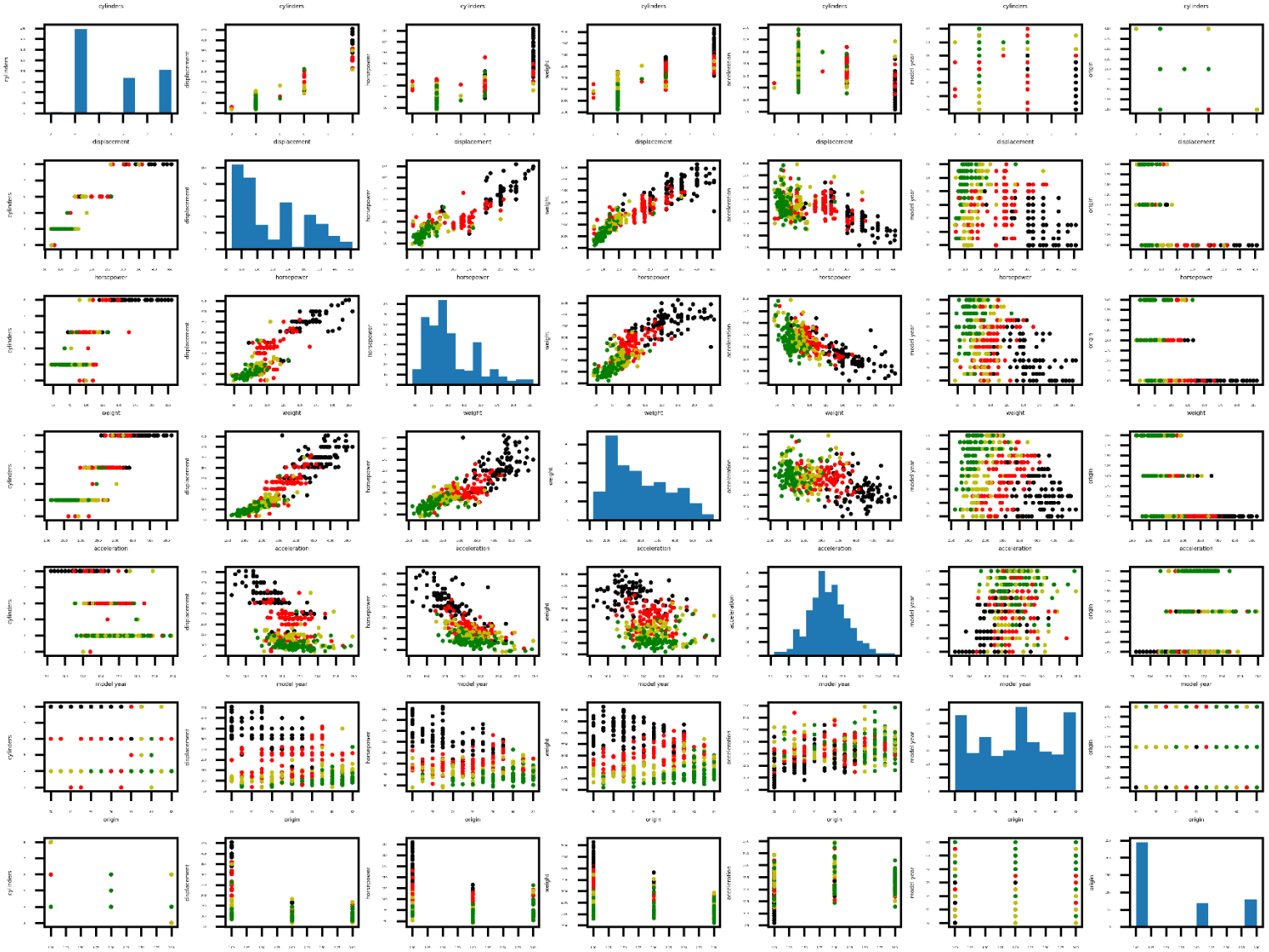
ECS 171 HW1

Wenhao Su 915297212

1. Run prob1.py and get the following results:
   LOW MPG Threshold is between 9.0 and 17.0
   MEDIUM MPG Threshold is between 17.0 and 23.0
   HIGH MPG Threshold is between 23.0 and 29.0
   VERY HIGH MPG Threshold is between 29.0 and 46.6

