	0.96	0.95	0.95	1

Exploratory Analysis - Cut

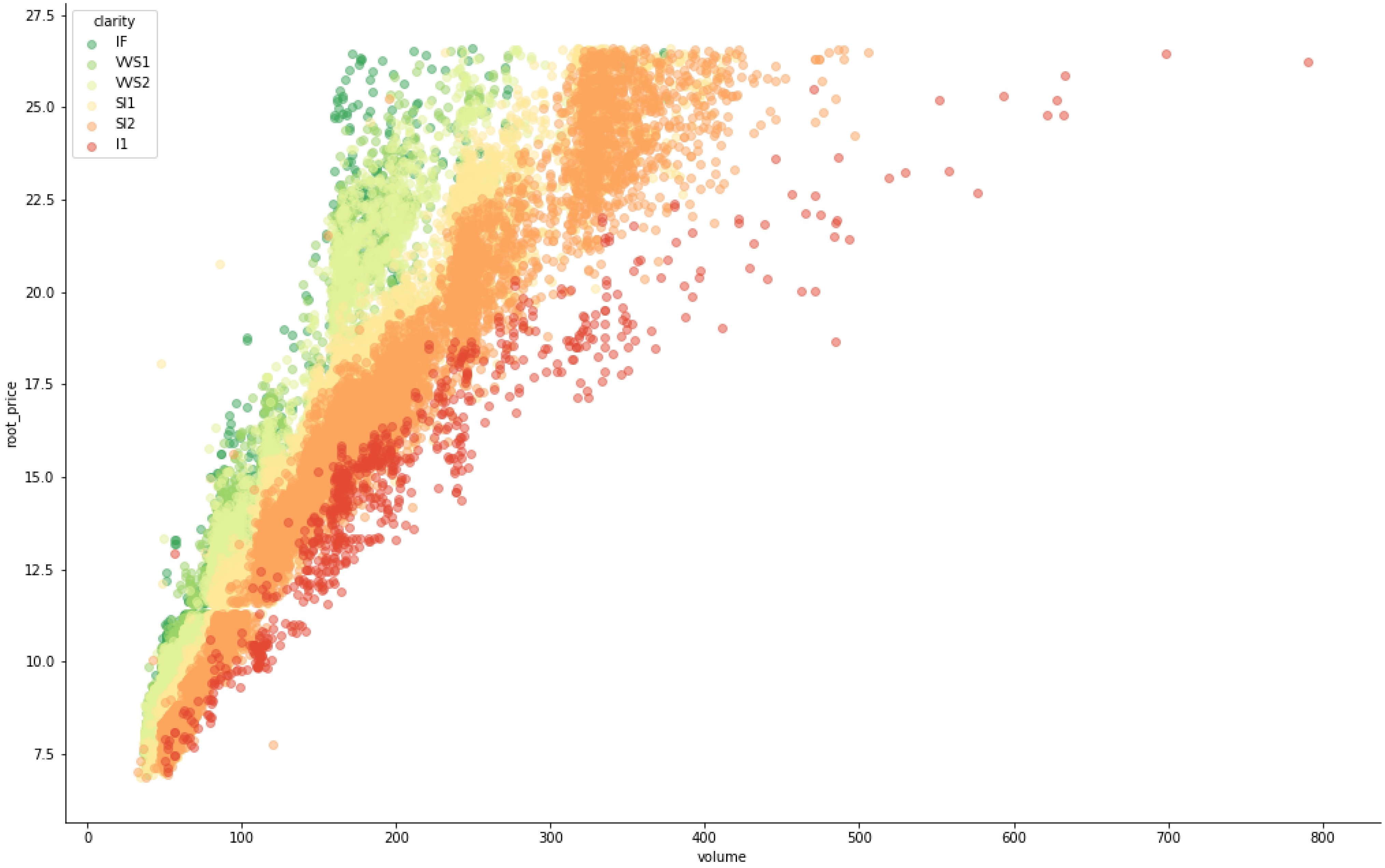


Sensitivity: Personal

