

COMP 551: Applied Machine Learning

Assignment 1

Name: Steve Lee

McGill ID: 260568977

Q1 Sampling

1)

Given multinomial distribution, p , we can define the range of the intervals of the categories as the following:

Categories	Movies	COMP-551	Playing	Studying
Probabilities	0.2	0.4	0.1	0.3
Limits of the subInterval	0.2	0.6	0.7	1.0

For example, Category "COMP-551" ranges from 0.2 to 0.6

Then the sampling of the activity from the distribution can be written as below

```
# Pseudocode: Sampling activity from p
activity = ""
s = Random number in range [0,1]
if s < 0.2 then
    activity is "Movie"
else if s >= 0.2 and s < 0.6 then
    activity is "COMP-551"
else if s >= 0.6 and s < 0.7 then
    activity is "Playing"
else if s >= 0.7 then
    activity is "Studying"
end
```

2)

When the activities were sampled for 100 days, the

	100 days (%)	1000 days (%)	Probability (%)
Movies	24.0	18.8	20.0
COMP-551	42.0	43.9	40.0
Playing	8.0	9.4	10.0
Studying	26.0	27.9	30.0

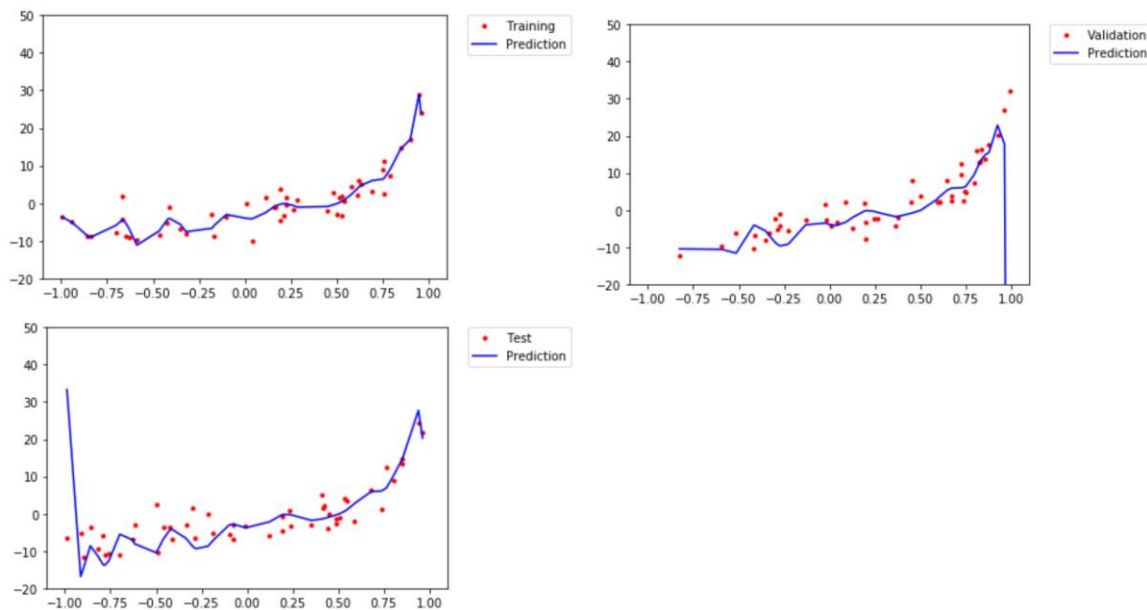
In conclusion, we can observe that the fraction for the activities are getting closer to the true multinomial distribution as the value of the "day" increased (i.e. more data).

