MMF Data Science 2022 Kaggle Project

Tongfei Zhou

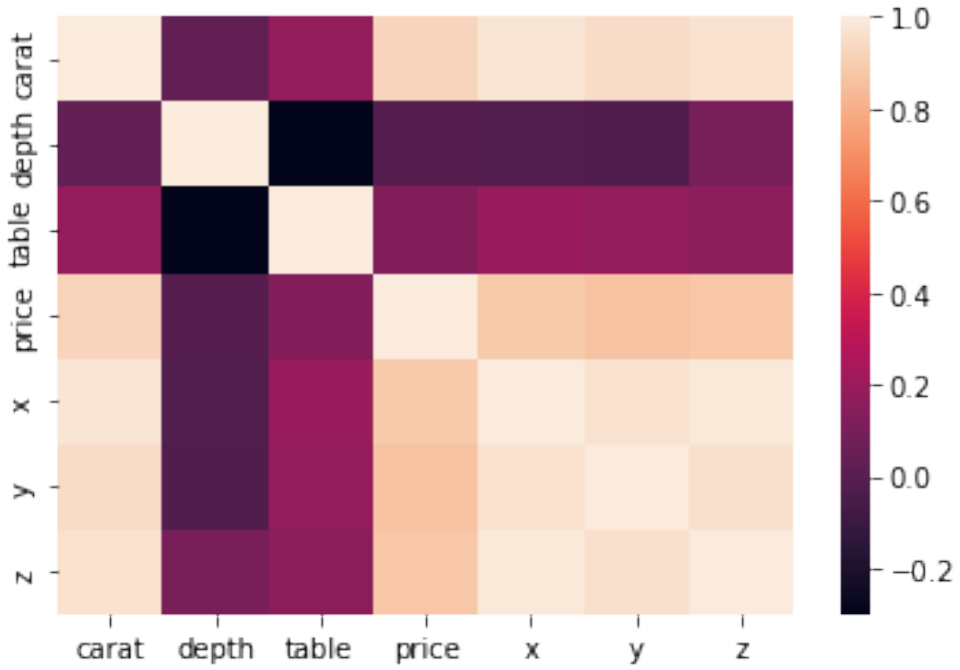
November 16th 2022

1 Introduction and Data Exploration

In this project, the task is to predict the price of the diamond, given features including carat, which is the weight of the diamond, cut quality (Fair, Good, Very Good, Premium and Ideal, from worst to best), color of the diamond, clarity (l1, Sl2, Sl1, VS2, VS1, VVS2, VVS1, IF, from worst to best), total depth percentage, and the xyz of the diamond. After some basic data exploration (no NA being checked), I got the following results:

	carat	depth	table	price	x	y	z
count	43154.000000	43154.000000	43154.000000	43154.000000	43154.000000	43154.000000	43154.000000
mean	0.799047	61.742925	57.459010	3946.777054	5.733798	5.737574	3.539338
std	0.475214	1.428410	2.227191	3998.657385	1.123004	1.150325	0.696203
min	0.200000	43.000000	43.000000	326.000000	0.000000	0.000000	0.000000
25%	0.400000	61.000000	56.000000	953.000000	4.710000	4.720000	2.910000
50%	0.700000	61.800000	57.000000	2406.500000	5.700000	5.710000	3.530000
75%	1.040000	62.500000	59.000000	5367.000000	6.540000	6.540000	4.040000
max	4.500000	79.000000	79.000000	18823.000000	10.230000	58.900000	8.060000

From the above table, we can tell that the price range is quite huge, but the rest features are quite centered since the standard deviation is quite small. I also explored the heat map so that I could tell the correlation among features. The heat map for the numerical features looks like below:



This heat map shows that carat, x, y and z are features showing high correlation with the price. Also, one can notice that the negative correlation here is quite small, and so I did not take particular considerations in handling the table and depth features. The reason why I did not do anything to the categorical features will be explained in the **Feature Engineering and Model Training** part.

2 Feature Engineering and Model Training

With the above heat map, I did a feature engineering by integrating the x, y and z. Since these three represents the length, width and depth of the diamond, I multiplied them together to get a rough volume of the diamond. I chose to fit this data with an AutoML package published by Amazon. The pacakge is called Autogluon, and it went famous because it got the first place in the Kaggle Titanic Competition with 3 lines of code. The package will automatically encode all of the categorical features, and that's why I did not do anything to the categorical features in the first place. The training code has only 3 lines here as well, as below:

```
label = "price"
from autogluon.tabular import TabularPredictor
predictor = TabularPredictor(label=label, eval_metric = 'root_mean_squared_error').fit(train_data=train_data_modified,
                                                                                       presets = 'best_quality')
```

The `train_data_modified` is the training data after feature engineering described above. `presets` is set to `best_quality` so that the AutoML will do the bagging and stacking over all the possible models, such as LightGBM, DNN, XGBoost and Random Forest. These three lines produces the score of 500 and got the first place by the end of this report being written.