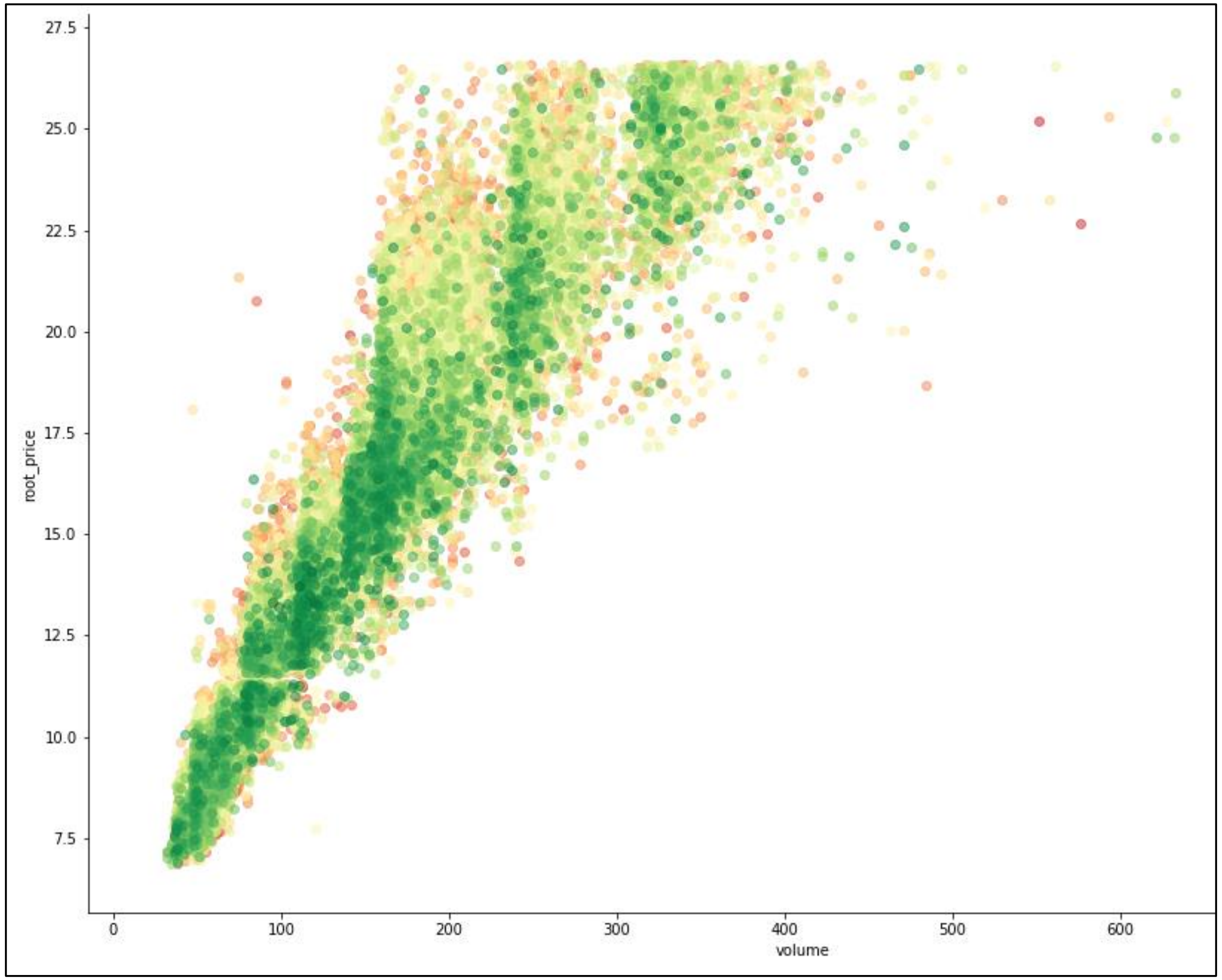
Exploratory Analysis - Color

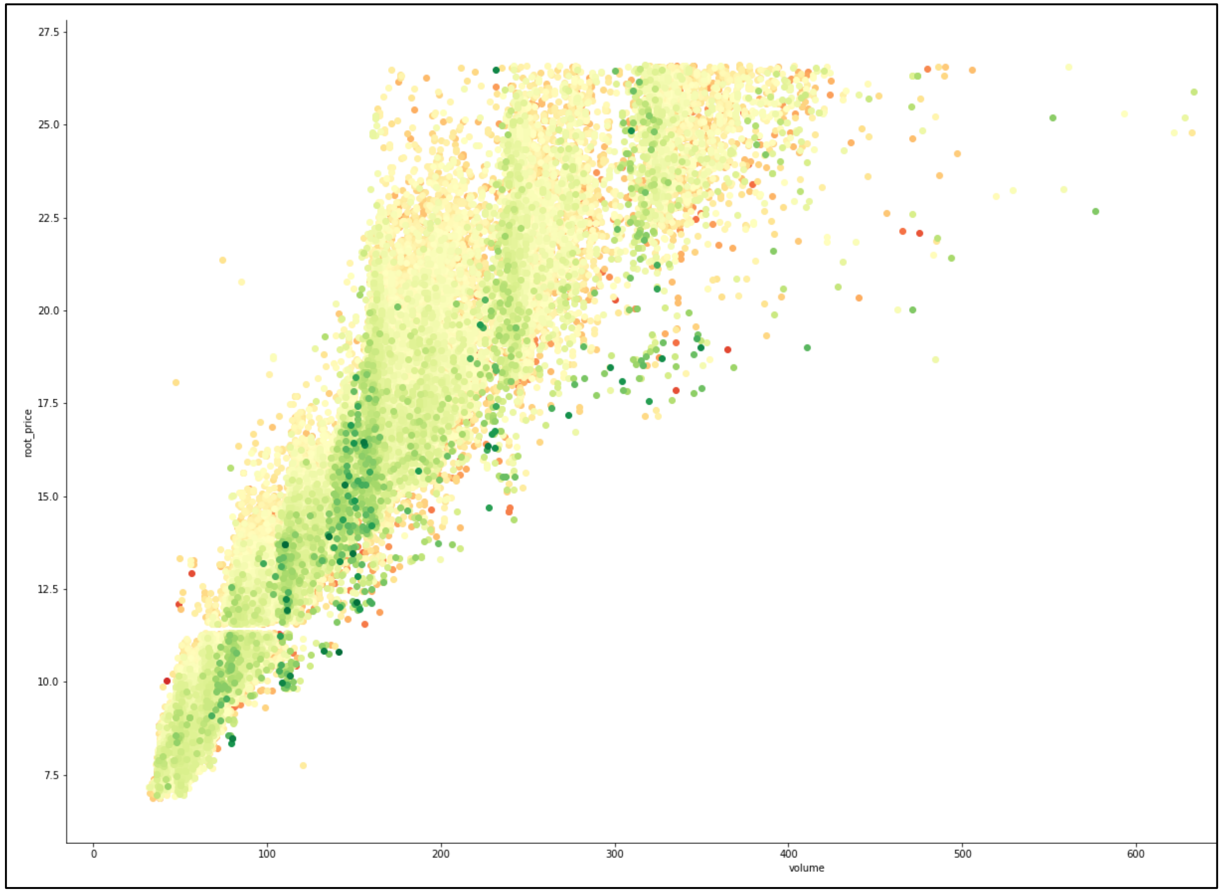


