MDL Assignment-1
Report

Mrinal khubchandani (201810153)
Karsh tandon (2018101034)
TEAM NO : 84

Question 1:

Solution:

```
dbfile = open('Q1_data/data.pkl','rb')
db = pickle.load(dbfile)

X = db[:,0]
y = db[:,1]

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.1)
```

In this first load the data in db and then save X coordinates into X and Y coordinates into y.
We then divide the data into training set and test set

```
for i in range(10):
    X_train,X_temp,y_train,y_temp = train_test_split(X_train,y_train,test_size = 0.1)
    X_train_number_list.append(X_temp)
    Y_train_number_list.append(y_temp)
```

Then we divide the training data into 10 equal subsets for different training models
then there is a loop for the degree of the model

```
for j in range(10):
    X_temo = np.array(X_train_number[j])
    X_temo = X_temo[:, np.newaxis]
    X_poly = poly.fit_transform(X_temo)

    reg.fit(X_poly,Y_train_number[j]);
    temp = reg.predict(X_test_poly)
    prediction_list.append(temp)
```

then we run a loop for getting predictions on models that are trained on different training sets

```
prediction = np.array(prediction_list)

expec = np.mean(prediction,axis = 0)
bias = np.subtract(expec,y_test)
bias**=2

varience = np.var(prediction,axis=0)

bias_avg = np.mean(bias)
bias_avg = math.sqrt(bias_avg)

var_avg = np.mean(varience)

final_varience_list.append(var_avg)
final_bias_list.append(bias_avg)
```

then we calculate bias and varience for the model in one go by using vectorization techniques. Here bias represents the bias for each x in test set for different instances of the model and varience represents variance for different intsances of the model. Then we calculate rms for bias and mean for variance and append into a list.

After this is done we have bias and variance for models of degrees (1,2,3...10) with us and we just plot the graph.

Analysis:

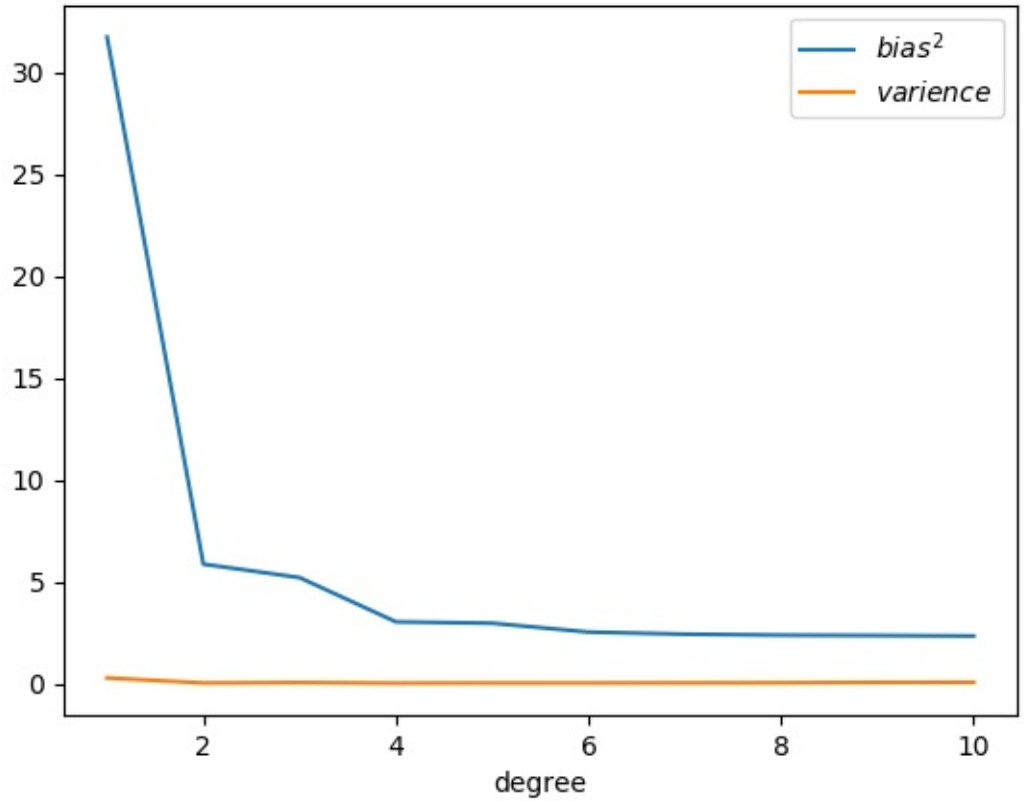
Table for bias and variance

Degree	Bias	Variance
1	5.546046	0.250193
2	2.346601	0.043405
3	2.179609	0.073022
4	1.743362	0.025810
5	1.704418	0.038493
6	1.640952	0.033606
7	1.609311	0.041954
8	1.599059	0.069137

9	1.598337	0.069770
10	1.592666	0.077050

The bias for the model keeps on decreasing with increase in the degree of the polynomial.This happens because we include more features in the model resulting in less bias.

