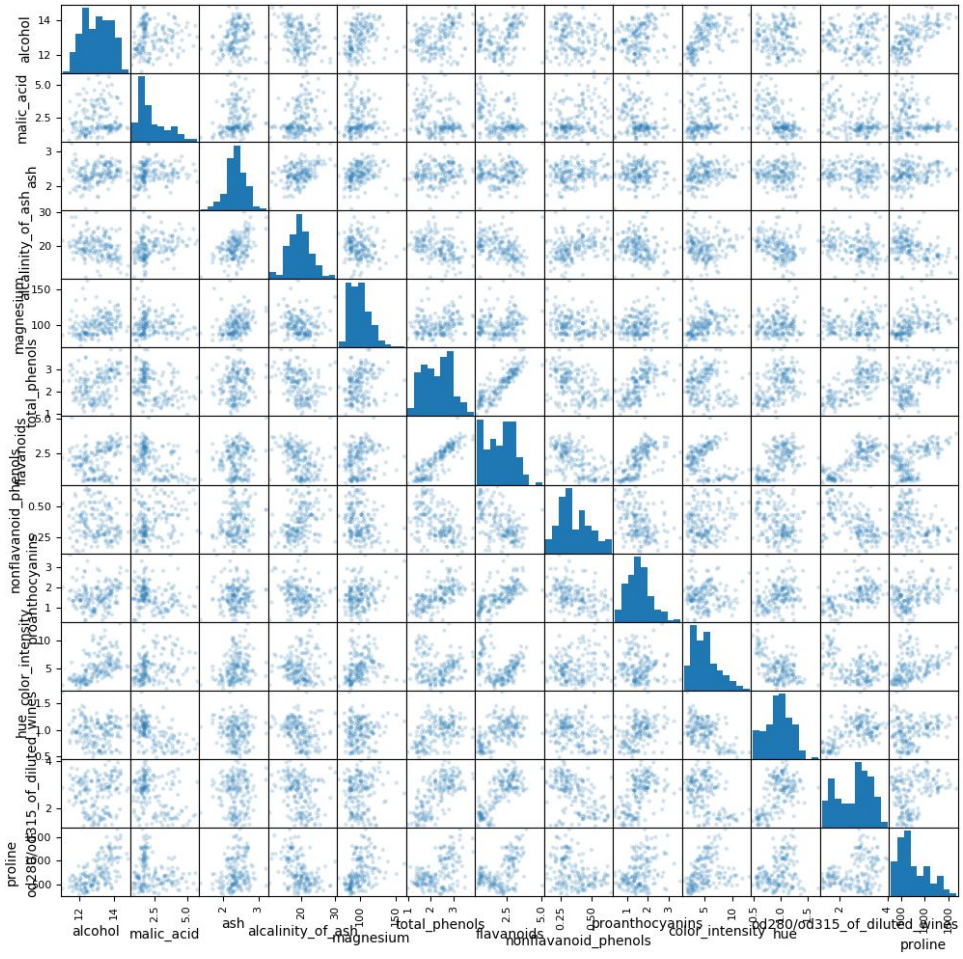
CS 455 - Artificial Intelligence
Homework 3 - KNN and Regression Models
Cameron Stark & Dustin Cribbs

Problem 1:

- Loaded in the wine dataset and pulled out the data points and the targets
- Found the shape of the data and the described the max, min and mean values for each column (using the pandas describe features)
- Plotted the data on a scatter matrix to determine if there was any correlation

(178, 13)

	alcohol	malic_acid	ash	alcalinity_of_ash	magnesium	total_phenols	flavanoids	nonflavanoid_phenols	proanthocyanins	color_intensity	hue	od280/od315_of_diluted_wines	proline
max	14.830000	5.800000	3.230000	30.000000	162.000000	3.880000	5.08000	0.660000	3.580000	13.00000	1.710000	4.000000	1680.000000
min	11.030000	0.740000	1.360000	10.600000	70.000000	0.980000	0.34000	0.130000	0.410000	1.28000	0.480000	1.270000	278.000000
mean	13.000618	2.336348	2.366517	19.494944	99.741573	2.295112	2.02927	0.361854	1.590899	5.05809	0.957449	2.611685	746.893258



K-Nearest Neighbors:

- Created the Classifier with 10 neighbors
- Fit the dataset to a scaler and used the fitted data to predict using the classifier
- The results were printed on a confusion matrix and the Accuracy, Precision and values were displayed

```
KNeighbors
Confusion Matrix:
[[59  0  0]
 [ 6 63  2]
 [ 0  0 48]]
Accuracy: 0.9550561797752809
Precision: 0.9550561797752809
Recall: 0.9550561797752809
F1: 0.9571677491410622
```