Exploratory Analysis - Clarity



Exploratory Analysis - Table & Depth





Exploratory Analysis - Table & Depth

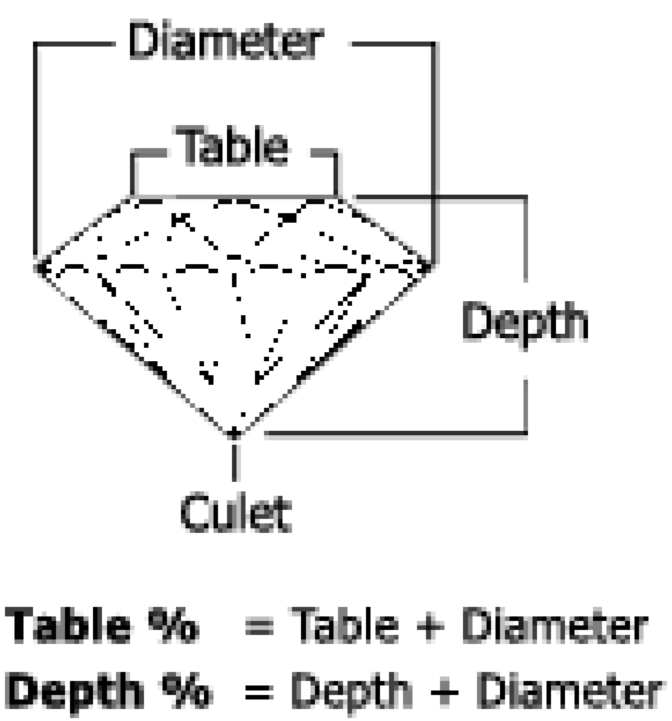
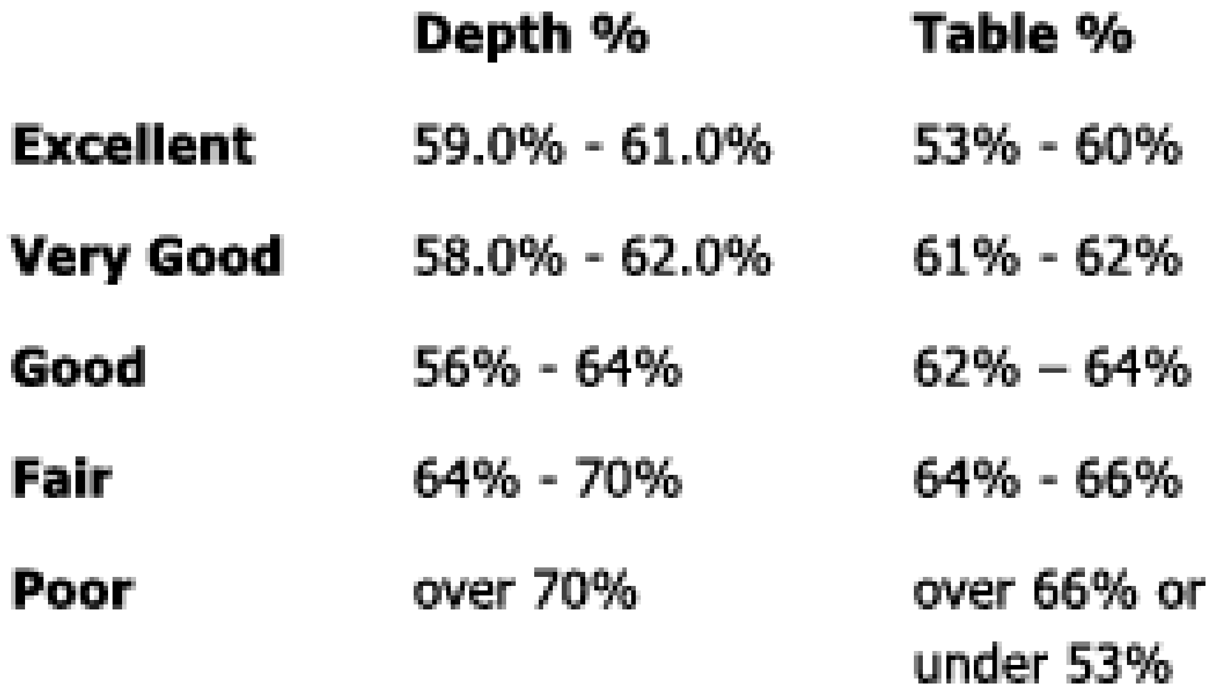