Q2 Model Selection

1)

After fitting 20-degree polynomial to the data, following result was obtained:

Training MSE : 6.4747040050552735
Validation MSE : 1417.8987342087942
Test MSE : 50.65363364426098

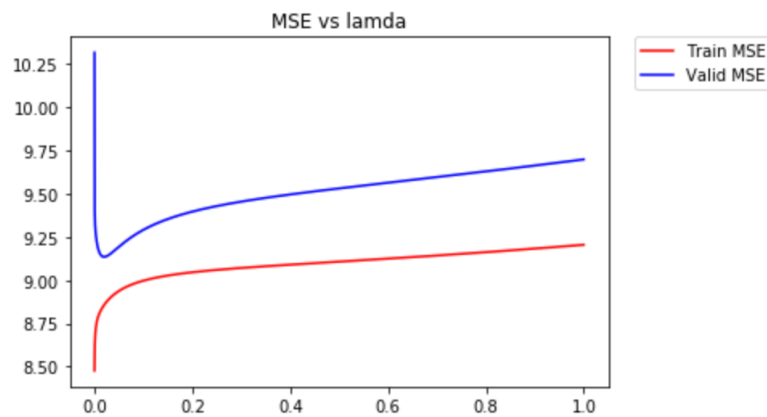


(Figure 1. Visualization of predicted output compared to real output for Training, Validation and Test Set)

In conclusion, the calculated MSE for Training set and Validation set was between ~ 6.5 and ~ 1417.9 . Moreover, we can observe that our prediction model shows excellent fit for training set, but not for the validation set.

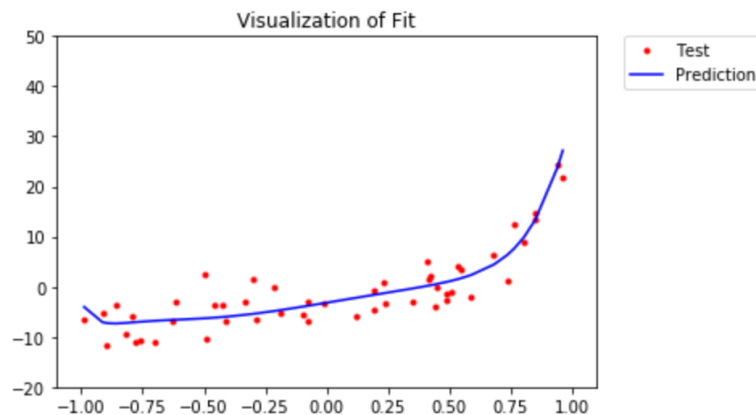
The prediction shows poor representation of the function and it indicates that the model is overfitting (Low MSE for training set, but high MSE for validation). More accurate prediction can be achieved by adding more data points or adding a penalty term to the error function to control coefficients (i.e. discouraging them from reaching large values)

2)



(Figure 2. MSE for different values of lamda)

From Figure 2, the best λ was 0.0197, which had the minimum MSE, 9.135 for the valid set.



(Figure 3. Visualization of the Fit on the Test set)

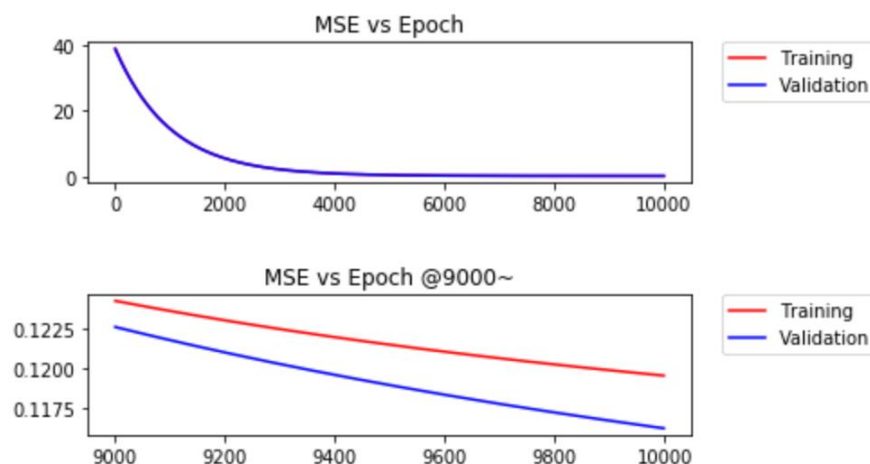
Adding L2 regulation improved the fit greatly compared to the method without regulation. In addition, the model has been tested with the separate test set to see the true performance. With current data, it seems the chosen lamda and the weights are good model where the fit is neither overfitting nor under fitting. However, it would require more data to conclude the model is truly good or not.

