MDL Assignment 1 Report

Team 48:

Abhijeeth Singam, Saravanan Senthil

Task 1:

LinearRegression().fit()

Given test data, LinearRegression().fit() minimises the square difference between the model and the data to find the weights and bias of a regression that yield the most accurate results. In the case of a simple regression, this happens to be the 'line of best fit'.

$$y = b + \sum_{i=1}^{n} w_1 x_1$$

To calculate the aforementioned 'accuracy' of a set of weights and bias, the MSE (Mean Squared Error) is used. LinearRegression().fit() aims to reduce this MSE as much as possible to result in the most accurate set of weights and bias that it can achieve.

Task 2:

Bias

Bias is one of two ways of measuring the accuracy of an ML model. It represents the difference between the 'expected value' or 'average value' of the model and the actual value we are trying to predict.

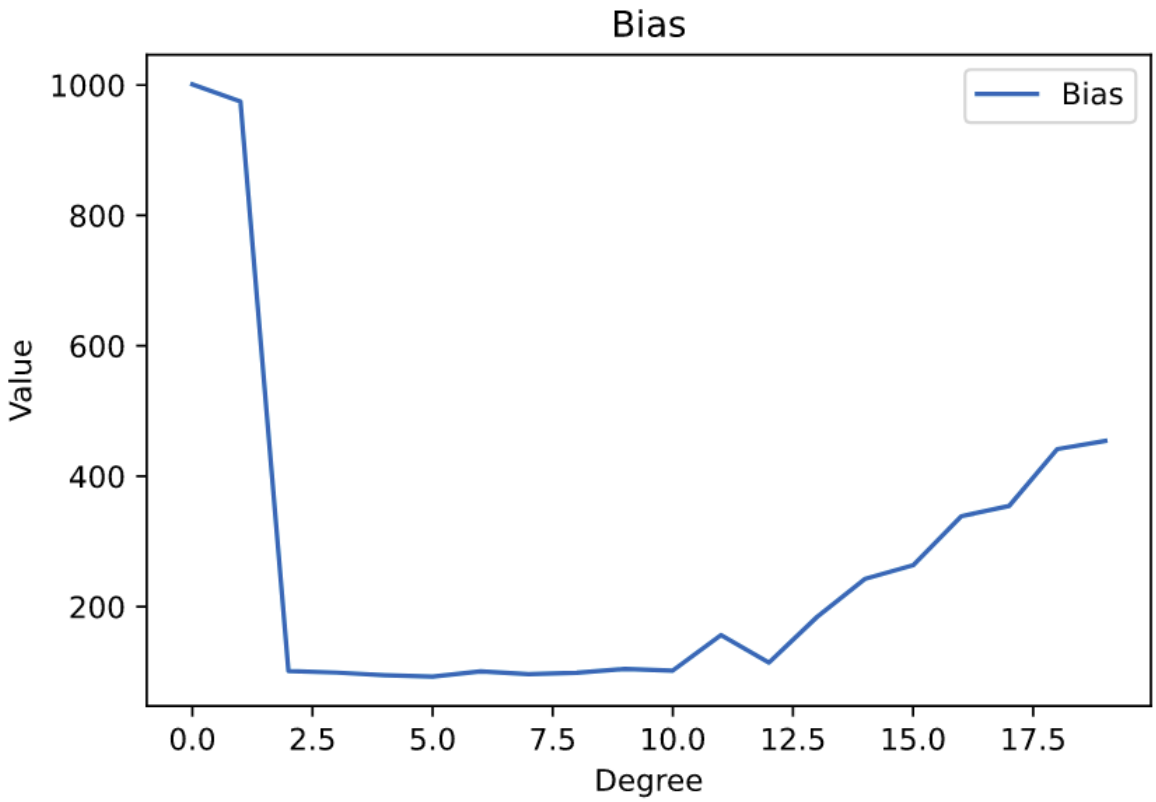
Bias is calculated by taking the square root of $Bias^2$ using the formula:

$$Bias^2 = (E[\hat{f}(x)] - f(x))^2$$

Code:

```
bias2 = np.mean( (np.mean(predMatrix, axis = 0) - testData[:, 1] ) ** 2 )
bias = np.sqrt(bias2)
```

where predMatrix is the collection of y_predict from the trained models for each degree



Variance

Variance is the other way in which the accuracy of an ML model is measured. It represent the 'variability' of the model's prediction, i.e. how much the predicted values vary for different realisations of that model.

