Machine Learning – Polynomial Linear Regression Assignment

Name: Nour Mohamed Hussein Mahmoud Mohamed Kamaly

ID: 20191700701

Department: Scientific Computing

Section: 5

Tried experiments:

Complexity (degree)	Test Error
4	9169801.024104655
5	9152319.710070144
6	9148367.375879873
7	9149342.347083537
8	9148269.463403001
9	9148716.562393721
10	9149796.663274197
11	9149231.627457887

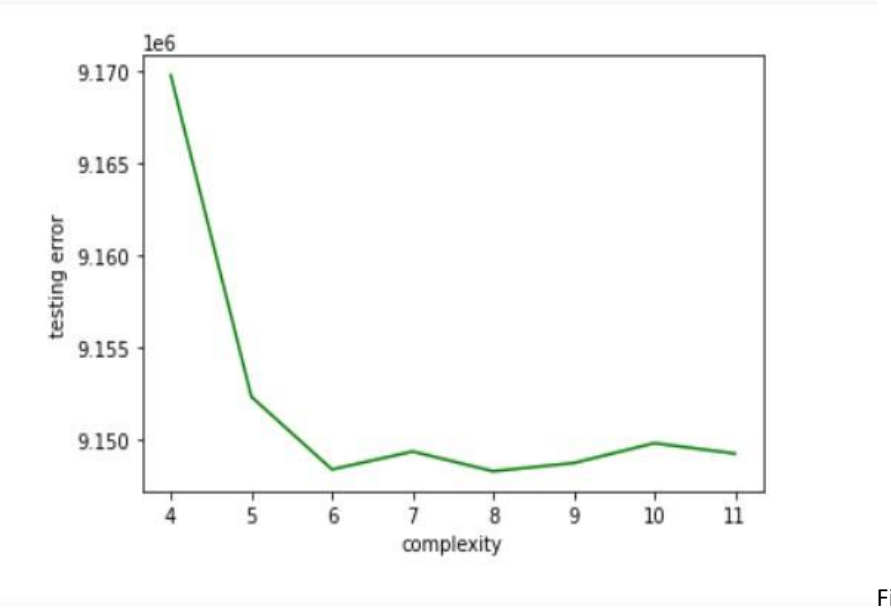

Figure 1

Features chosen:

1. In year and car maker features, there are plenty of classes that have low frequencies in the dataset (getting the percentage of the count relative to the dataset rows), and the most important classes are the ones that appear a lot in the observations, so by getting the categories in each one of those features, I saw the value count for every class, kept the high frequencies classes and made a class called "Other" for the other categories. This is visualized by seaborn library (to measure the count).

2. Kilometers feature is normalized as its very large values may lead to undesirable effects when fitting the model.

- All features were chosen, and none were removed as I used Lasso regression as it already selects the features by making their weights almost zero, making very little contribution to the model prediction, also as a regulizer, as the dataset is actually 37 unique rows repeated almost 15 times each.

Parameters chosen:

Random state in:

1. dataset = dataset.sample(frac=1,random_state = 26)
   As the data is repeated, it is a good approach to shuffle the data as not to be biased to a certain observation if they are found after one another in the dataset. Choosing a random state allows me to reach the same outputs each run.

2. x_train,x_test,y_train,y_test = train_test_split(x,y,test_size = 0.2,random_state = 404)
   Choosing a random state to ensure that I have the same test set each fresh run.

- I have used google colab's GPU for training as the larger complexities took some time on my local machine's CPU, so colab notebook can be found here.