3)

By observation, the shape of the fit looks like an exponential graph or a polynomial with even degrees.

Q3 - Gradient Descent for Regression

1)



(Figure 4. Learning curve by stochastic gradient descent)

At epoch=10000 with step size $1e-6$, following values were obtained:

$w_0 = 3.90531794303802$, $w_1 = 3.8901811576993057$

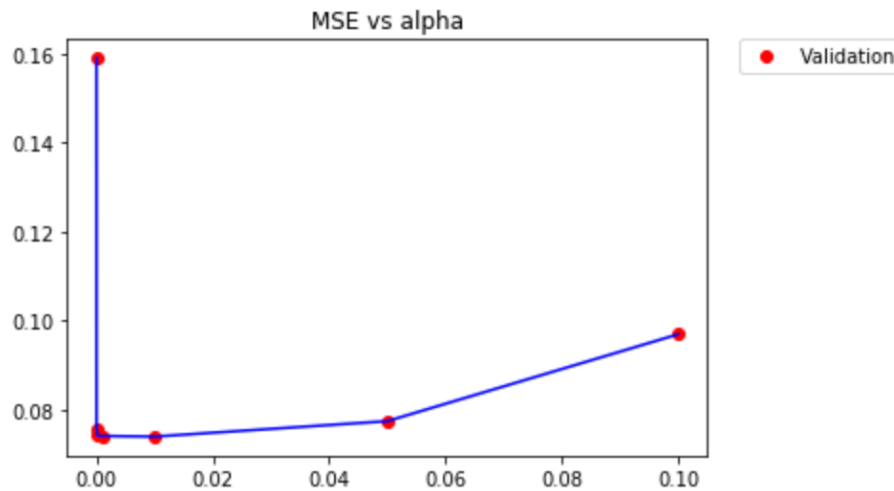
Train set MSE = [0.12764713274905526]

Valid set MSE = [0.1281358767073476]

The MSE values for training set and validation set were really close, but the training set showed slightly better performance.

2)

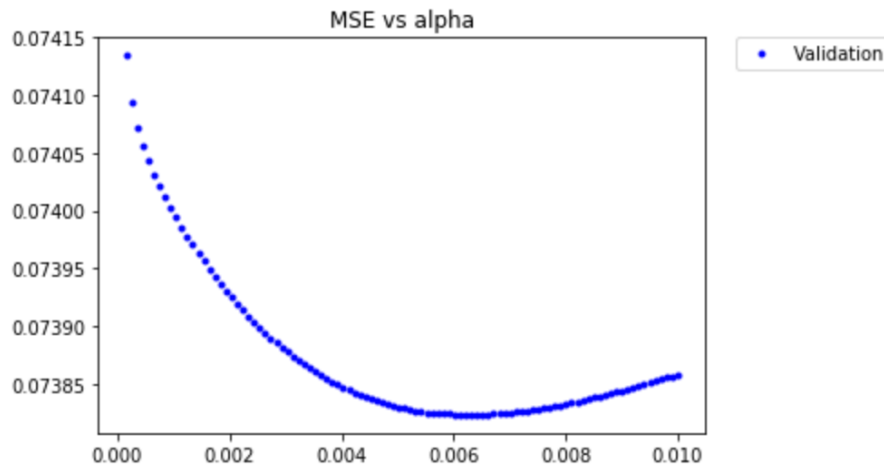
Minimum MSE 0.07385918397886929 at step size 0.01



(Figure 5. MSE vs Step Size)

To save computing time, first step size with multiple of 10s were used to see the general trend of MSE.