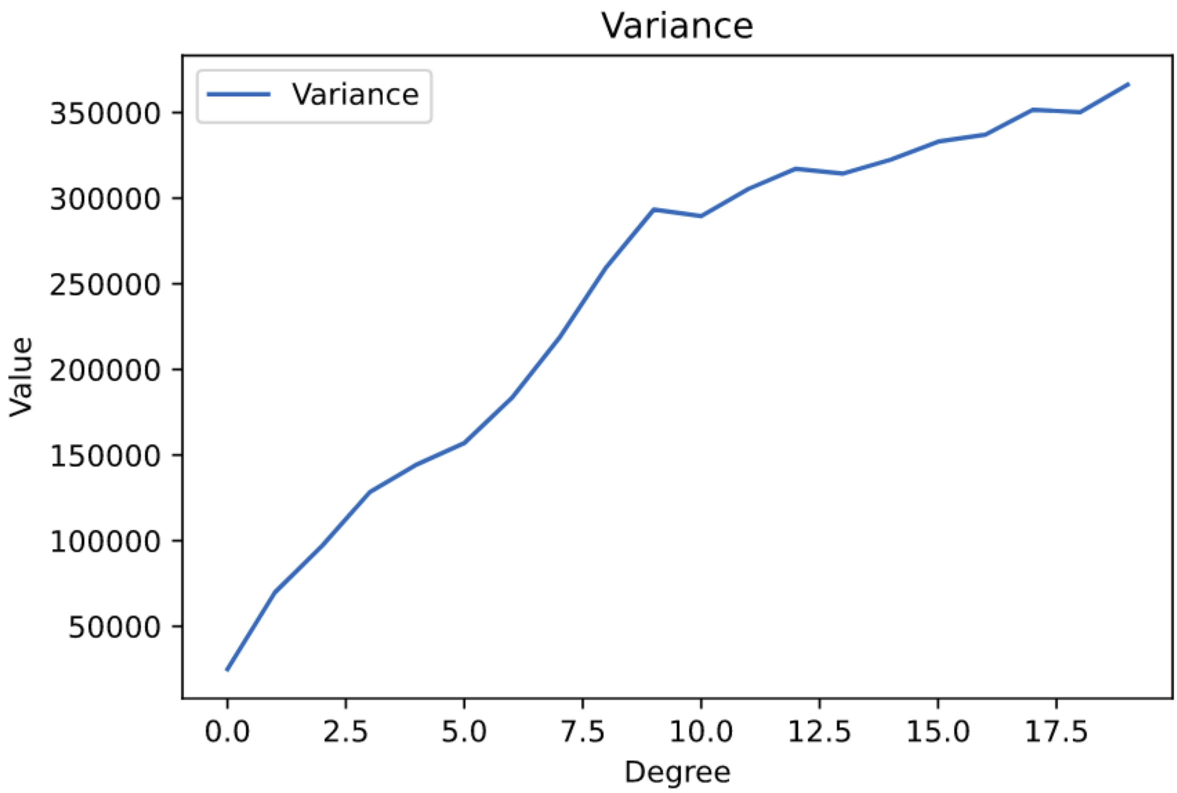
Variance is calculated using the formula:

$$Variance = E\left[(\hat{f}(x) - E[\hat{f}(x)])^2\right]$$

Code:

```
variance = np.mean(np.var(predMatrix, axis = 0))
```

where predMatrix is the collection of y_predict from the trained models for each degree
Where np.var is numpy's builtin to function compute variance



Task 3:

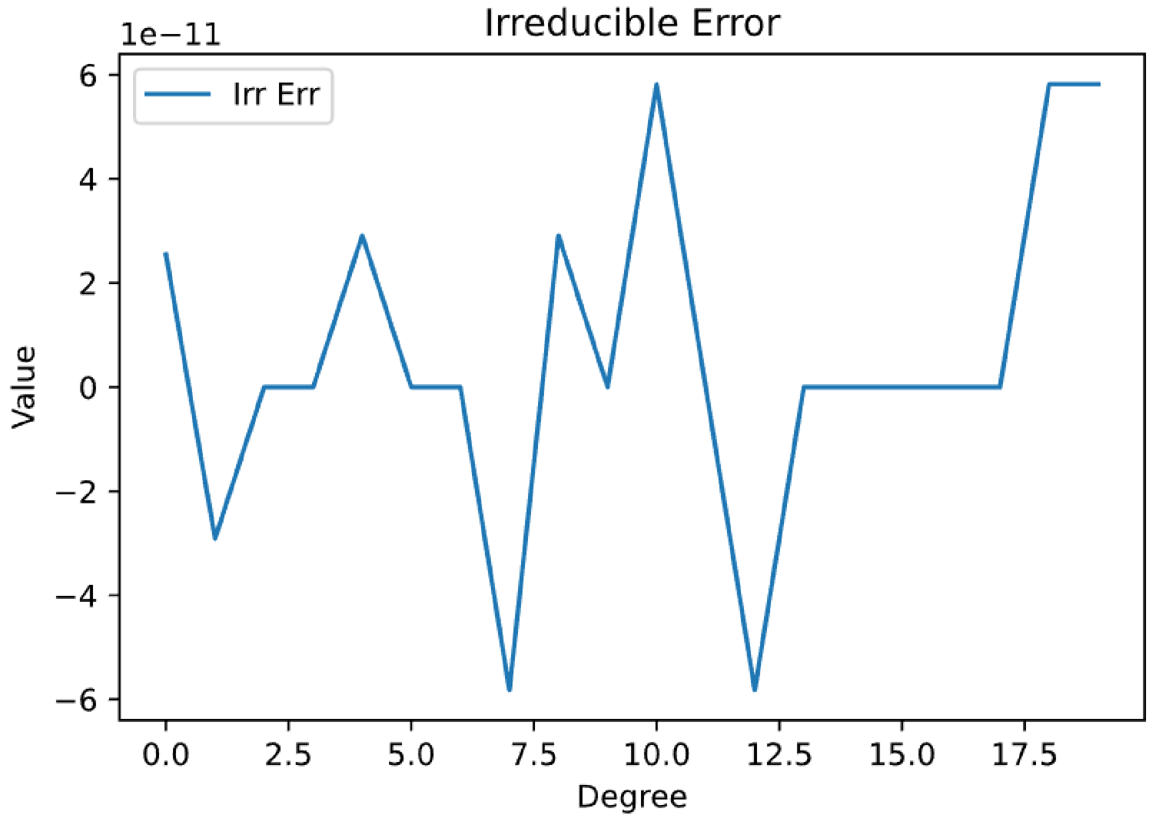
Calculating Irreducible Error

Irreducible error is a measure of the 'noise' in the supplied data. It is referred to as 'irreducible' as this error arises from the data and not the model and thus cannot be reduced no matter how good the created model is.

To calculate irreducible error, we used the formula:

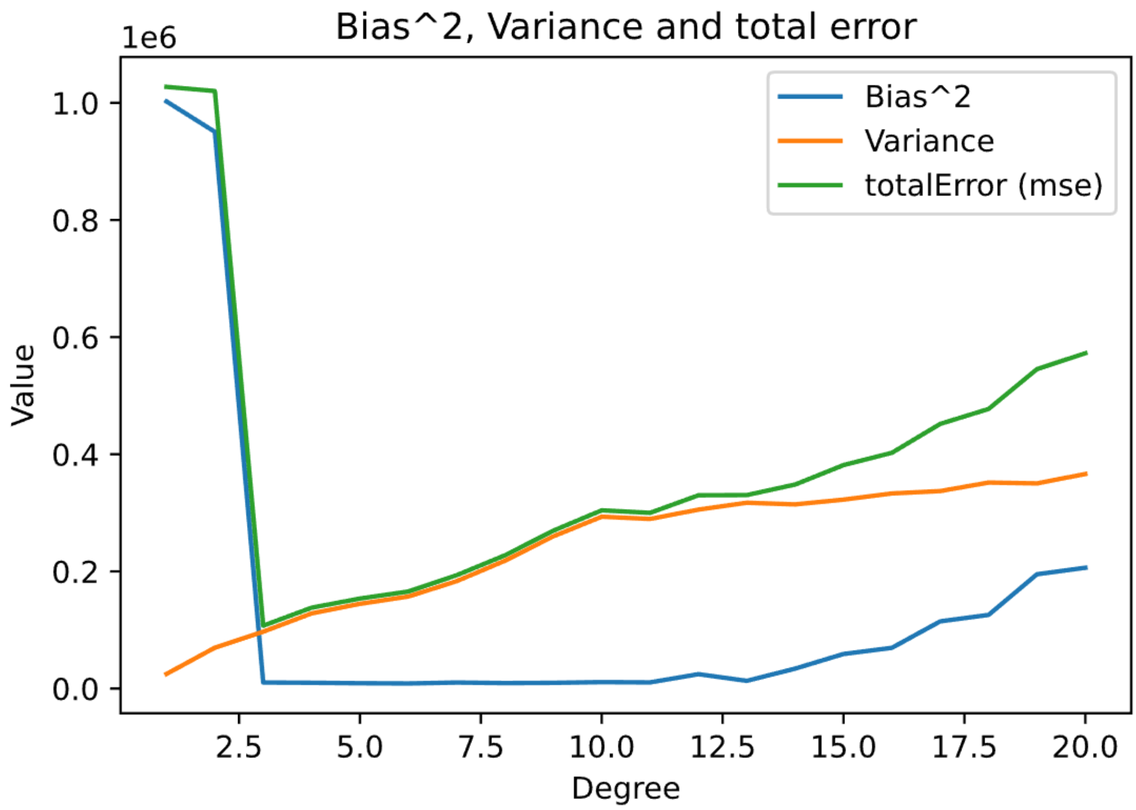
$$E[(f(x) - \hat{f}(x))^2] = Bias^2 + \sigma^2 + Variance$$
$$\sigma^2 = E[(f(x) - \hat{f}(x))^2] - (Bias^2 + Variance)$$

Plotting irreducible error



Task 4:

Plotting Bias^2 - Variance graph



Understanding the graph:

In this graph we display three different values: $Bias^2$, Variance and Total Error. Total error, as the name suggests, represents the total of $Bias^2$, Variance, and Irreducible error. This is what we aim to minimise when optimising our model. We observe that as the 'degree' or the complexity of our model increases, the variance increases and the $bias^2$ decreases (in an ideal situation it would continuously decrease but here we observe an increase towards the end).

At lower degrees, the model fails to accurately represent the training data and the test data. This is due to the lower number of features not being able to fully represent the data being provided. Due to this we observe a high bias. The low observed variance is due to the model being consistent but inaccurate which leads to a lower variance but a higher bias.

At higher degrees the model 'overfits', i.e. the model represents the training data very accurately but with the loss of generality. This results in the model performing poorly on test data or any data other than the data it was trained with. This leads to a very low bias as it very closely represents the training data but an high variance as it fails to remain consistent due to the differences between different training sets.

Tabulating the results:

degree	bias	variance
1	1001.39	22550.4
2	978.315	37427.8
3	93.0622	40282.1
4	89.8518	44151
5	87.3383	49296.9
6	86.417	59239.7
7	94.5778	89097.7
8	100.216	100876
9	95.924	124155
10	107.308	134876
11	99.785	142946
12	152.416	150638
13	117.411	153307
14	183.001	147355
15	241.782	146984
16	260.298	151108
17	337.601	150028
18	351.453	155541
19	440.471	155380
20	451.163	161849