Assignment 1: Report

Team:
Sharadha Iyer: 2019101048
Hasvitha Varma: 2019101030

2.1 Task 1 : Linear Regression

The linear regression model takes given pairs of x and y values from the training data as arguments and fits them along a straight line/curve depending on the order of the x value projection.

LinearRegression().fit() does this in the following steps:

- ➔ Arbitrarily initializing a set of coefficients for these models.
- ➔ Calculating mean squared error for a predicted model
- ➔ The regressor now penalizes the existing model and consequently provides us a new set of coefficients.
- ➔ These two steps are repeated until the penalizing has little to no effect on the existing model

Parameters:

fit_intercept : *bool, default=True*
Whether to calculate the intercept for this model. If set to False, no intercept will be used in calculations (i.e. data is expected to be centered).

normalize : *bool, default=False*
This parameter is ignored when `fit_intercept` is set to False. If True, the regressors X will be normalized before regression by subtracting the mean and dividing by the l2-norm. If you wish to standardize, please use `StandardScaler` before calling `fit` on an estimator with `normalize=False`.

copy_X : *bool, default=True*
If True, X will be copied; else, it may be overwritten.

n_jobs : *int, default=None*
The number of jobs to use for the computation. This will only provide speedup for n_targets > 1 and sufficient large problems. `None` means 1 unless in a `joblib.parallel_backend` context. `-1` means using all processors. See Glossary for more details.

positive : *bool, default=False*
When set to `True`, forces the coefficients to be positive. This option is only supported for dense arrays.