Table % = Table + Diameter
Depth % = Depth + Diameter



	Depth %	**Table %**
Excellent	59.0% - 61.0%	53% - 60%
Very Good	58.0% - 62.0%	61% - 62%
Good	56% - 64%	62% – 64%
Fair	64% - 70%	64% - 66%
Poor	over 70%	over 66% or under 53%

Source: https://www.torresjewelco.com.au/diamonds/education/depth-table-percentage.html

Label Encoding

- df['cut']

{'Fair': 1, 'Good': 2, 'Very Good': 3, 'Premium': 4, 'Ideal':5}

- df['color']

{'J': 1, 'I': 2, 'H': 3, 'G': 4, 'F':5, 'E':6, 'D':7}

- df['clarity']

{'I3': 1, 'I2': 2, 'I1': 3, 'SI2': 4, 'SI1':5,

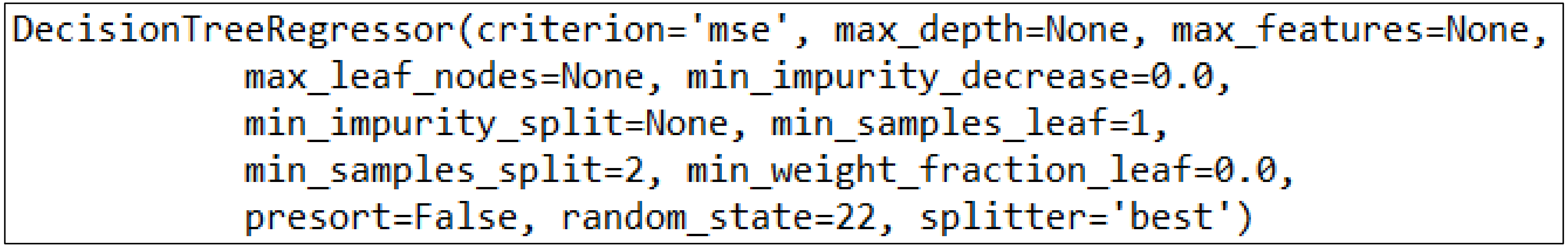
'VS2':6, 'VS1':7, 'VVS2': 8, 'VVS1':9, 'IF':10, 'FL':11}

Train/Validation/Test Split

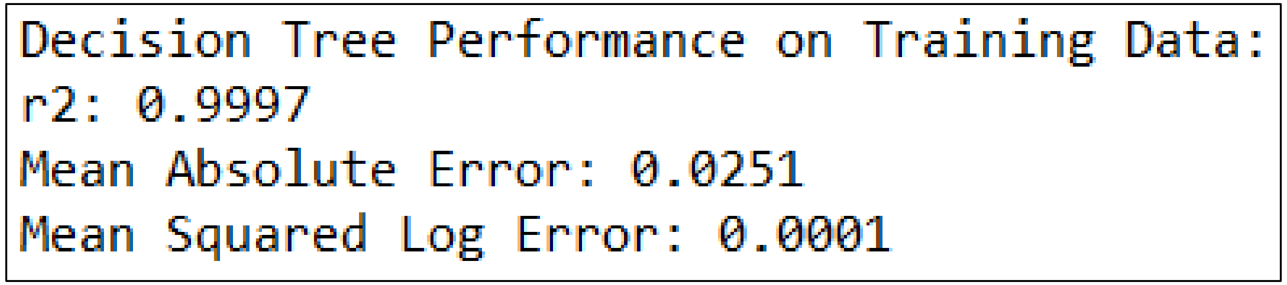
- Separation
  - Train: %50
  - Validation: %25
  - Test: %25
- Strafitication
  - 100 bins on the target using np. digitize, as root_price is continuous
- Shuffling
  - Dataset is ordered on price when imported