Logistic Regression:

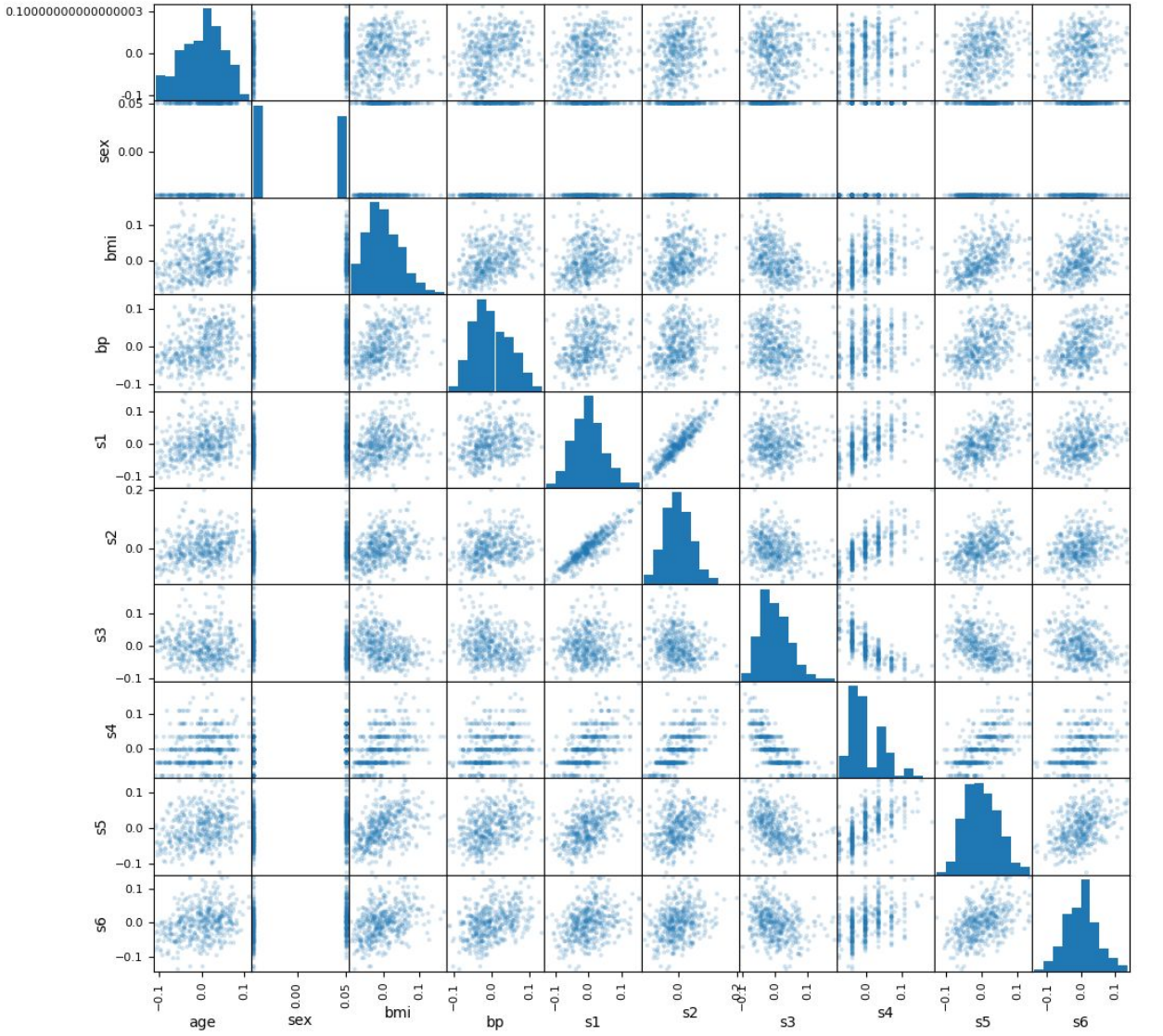
- Transformed the data with a StandardScaler
- Created the Logistic Regression model and fitted it with the data
- Used the Cross Value Predict tool similar to the KNN and then printed the results

```
Logistic Regression
Confusion Matrix:
[[59  0  0]
 [ 0 69  2]
 [ 0  0 48]]
Accuracy: 0.9887640449438202
Precision: 0.9887640449438202
Recall: 0.9887640449438202
F1: 0.9887640449438202
```