2. Run prob2.py and the following plot will be saved to the current working directory with name "ScatterPlotMatrix.png". I think the most informative pair-wise feature would be (weight, horsepower) for the simple reason that the distribution of plots with these two features fit into more distinctive clusters in terms of MPG which is indicated by colors of dots. Moreover, the main distribution of plots also fits into a nice regression line with direct proportion relationship. Apparently, more horsepower and weight will contribute to the result of less MPG.

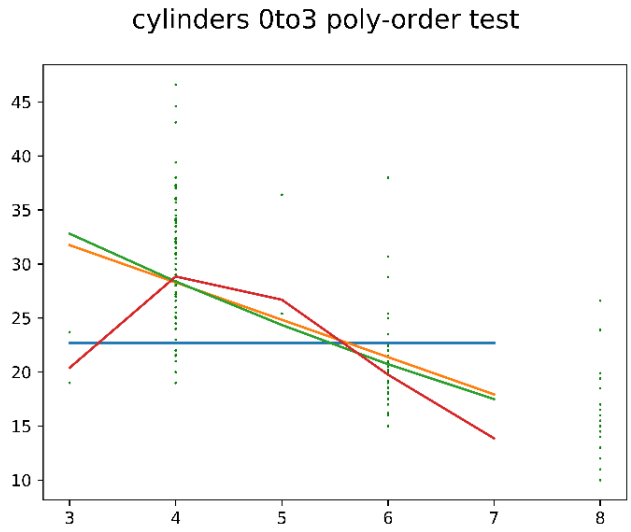


3. Linear regression solver for a single feature is the first function in prob3_utility_funcs.py with the following function signature:

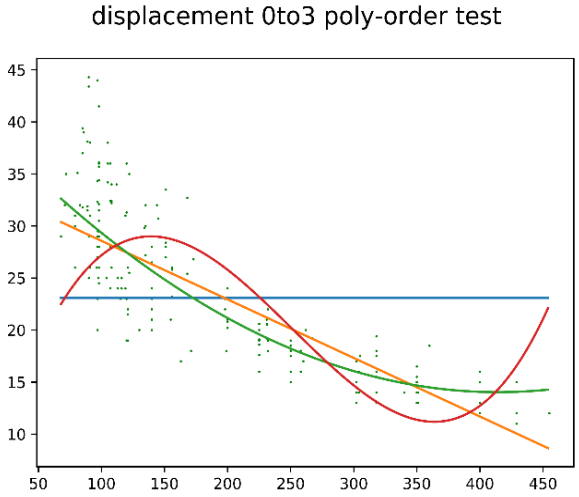
```
def single_feature_regression(feature_vec, MPG_vec, polyBasis)
```

4. Run prob4.py and get the following printed results (7 plots will be saved inside folder):

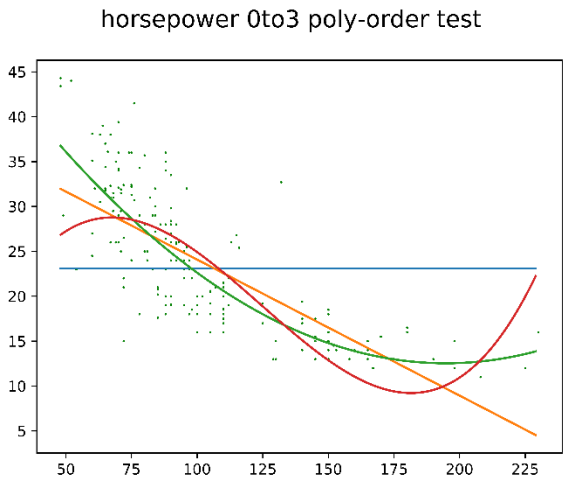
| Cylinder | | |
|---|---|---|
| order | training | testing |
| 0 | 60.45 | 62.20 |
| 1 | 22.21 | 26.09 |
| 2 | 22.11 | 25.89 |
| 3 | 20.27 | 23.63 |