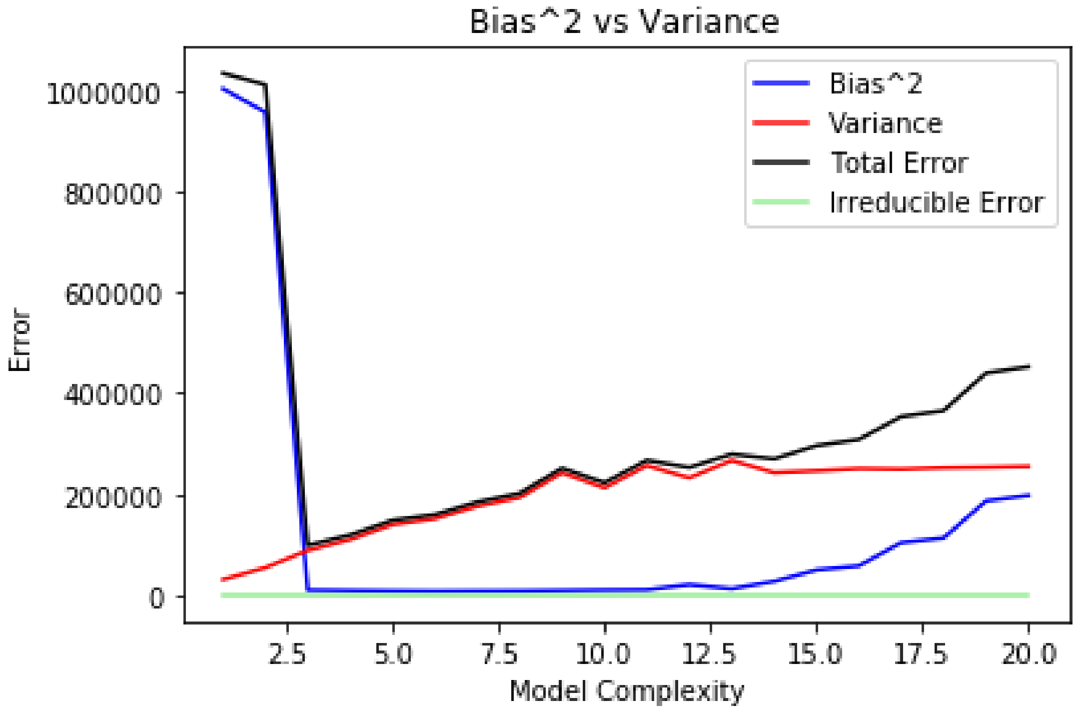
2.2.2 Task

Bias and variance change on varying function classes

Here is a table of **Degree-Bias-Variance-TotalError**

Degree	Bias	Variance	Total Error
1	231.357297	30861.144128	1.033871e+06
2	227.326495	54270.382667	1.010896e+06
3	15.970239	88571.607355	9.860308e+04
4	10.159046	109998.124526	1.190728e+05
5	6.952178	139727.351411	1.481205e+05
6	8.811795	151462.424236	1.593715e+05
7	2.629138	176036.484613	1.842606e+05
8	7.689096	193223.168482	2.016299e+05
9	6.241115	242173.902770	2.513633e+05
10	11.898679	212535.367371	2.226044e+05
11	9.204463	256419.447705	2.670020e+05
12	5.094152	232219.386098	2.528695e+05
13	12.316387	266680.328945	2.792715e+05
14	21.939739	242568.968659	2.696649e+05
15	30.129640	246238.836329	2.959643e+05
16	31.325810	250506.007469	3.081951e+05
17	43.970706	249467.889831	3.532988e+05
18	46.828925	252249.732550	3.652708e+05
19	61.438390	253243.693744	4.397213e+05
20	65.461974	254879.277230	4.522052e+05

And here is the **Bias^2 vs Variance** graph for reference (Task 4)



- ➔ The best fit possible (Minimum Total Error) is observed around a polynomial of degree 3.
- ➔ Bias stays low until around degree 10-11, but the variance increases steadily.
- ➔ Hence, the total error increases as well.

Polynomials of degree 1 & 2:

- ➔ They are oversimplified and do not generalize the data well, leading to a high bias
- ➔ They are consistent with not very scattered predicted values (lesser features), which is why they have low variance.
- ➔ These models are overall inaccurate, evident from the total error value.

Polynomial of degree 3:

- ➔ The lowest total error is demonstrated by this regression.

- ➔ We can see that it generalizes best to data not seen by the model before.

Polynomials of degree 4+:

- ➔ Variance increases gradually with an increase in complexity.
- ➔ The is expected since the number of features is higher, leading to overfitting of models over the trained data, and can be explained by regression models trying to fit the noise from the training data.
- ➔ The flexibility provided by having more features leads to noise being captured which causes incorrect prediction for unseen data.
- ➔ There is relatively low bias as the more complex model is able to model the training set well.

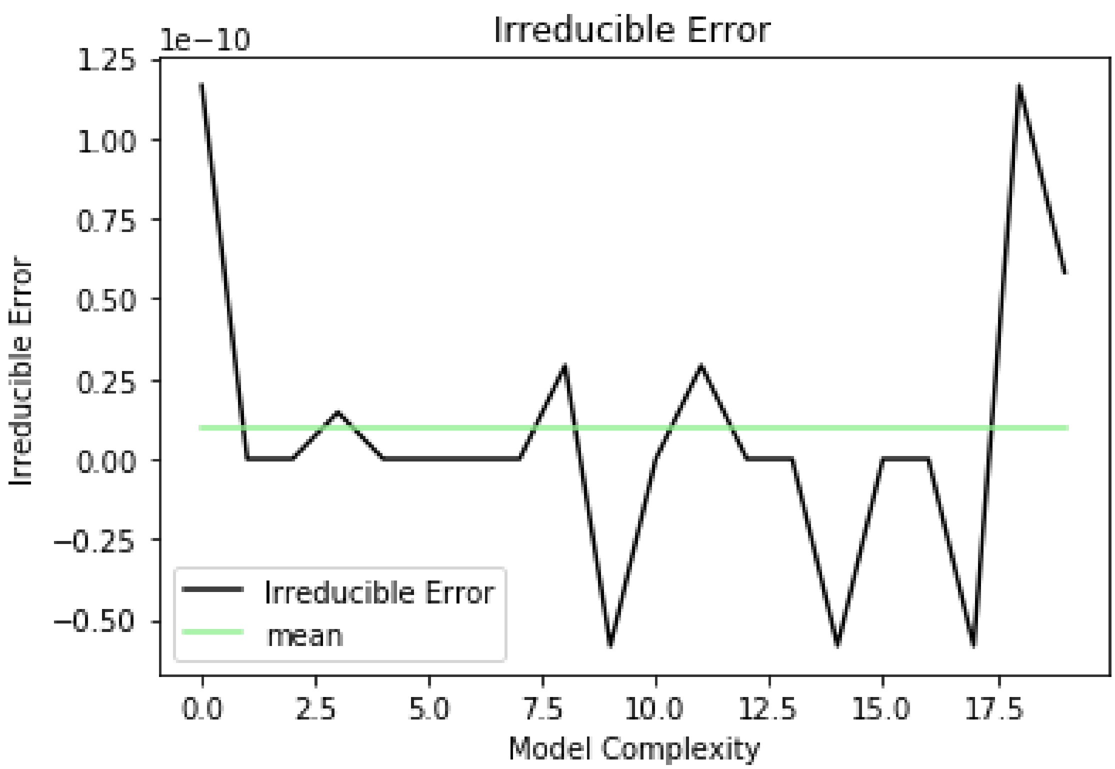
2.3 Task 3 : Calculating Irreducible Error