Problem 2:

- Loaded the Diabetes dataset and split into the data and targets
- Found shape of data and the max, min and mean of the dataset

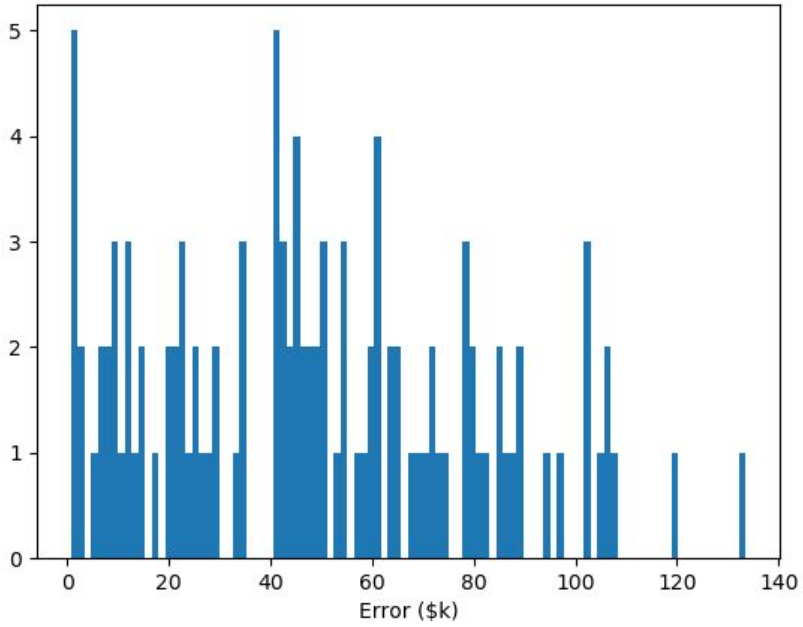
```
(442, 10)
              age           sex           bmi           bp           s1           s2           s3           s4           s5           s6
max   1.107267e-01  5.068012e-02  1.705552e-01  1.320442e-01  1.539137e-01  1.987880e-01  1.811791e-01  1.852344e-01  1.335990e-01  1.356118e-01
min  -1.072256e-01 -4.464164e-02 -9.027530e-02 -1.123996e-01 -1.267807e-01 -1.156131e-01 -1.023071e-01 -7.639450e-02 -1.260974e-01 -1.377672e-01
mean -3.639623e-16  1.309912e-16 -8.013951e-16  1.289818e-16 -9.042540e-17  1.301121e-16 -4.563971e-16  3.863174e-16 -3.848103e-16 -3.398488e-16
```

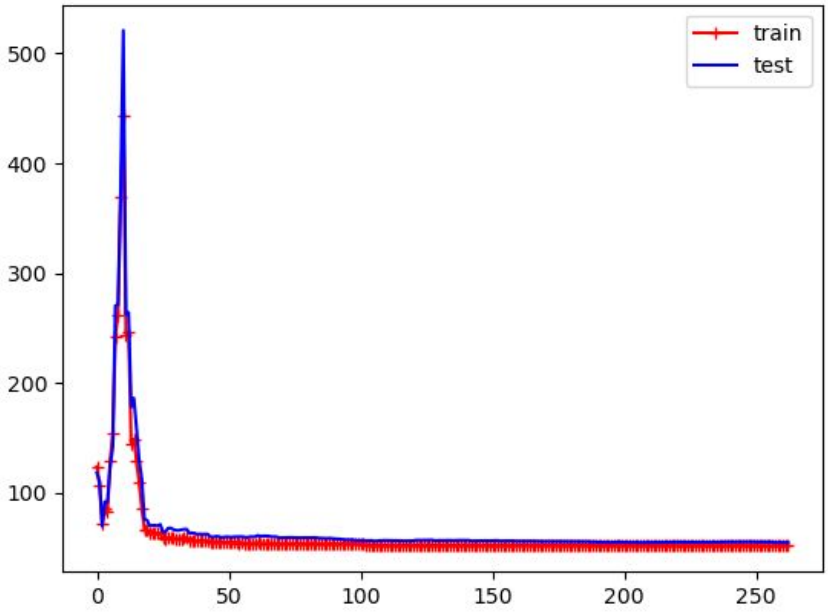


Linear Regression:

- Perform the splitting of the data into training and test data, to fit and plot into the linear regression model
- Used the linear regression model to return a set of predicted data that will be used to plot against the test data

```
Theta:
148.9405112580714 [   34.03974468  -208.02805486   576.66694246   309.89929309
 -1166.52587852   665.73476792   357.34562192   400.66004841
   821.76241919    53.79111382]
MAE = 48.7542443993026
```