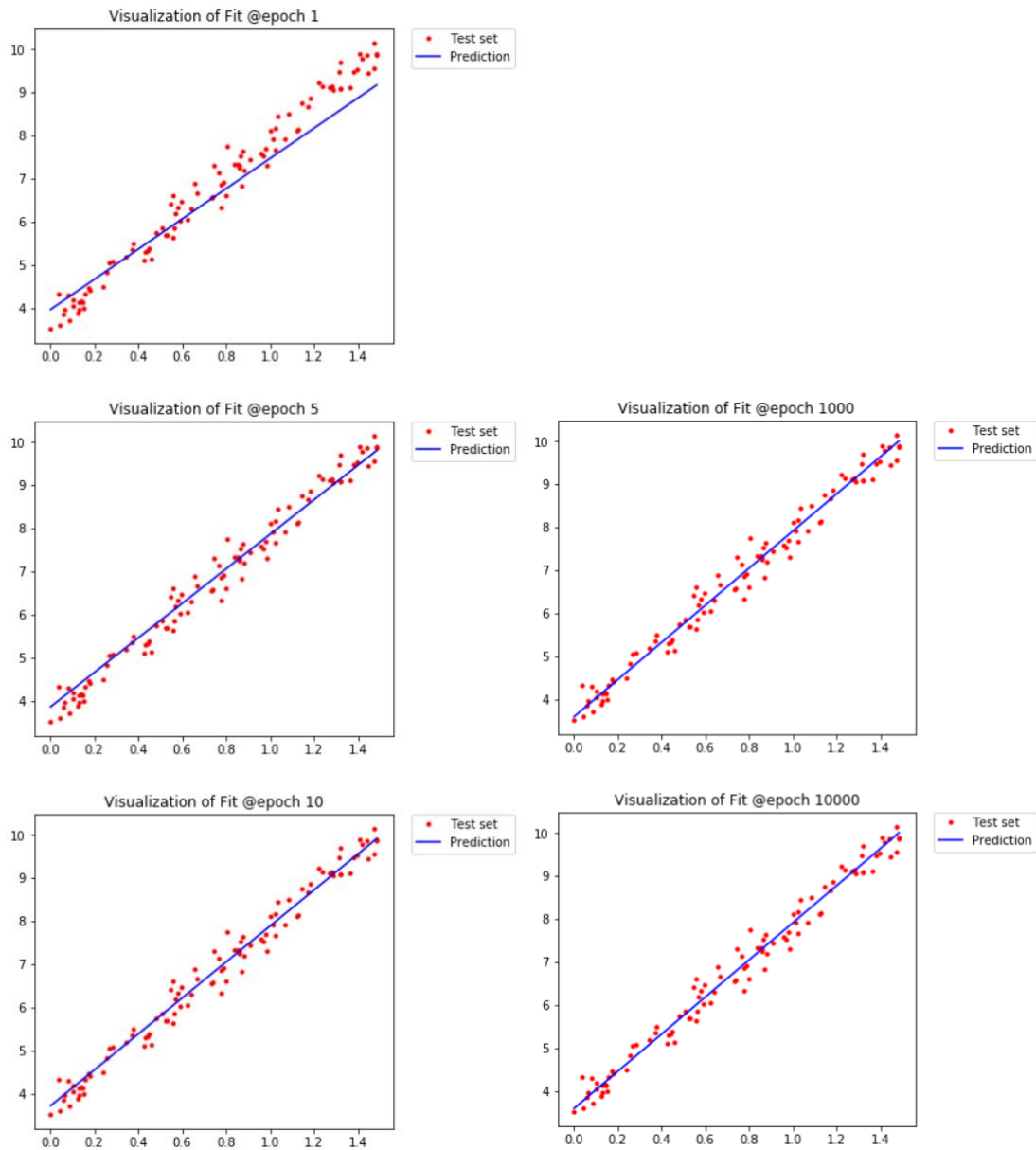
From the graph, MSE value decreases until the step size is $1e-6$ and increases later on. Thus, it can be assumed that best step size could be found somewhere in between $1e-2 \sim 5e-2$