The variance decreases a bit initially and then keeps on increasing because of overfitting. With increase in the degree of the polynomial the spread of the predicted value increases over the test set increasing the variance.



Graph plot for question 1

Question 2:

Solution:

```
xfile1 = open('Q2_data/X_train.pkl','rb')
xfile2 = open('Q2_data/X_test.pkl','rb')
yfile1 = open('Q2_data/Y_train.pkl','rb')
yfile2 = open('Q2_data/Fx_test.pkl','rb')

X_train_number = pickle.load(xfile1)
Y_train_number = pickle.load(yfile1)
X_test = pickle.load(xfile2)
y_test = pickle.load(yfile2)
```

First load the test and training data.
The reset of the steps are the same as the first question

Analysis:

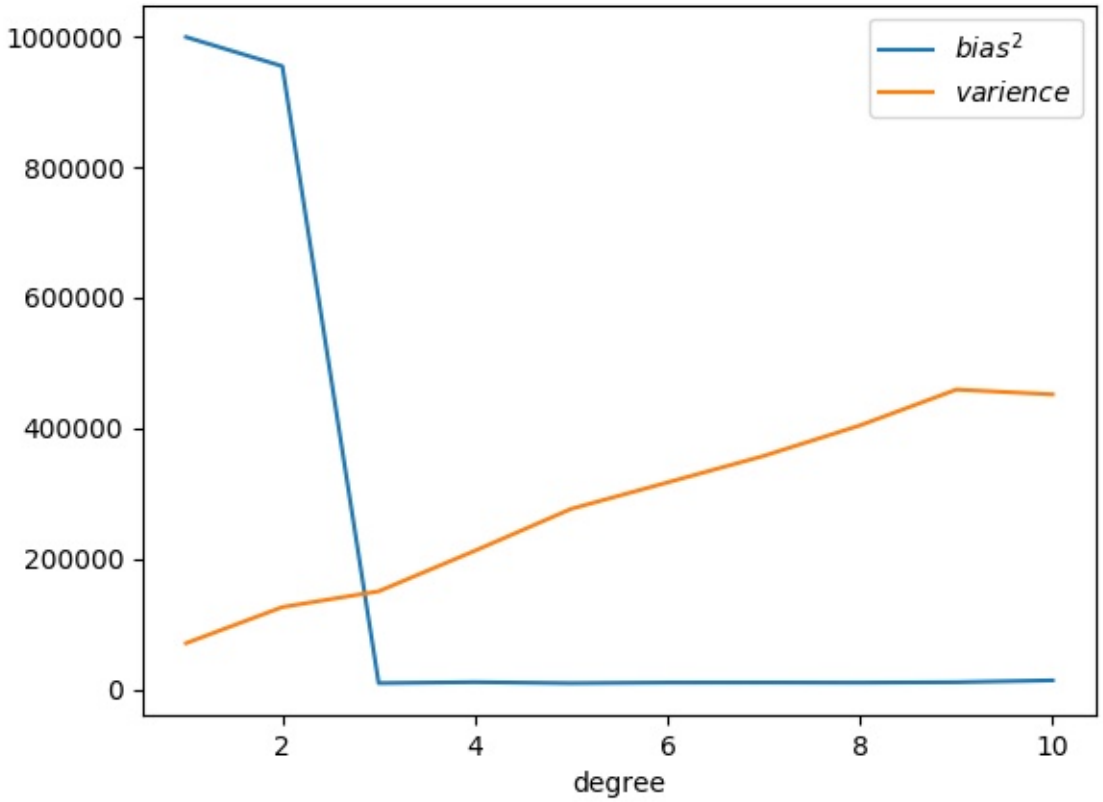
Table for bias and variance

Degree	Bias	Variance
1	999.614124	70545.489146
2	977.046198	125870.855549
3	96.900620 1	50073.739546
4	104.438250	212235.708325
5	96.639507	276388.480255
6	101.235300	316863.498437
7	101.662559	357510.984757
8	100.744326	404286.670686
9	103.997534	459132.378372
10	116.743129	451749.786753

As the degree of the polynomial increases,the bias decreases initially and then becomes almost constant because we have already included many features and with the addition of a new feature bias doesnt change that much.(Initially the bias is high because of under fitting).

As the degree of the polynomial increases the varience increases.This happens because of over fittiong.

We can say that the data best fits on a curve of degree 3 as the total error is minimum for polynomial of degree 3. This can be seen by plotting the bias variance tradeoff.
So Below this degree it is under fit and beyond it is overfit.



Graph plot for question 2