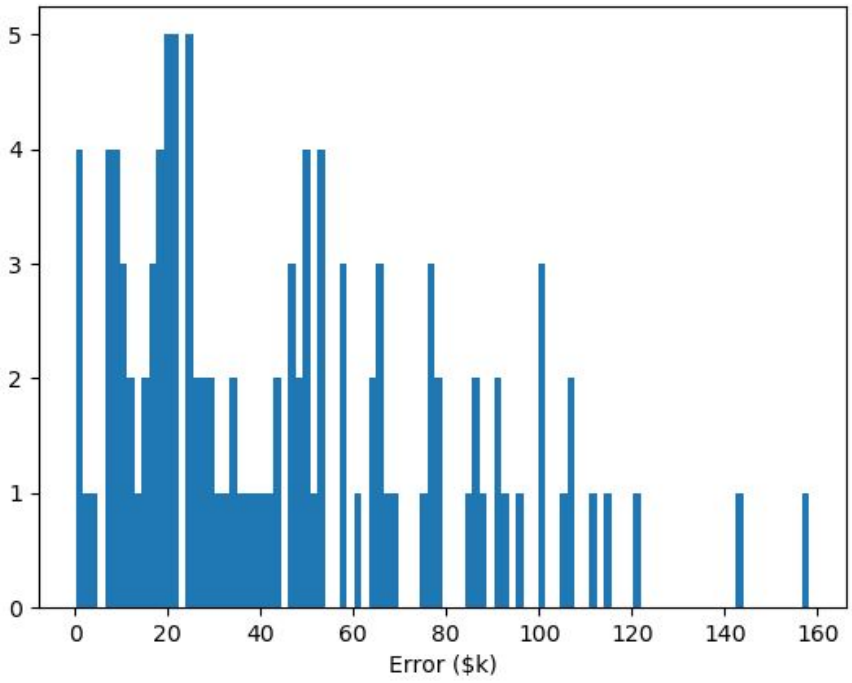


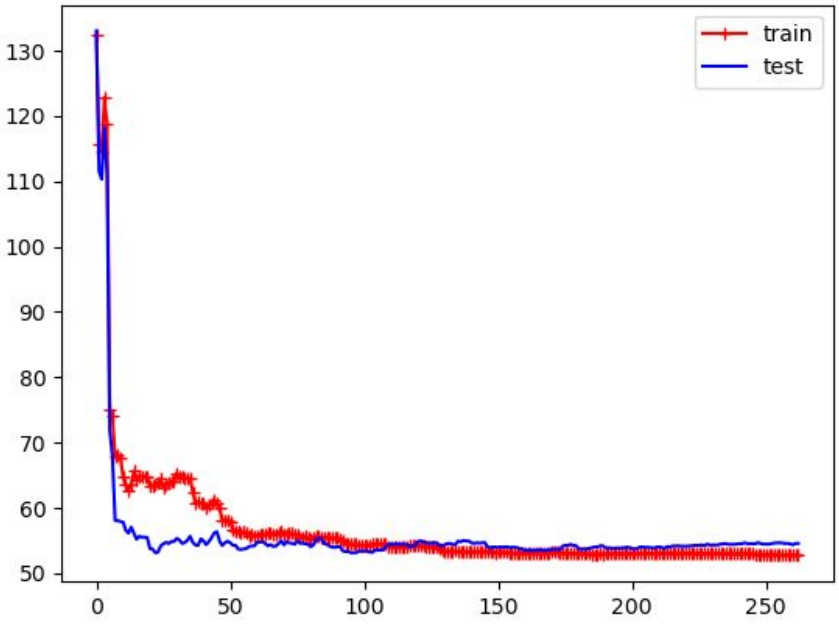
Polynomial Regression:

Linear Regression With Regularization:

- Everything that was done with regular Linear Regression, however with the addition of a regularization function in this case the Ridge function, which is used as the prediction model and is plotted against the test data

```
Theta:
148.6723678274843 [  -3.97155637 -205.20105976  541.43025842  279.54719517 -370.57824061
   18.28930853   18.82265869  262.68067838  509.73449372  148.75210837]
MAE = 45.51136655981206
```


5
4
3
2
1
0
0
20
40
60
80
100
120
140
160
Error ($k)


train
test
130
120
110
100
90
80
70
60
50
0
50
100
150
200
250