(Figure 6. MSE vs Step Size)

Then, the range found above was thoroughly inspected to get accurate result. In Figure 6. , the best step size(~ 0.006318) was found with the minimum MSE value 0.0738. But, the MSE value for the test set was approximately 28.67

3)



(Figure 7. Visualization of the fit at epoch 1, 5, 10, 1000, and 10000)


As the epoch increased, the fit improve. However, after 1000 epoch, there was not much change in the fit, which indicates that the weights w_0 and w_1 have converged.

Q4 – Real life dataset

1)

By observation, there were 25 columns that had missing values.

	0	1	2		3	4	5	6	7	8	9	...
0	8	NaN	NaN	Lakewoodcity	1	0.19	0.33	0.02	0.90	0.12	...	
1	53	NaN	NaN	Tukwilacity	1	0.00	0.16	0.12	0.74	0.45	...	
2	24	NaN	NaN	Aberdeentown	1	0.00	0.42	0.49	0.56	0.17	...	
3	34	5.0	81440.0	Willingborotownship	1	0.04	0.77	1.00	0.08	0.12	...	
4	42	95.0	6096.0	Bethlehemtownship	1	0.01	0.55	0.02	0.95	0.09	...	
5	6	NaN	NaN	SouthPasadenacity	1	0.02	0.28	0.06	0.54	1.00	...	
6	44	7.0	41500.0	Lincolntown	1	0.01	0.39	0.00	0.98	0.06	...	
7	6	NaN	NaN	Selmacity	1	0.01	0.74	0.03	0.46	0.20	...	
8	21	NaN	NaN	Hendersoncity	1	0.03	0.34	0.20	0.84	0.02	...	
9	29	NaN	NaN	Claytoncity	1	0.01	0.40	0.06	0.87	0.30	...	



	0	1	2		3	4	5	6	7	8	9	...
0	8	58.826829	46188.336597	Lakewoodcity	1	0.19	0.33	0.02	0.90	0.12	...	
1	53	58.826829	46188.336597	Tukwilacity	1	0.00	0.16	0.12	0.74	0.45	...	
2	24	58.826829	46188.336597	Aberdeentown	1	0.00	0.42	0.49	0.56	0.17	...	
3	34	5.000000	81440.000000	Willingborotownship	1	0.04	0.77	1.00	0.08	0.12	...	
4	42	95.000000	6096.000000	Bethlehemtownship	1	0.01	0.55	0.02	0.95	0.09	...	
5	6	58.826829	46188.336597	SouthPasadenacity	1	0.02	0.28	0.06	0.54	1.00	...	
6	44	7.000000	41500.000000	Lincolntown	1	0.01	0.39	0.00	0.98	0.06	...	
7	6	58.826829	46188.336597	Selmacity	1	0.01	0.74	0.03	0.46	0.20	...	
8	21	58.826829	46188.336597	Hendersoncity	1	0.03	0.34	0.20	0.84	0.02	...	
9	29	58.826829	46188.336597	Claytoncity	1	0.01	0.40	0.06	0.87	0.30	...	

(Figure 8. Filling missing values with mean)

Filling out missing values in the dataset can be tricky. There are many techniques to approach this problem and using "sample mean" is one of them. This approach is not a bad choice when the number of missing value is significantly small compared to the total number of data.

However, in this dataset, there are total 1994 data points and 1675 missing values for certain attributes like *PoliceReqPerOffic*, *LemasSworn* and etc. This is approximately 84% and it is hard to suggest that filling mean values is ok.