Model Selection – Decision Tree

- Default Decision Tree regressor on training data



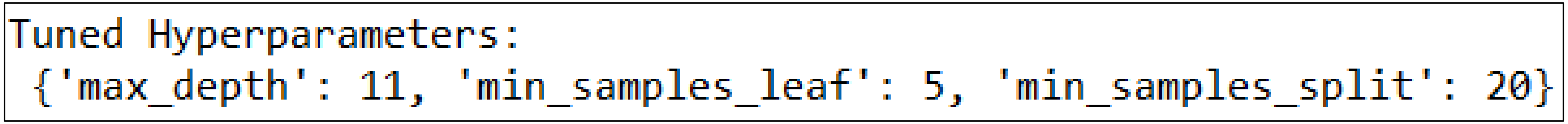
```
DecisionTreeRegressor(criterion='mse', max_depth=None, max_features=None,
           max_leaf_nodes=None, min_impurity_decrease=0.0,
           min_impurity_split=None, min_samples_leaf=1,
           min_samples_split=2, min_weight_fraction_leaf=0.0,
           presort=False, random_state=22, splitter='best')
```



```
Decision Tree Performance on Training Data:
r2: 0.9997
Mean Absolute Error: 0.0251
Mean Squared Log Error: 0.0001
```

Decision Tree Hyperparameter Tuning

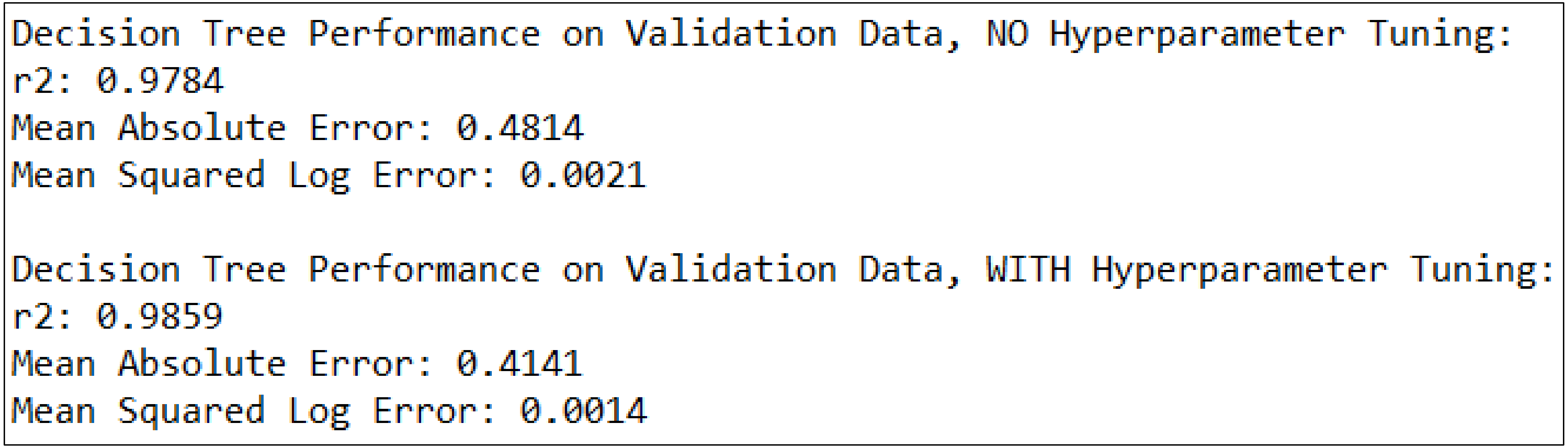
- GridSearchCV is used
- grid={"max_depth":[1,2,3,4,5,6,7,8,9,10,11,12]
  "min_samples_split":[10,20,50,100],
  "min_samples_leaf":[1,5,10,20,50,100]}



```
Tuned Hyperparameters:
 {'max_depth': 11, 'min_samples_leaf': 5, 'min_samples_split': 20}
```

