Machine, Data and Learning

Assignment 1

Jashn Arora

2018114006

Akshit Garg

2018113006

Details of the Algorithm

Linear Regression is a supervised machine learning algorithm where the predicted output is continuous and has a finite slope. It's used to predict values within a continuous range, rather than trying to classify them into categories. It performs the task to predict a dependent variable value (y) based on a given independent variable (x). So, this regression technique finds out a linear relationship between x (input) and y (output).

Hypothesis Function for Linear Regression

Linear regression uses traditional slope-intercept form, where m and b are the variables our algorithm will try to "learn" to produce the most accurate predictions. x represents our input data and y represents our prediction.

$$y = mx + b$$

Question 1:

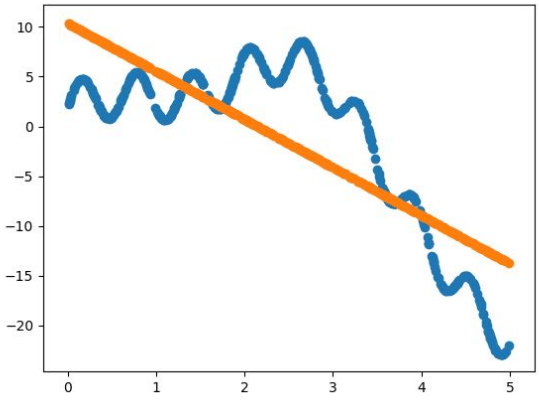
Explanation of Code

1. Loaded the data using pickle.load
2. Splitted the data into training and test data using train_test_split function(in 90:10 ratio)
3. The training data is further divided into 10 segments
4. We transformed input dimensions using PolynomialFeatures.fit_transform for fitting x for appropriate degrees.
5. Model was generated by linear_model.LinearRegression().
6. We generated 10 models each with 450 values of training set for each degree upto 9.
7. It was tested on testing data of 500 values.
8. Bias was calculated as the average of the square of the difference between the mean prediction of 10 models and actual value.
9. Variance: For each testing point variance for the 10 models is calculated, all these 500 variances are averaged out to get variance of particular degree model.

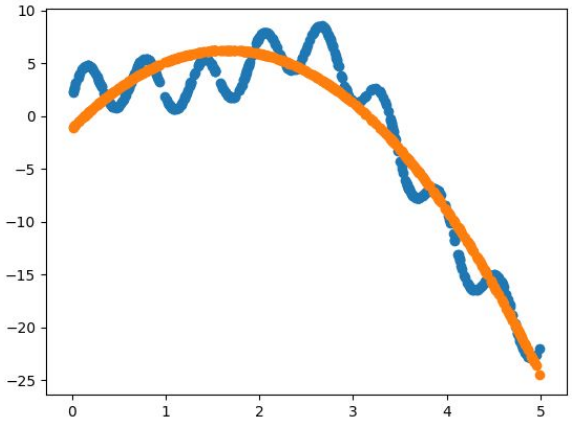
Results:

Degree	Bias^2	Variance	Bias
1	31.517197128271825	0.19280586070624245	5.614017913070088
2	5.52380138149305	0.05207260530626064	2.3502768733689763
3	4.688766421084347	0.049453468182275445	2.165355957131378
4	2.8323559295630845	0.029977926437126108	1.6829604658348587
5	2.670749149614037	0.03157025348726591	1.6342426838184214
6	2.570334329812322	0.03441387976884801	1.6032262254006207
7	2.4273168564625682	0.03661106226769666	1.5579848704215866
8	2.4375032065743185	0.0453308210738345	1.5612505265249133
9	2.4388443118241936	0.048045471478716746	1.5616799645971622

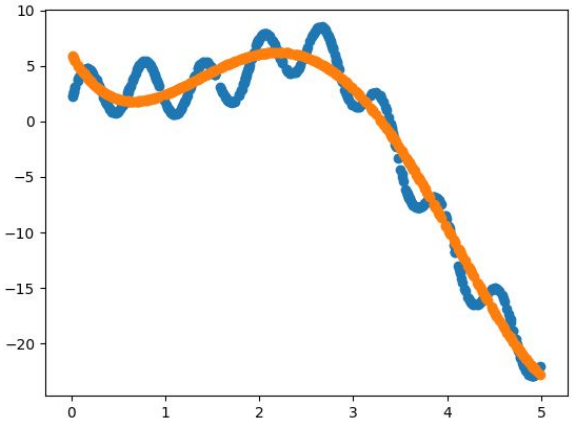
Some Plots of **Testing data** vs **Predicted values**



Degree 1



Degree 2



Degree 3

Model Actual

Analysis:

- As complexity i.e Degree of the model increases the Bias decreases because we are increasing the number of features. Hence the model is fitting the curve even better (evident from data vs predicted curve plot at *page 3*) decreasing the bias as we increase the complexity .