Here is the **irreducible error by degree** table

Degree	Irreducible Error
1	1.164153e-10
2	0.000000e+00
3	0.000000e+00
4	1.455192e-11
5	0.000000e+00
6	0.000000e+00
7	0.000000e+00
8	0.000000e+00
9	2.910383e-11
10	-5.820766e-11
11	0.000000e+00
12	2.910383e-11

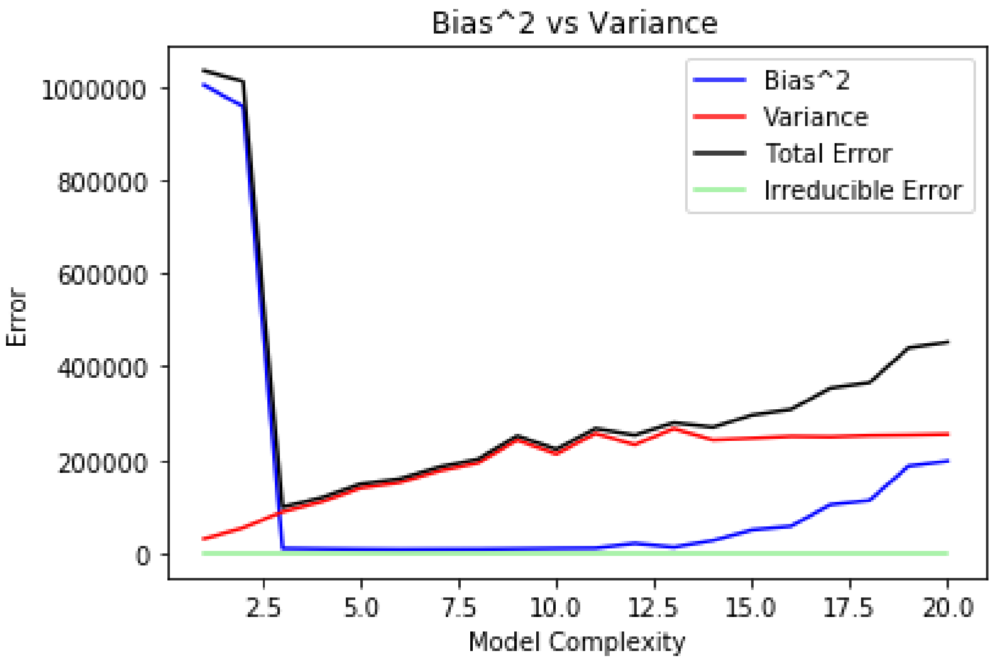
13	0.000000e+00
14	0.000000e+00
15	-5.820766e-11
16	0.000000e+00
17	0.000000e+00
18	-5.820766e-11
19	1.164153e-10
20	5.820766e-11

Here is the **irreducible error vs degree** graph



- ➔ We can see that the irreducible error shows no consistent patterns or variations.
- ➔ They cannot be reduced, regardless of the algorithm applied
- ➔ It is a measure of noise in the data, which is independent of the model or regression analysis used to make predictions.

2.4 Task 4 : Plotting *bias^2 - variance* graph




```
Observations above, in 2.2.2.
```