Model Selection – Decision Tree

- Decision Tree regressor on validation data, w/ or w/o tuning



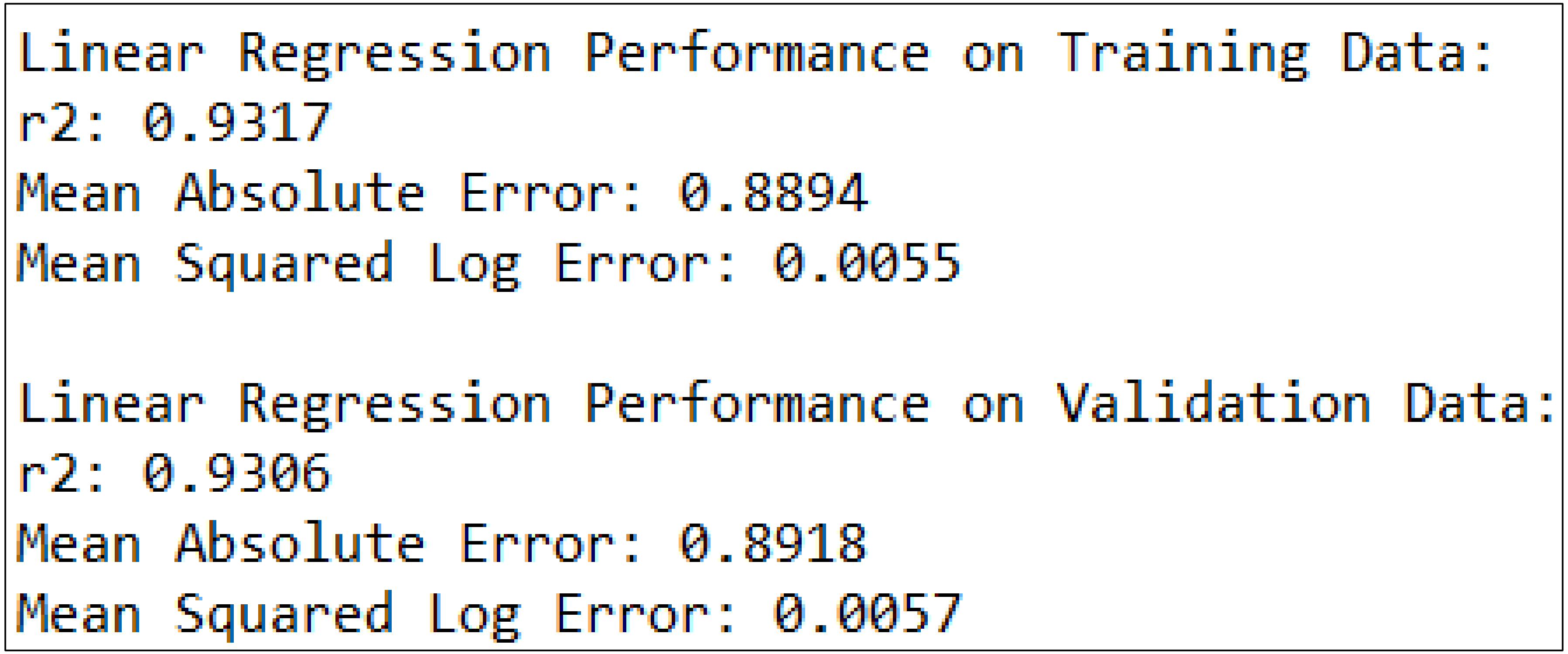
```
Decision Tree Performance on Validation Data, NO Hyperparameter Tuning:
r2: 0.9784
Mean Absolute Error: 0.4814
Mean Squared Log Error: 0.0021

Decision Tree Performance on Validation Data, WITH Hyperparameter Tuning:
r2: 0.9859
Mean Absolute Error: 0.4141
Mean Squared Log Error: 0.0014
```

Model Selection – Linear Regression

- Linear Regression with the following (rounded) formula:

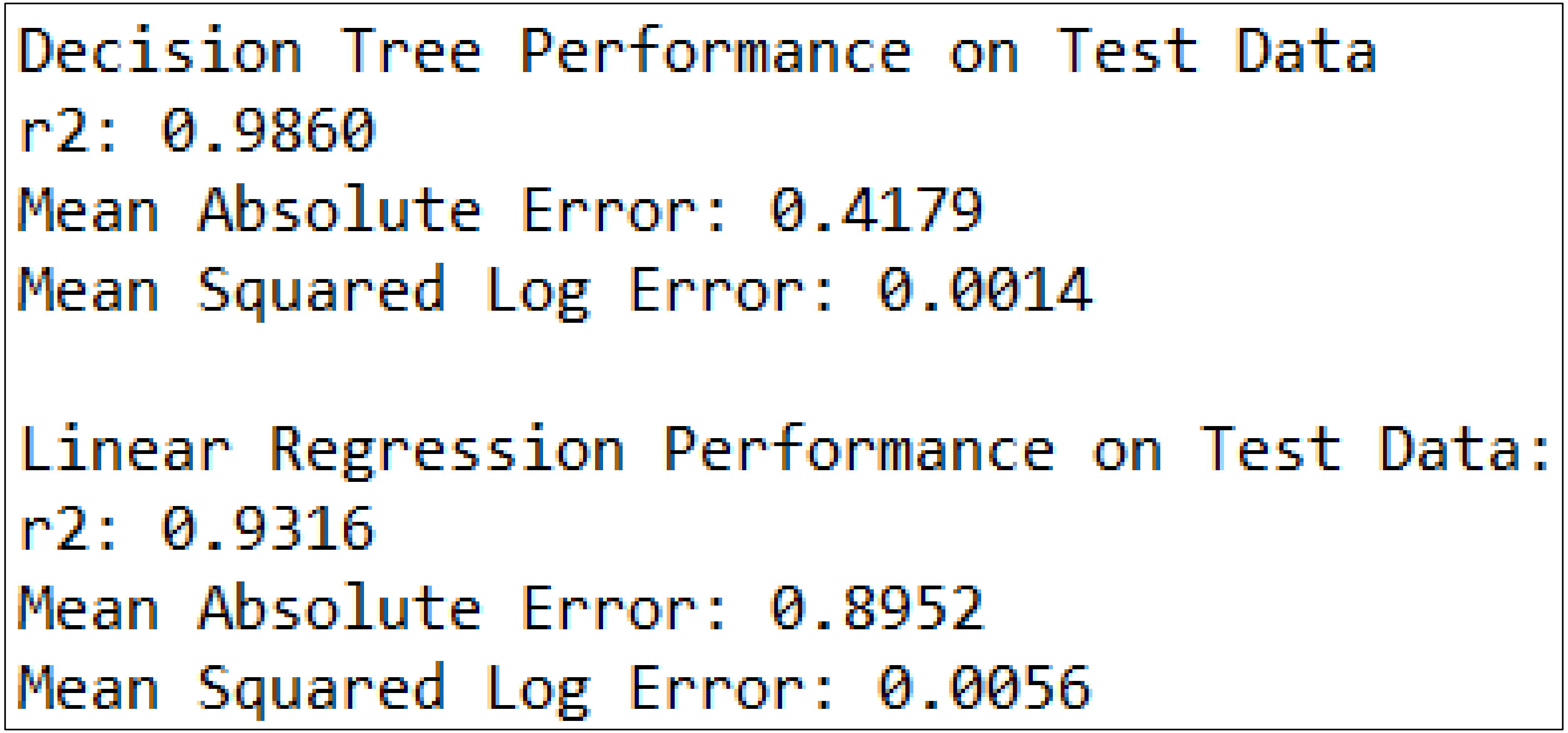
Price = -4.46 + (Volume*0.07) +(Clarity*0.5)+(Color*0.4)+(Cut*0.09)+(Table*0.03)+(Depth*0.05)



```
Linear Regression Performance on Training Data:
r2: 0.9317
Mean Absolute Error: 0.8894
Mean Squared Log Error: 0.0055

Linear Regression Performance on Validation Data:
r2: 0.9306
Mean Absolute Error: 0.8918
Mean Squared Log Error: 0.0057
```

Model Selection – Final Performance



```
Decision Tree Performance on Test Data
r2: 0.9860
Mean Absolute Error: 0.4179
Mean Squared Log Error: 0.0014

Linear Regression Performance on Test Data:
r2: 0.9316
Mean Absolute Error: 0.8952
Mean Squared Log Error: 0.0056
```