- As complexity increases variance decreases initially and then it increases because of overfitting as the curve generated by the model tends to follow only the training set.

Question 2:

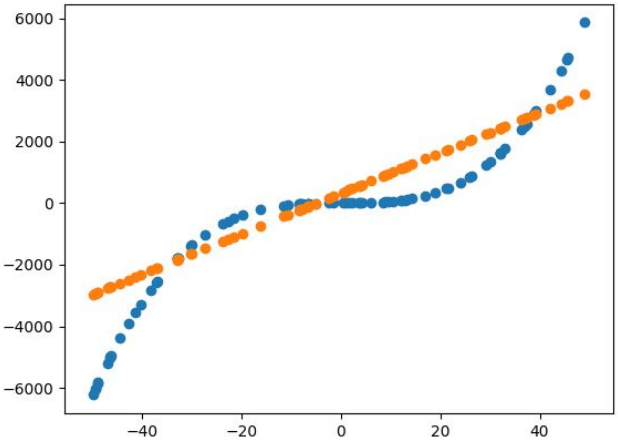
Explanation of Code

1. Loaded the training and testing data using pickle.load.
2. We transformed input dimensions using reshape and PolynomialFeatures.fit_transform.
3. Model was generated by linear_model.LinearRegression().
4. We generated 20 models with 400 values of training set for each function of degree x.
5. It was tested on testing data of 80 values.
6. Bias was calculated as the average of the square of the difference between the mean prediction of 20 models and actual value.
7. Variance is calculated as the average variance of each prediction.

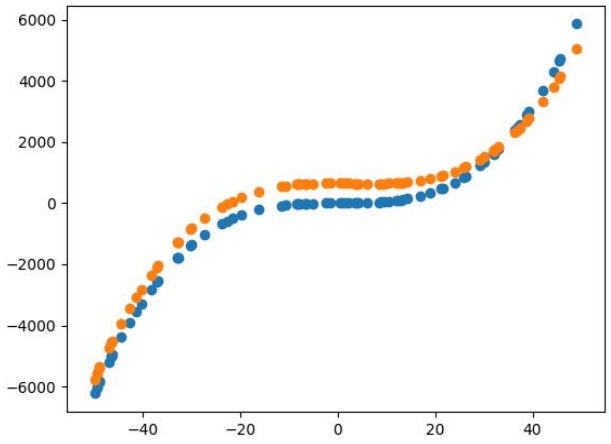
Results:

Degree	Bias^2	Variance	Bias
1	999228.3968719237	70545.48914575046	999.6141239858127
2	954619.273794425	125870.85554877335	977.0461983931082
3	9389.730116791201	150073.7395464768	96.90061979570203
4	10907.348134071308	212235.70832526154	104.43825033995594
5	9339.194291326017	276388.4802547406	96.63950688681113
6	10248.585941147874	316863.49843748985	101.23529987681113
7	10335.275861649096	357510.98475735466	101.66255879943755
8	10149.419243937276	404286.670685786	100.74432611287484
9	10815.48703657424	459132.37837248633	103.9975338004428

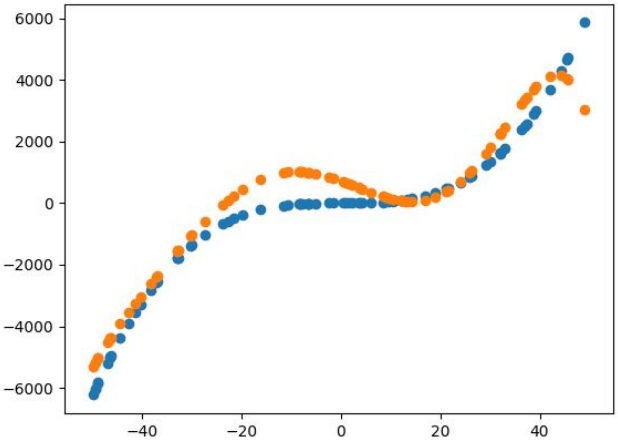
Some Plots of **Testing data** vs **Predicted values**