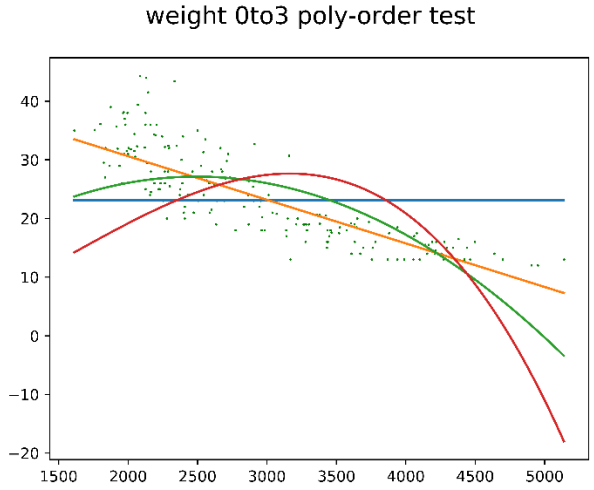

| Displacement | | |
|---|---|---|
| order | training | testing |
| 0 | 60.45 | 62.20 |
| 1 | 19.07 | 23.88 |
| 2 | 16.80 | 21.27 |
| 3 | 30.15 | 36.65 |



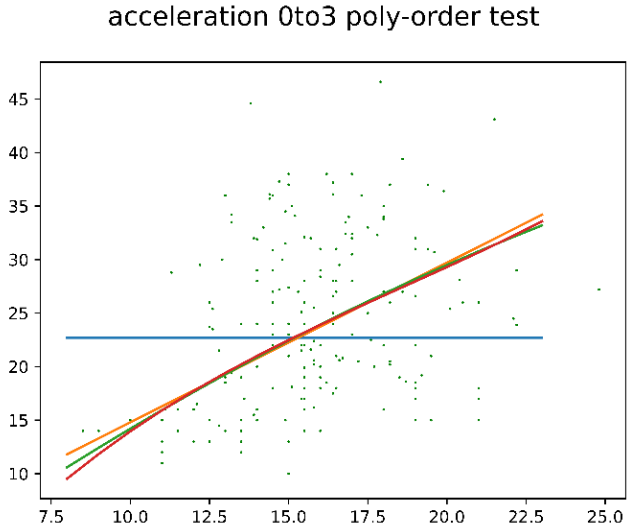
| Horsepower | | |
|---|---|---|
| order | training | testing |
| 0 | 60.45 | 62.20 |
| 1 | 21.24 | 26.83 |
| 2 | 16.03 | 22.14 |
| 3 | 26.43 | 29.13 |



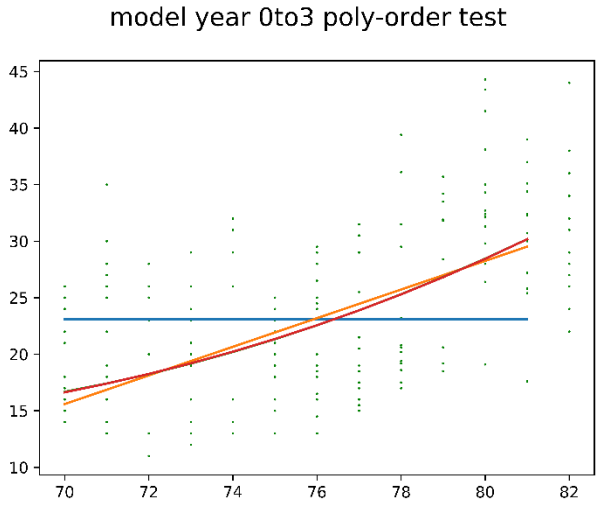

| Weight | | |
|---|---|---|
| order | training | testing |
| 0 | 60.45 | 62.20 |
| 1 | 17.62 | 19.97 |
| 2 | 35.89 | 37.87 |
| 3 | 89.18 | 92.66 |