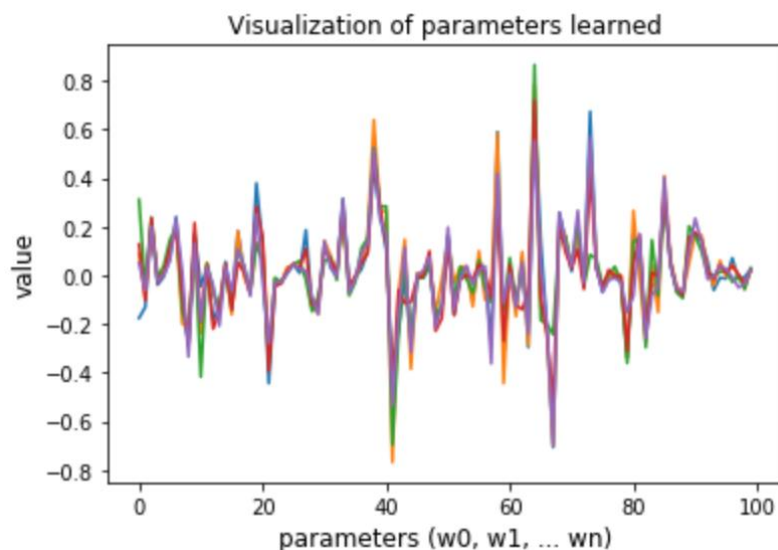
For alternative choices we have:

1. Replace missing values with median/mode
2. Delete rows or columns with missing values
3. Predict the missing value using methods like linear regression
4. Use other algorithms(e.g. KNN) that supports missing values

First of all, the data has to be cleaned up by eliminating non-predictive attributes described in the Data Set Description. Thus, first 5 columns (*state*, *county*, *community*, *communityname*, and *fold*) will not be used for training. Moreover, we can observe that there are 25 attributes that has missing values. 24 attributes have high ratio of missing data (~84%) and 1 attribute (i.e. *OtherPerCap*) with 1 missing value. Attributes that have large ratio of missing data will be removed rather than filling it with mean to avoid false prediction, and for attribute *OtherPerCap* will be filled with mean. Moreover, the cleaned up data will give better performance to the prediction model since it has less parameters to compute.

The complete data set can be found in *Datasets* folder.

2)



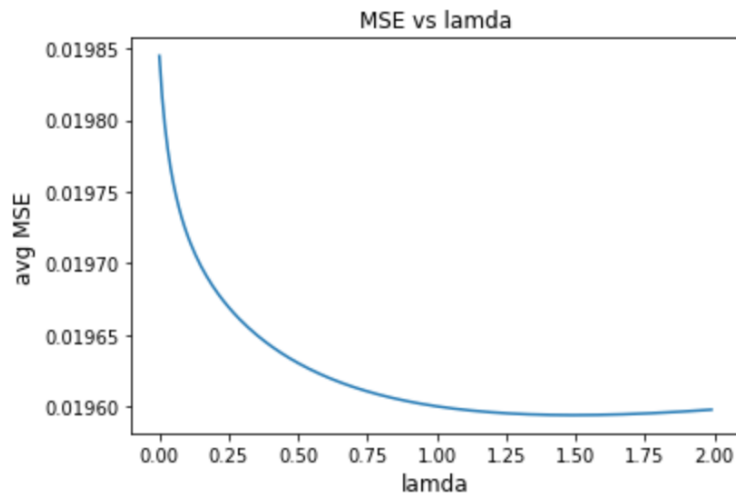
Dataset	MSE on Test set
CandC_train/test1.csv	0.017956454132902627
CandC_train/test2.csv	0.018766924979192308
CandC_train/test3.csv	0.019084229435376156
CandC_train/test4.csv	0.019282209263470854
CandC_train/test5.csv	0.02413394883262758

(Figure 9. Parameters learned and MSE for each Dataset)

Mean of the MSE was 0.019844753328713903.

More details on the values of the parameters can be found in the jupyter notebook.