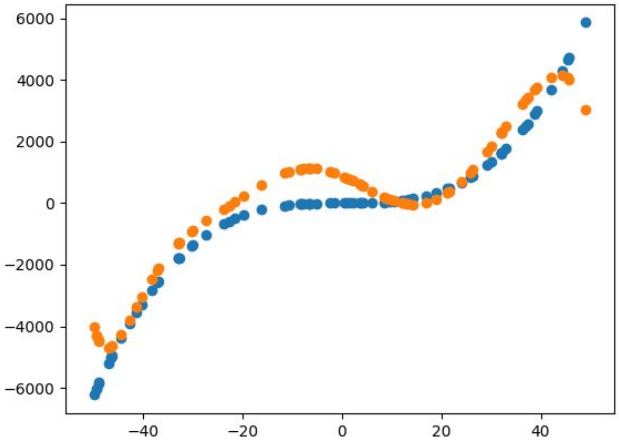
Degree 1



Degree 3



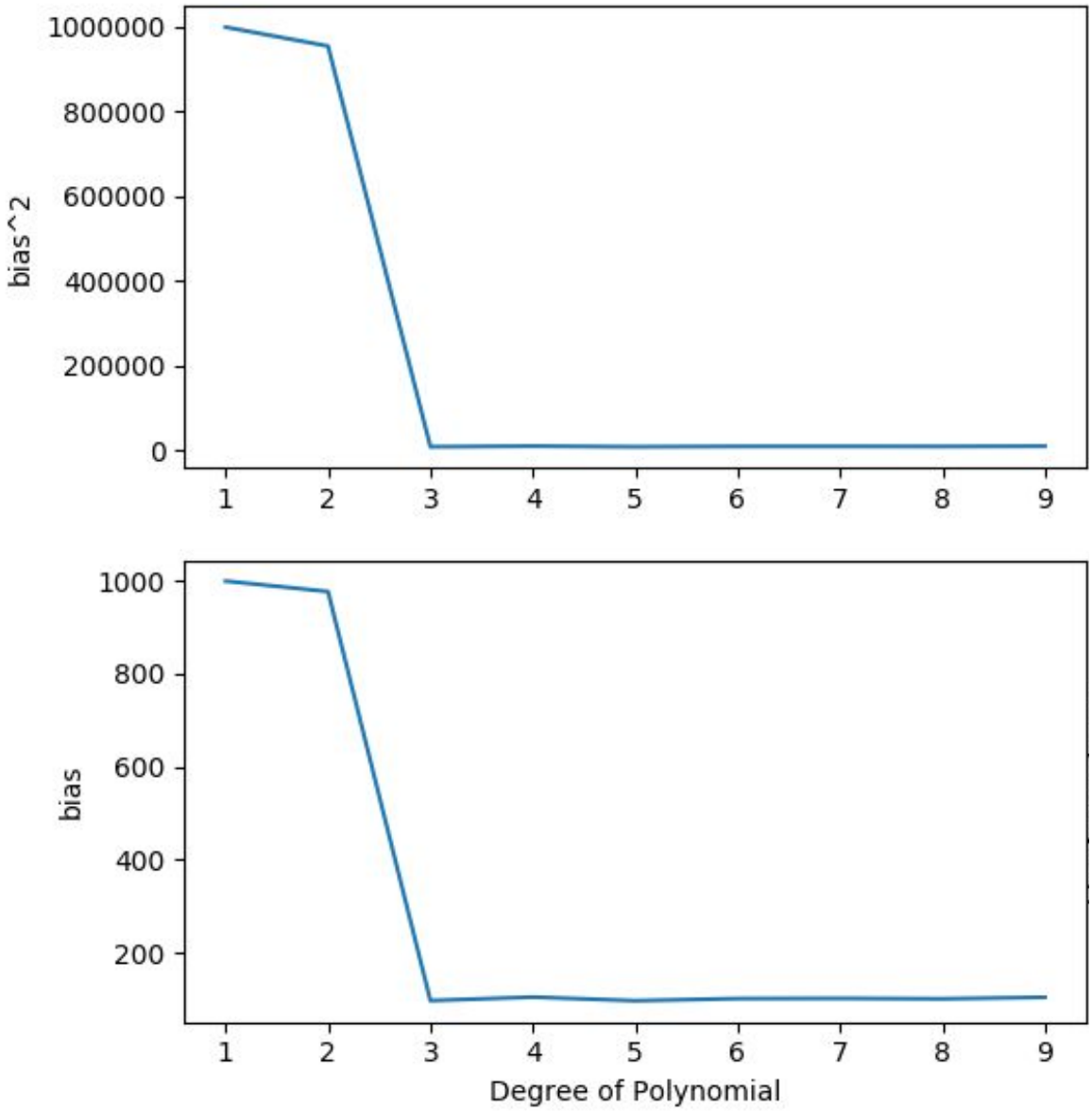
Degree 6

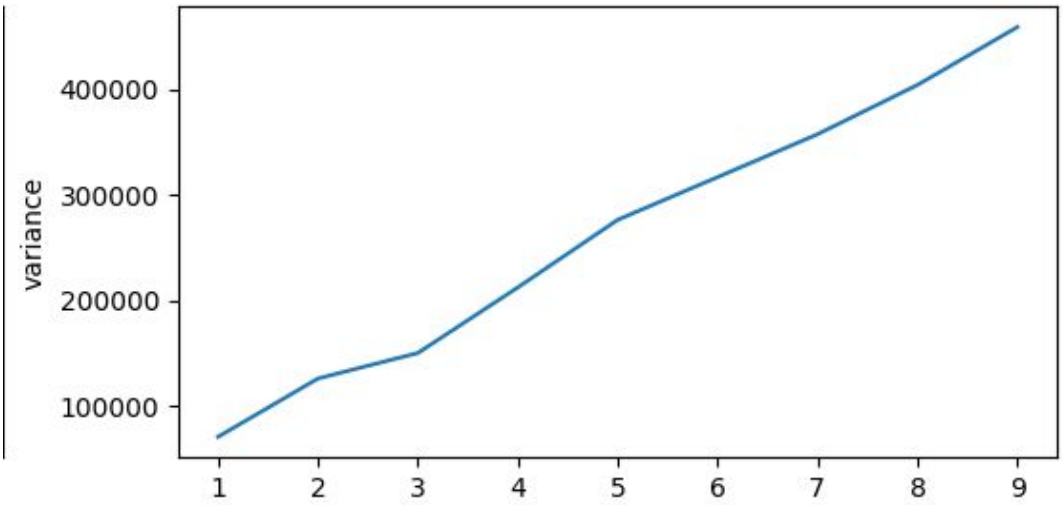


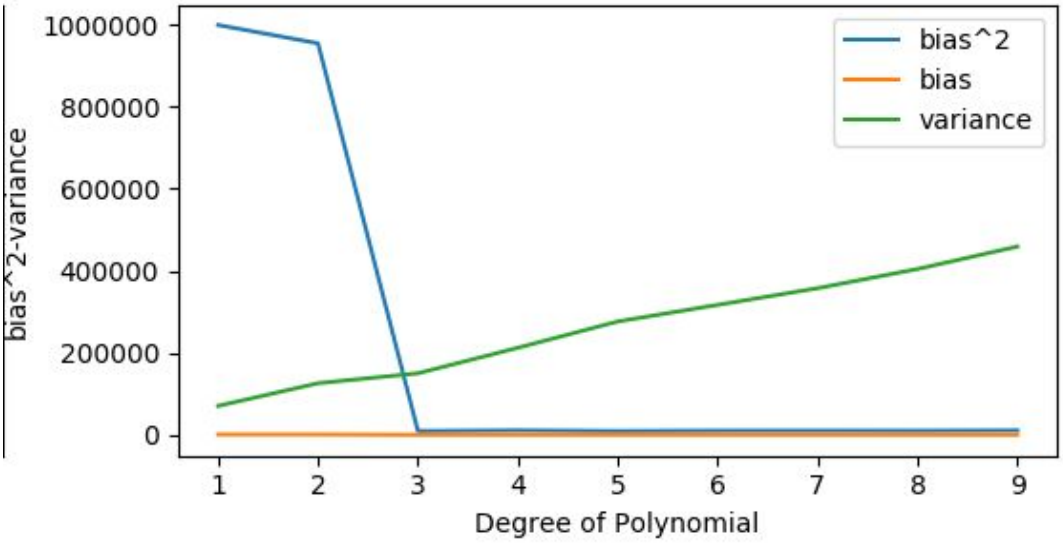
Degree 9

Model Actual

Graphs:


1000000
800000
600000
400000
200000
0
bias^2
1
2
3
4
5
6
7
8
9
1000
800
600
400
200
bias
1
2
3
4
5
6
7
8
9
Degree of Polynomial


variance
400000
300000
200000
100000
1
2
3
4
5
6
7
8
9


bias^2-variance
1000000
800000
600000
400000
200000
0
bias^2
bias
variance
1
2
3
4
5
6
7
8
9
Degree of Polynomial

Analysis:

- As complexity increases variances increases, this is because the model is getting overfit on training data.

- The bias decreases initially but after n = 3 , the value becomes almost constant, because at cubic polynomial the model almost fits the data and further complex models aren't much different from this cubic model(as clear from the actual data vs predicted data plots for various degree models at *page 6*)

- At degree = 3, Bias^2+ variance + sigma is low i.e total error is low hence the model best fits at n = 3.

- Initially the model was underfit so there was high bias and low variance and as the complexity increases the model tends to become overfit resulting in low bias and high variance.

- The Data seems to be of type of cubic polynomial because total error=variance + bias^2 is minimum for linear regression of a cubic polynomial.