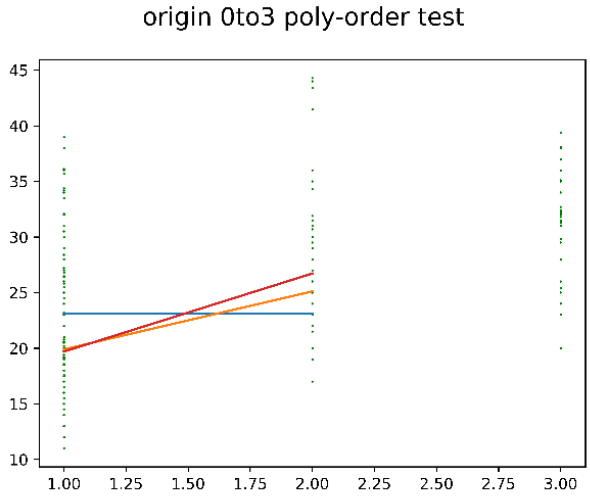

| Acceleration | | |
|---|---|---|
| order | training | testing |
| 0 | 60.45 | 62.20 |
| 1 | 43.09 | 58.59 |
| 2 | 43.01 | 57.72 |
| 3 | 42.98 | 57.56 |



| Model Year | | |
|---|---|---|
| order | training | testing |
| 0 | 60.45 | 62.20 |
| 1 | 39.52 | 41.38 |
| 2 | 37.80 | 39.45 |
| 3 | 37.78 | 39.54 |




| Origin | | |
|---|---|---|
| order | training | testing |
| 0 | 60.45 | 62.20 |
| 1 | 41.63 | 41.59 |
| 2 | 39.70 | 42.67 |
| 3 | 39.70 | 42.67 |




Overall, I believe that 2rd polynomial's performance is the best. It is both generic while still reflecting the overall feature of this data distribution.
I think horsepower is the most informative feature for the simple reason that data distribution is more condense and less spread out while fitting well into the 2rd polynomial regression.

5. Run prob5.py and get the following printed results for MSE:

| Multi-Feature | | |
|---|---|---|
| order | training | testing |
| 0 | 60.45 | 62.20 |
| 1 | 10.76 | 11.53 |
| 2 | 7.76 | 9.33 |

6. Run prob6.py and get the following printed results:
   (0 for low, 1 for middle, 2 for high, 3 for very high)
   training: mean precision is 0.79

| | precision |
|---|---|
| 0 | 0.91 |
| 1 | 0.76 |
| 2 | 0.68 |
| 3 | 0.81 |

   testing: mean is 0.7575

| | precision |
|---|---|
| 0 | 0.97 |
| 1 | 0.67 |
| 2 | 0.57 |
| 3 | 0.82 |

7. Run prob7.py and get the following printed results:
   (0 for low, 1 for middle, 2 for high, 3 for very high)
   training precision is 0.775

| | precision |
|---|---|
| 0 | 0.82 |
| 1 | 0.80 |
| 2 | 0.66 |
| 3 | 0.82 |

   testing precision is 0.72

| | precision |
|---|---|
| 0 | 0.94 |
| 1 | 0.67 |
| 2 | 0.49 |
| 3 | 0.78 |

8. Run prob8.py and get the following printed results:
   20.844236780393395
   The label of this car is: [1]
   ([1] is the label for middle MPG)