3)



Minimum Average MSE 0.019593849068351425 @lamda = 1.49

(Figure 10. MSE vs Lamda with Ridge-regression)

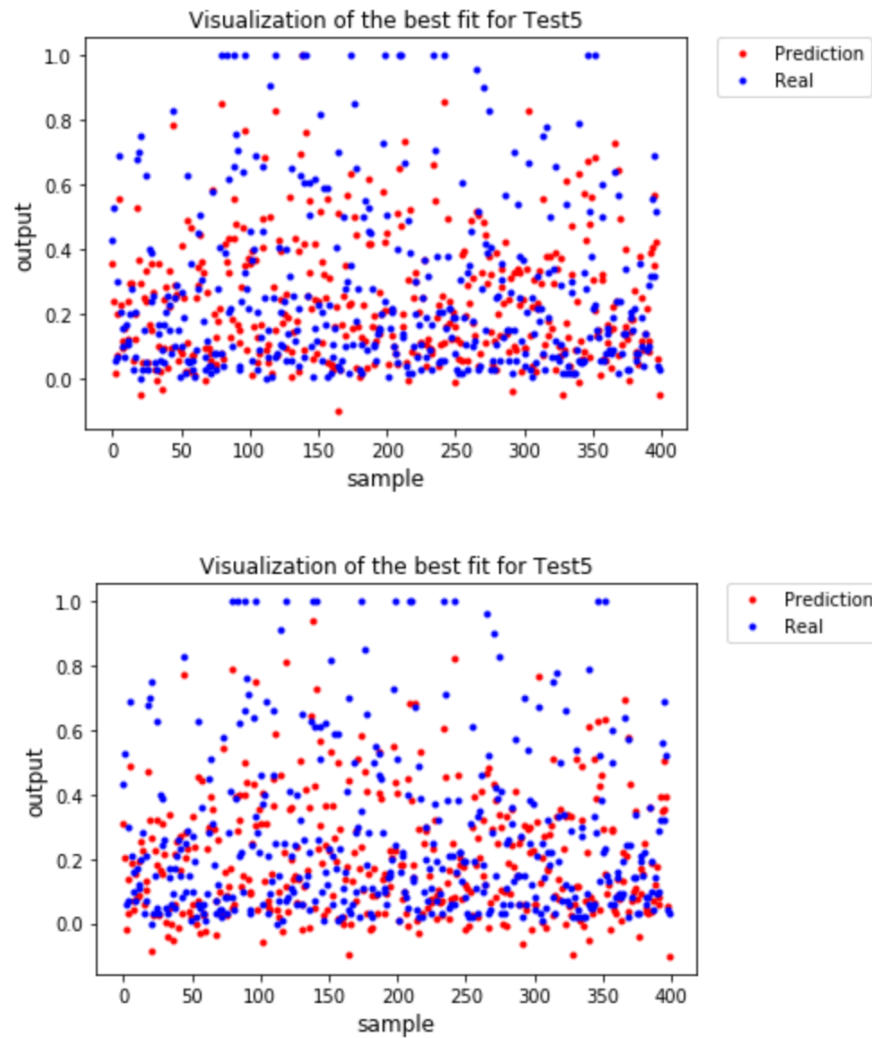
From the result above, the minimum MSE was achieved when the value of lamda was equal to 0.149

Since the features are linear to the model,

$$y = w_0 * x_0 + w_1 * x_1 + w_2 * x_2 + \dots + w_n * x_n \text{ (n features)}$$

The parameters that are small are going to have less impact on the output, compare to the larger parameters. Thus, by dropping the smallest absolute parameters, we can choose the most effective parameters.

Each time, the smallest feature will be dropped, and the MSE will be calculated. If the MSE is in acceptable range (i.e. equal or below the average MSE found at part a), then that feature will be removed. The procedure will be repeated until the MSE reaches the average MSE.



Final MSE after selecting features : [0.019326688217631095]

(Figure 11. Prediction of output with 101 features (above) and 67 features (below))

Compared to the previous model, the amount of features significantly decreased (101 -> 67 features). Because the amount of parameters that have to be computed, has decreased, the total computing time has decreased as well. Moreover, the new model's accuracy in terms of MSE, is as good as the old model.