# MDL Assignment 1

February 6, 2021


- Dayitva Goel: 2019101005
- Prajneya Kumar: 2019114011


## 1 Task 1: Linear Regression

***LinearRegression.fit()*** is a function provided by the library sklearn that fits a linear model to minimize the residual sum of squares between the observed points in a given dataset and the points which are predicted by the approximation.

In other words, the function tries to find a straight line, so that the Euclidean distance between the line and the points is minimized over the entire dataset. This line is also known as the line of best fit, hence the the name of the function.

This is mathematically denoted as:

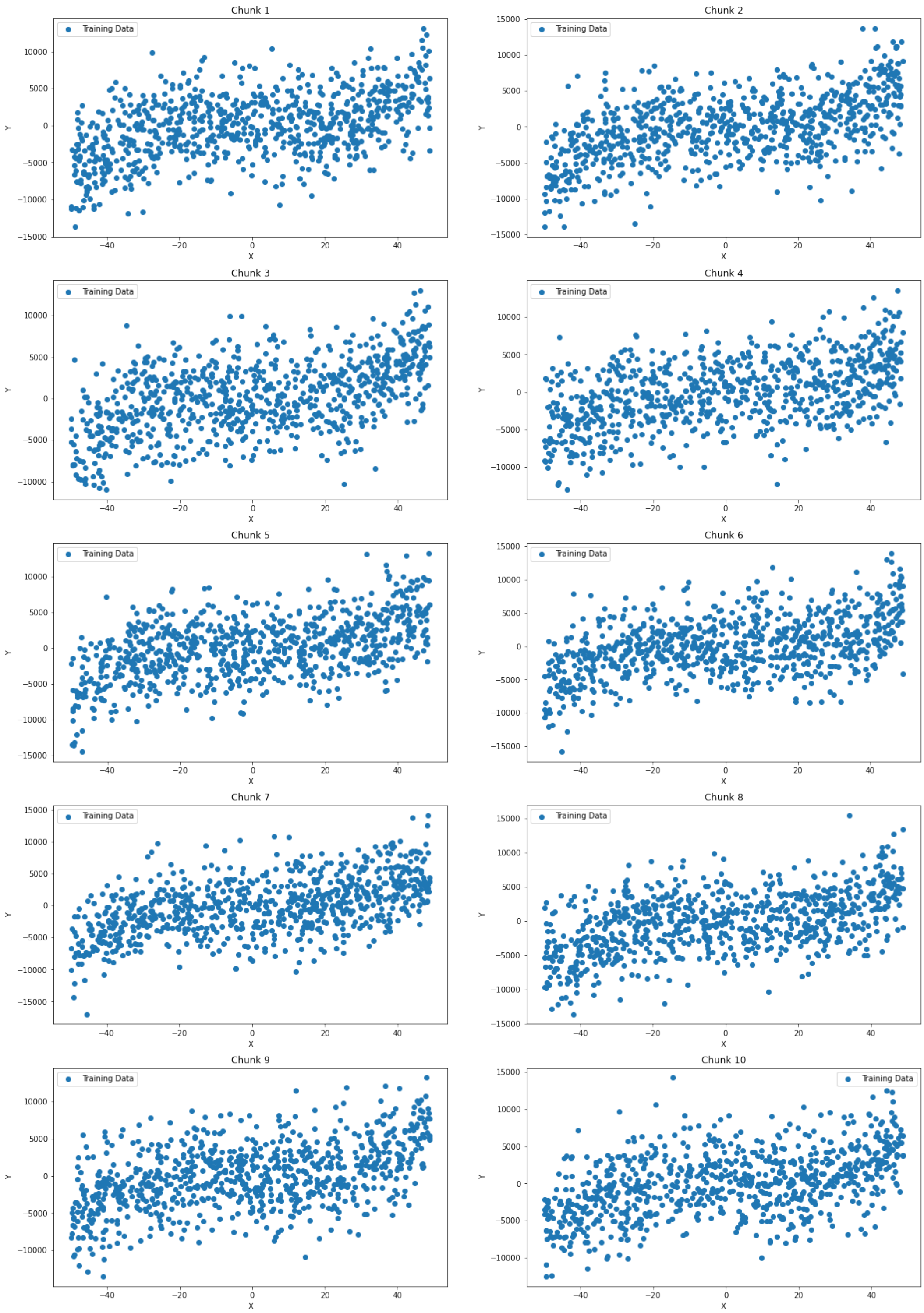
$$\min_w ||Xw - y||_2^2$$

where $w = (w_1, w_2, ...w_n)$ are the coefficient of the given linear model.

# 2 Task 2: Calculating Bias and Variance

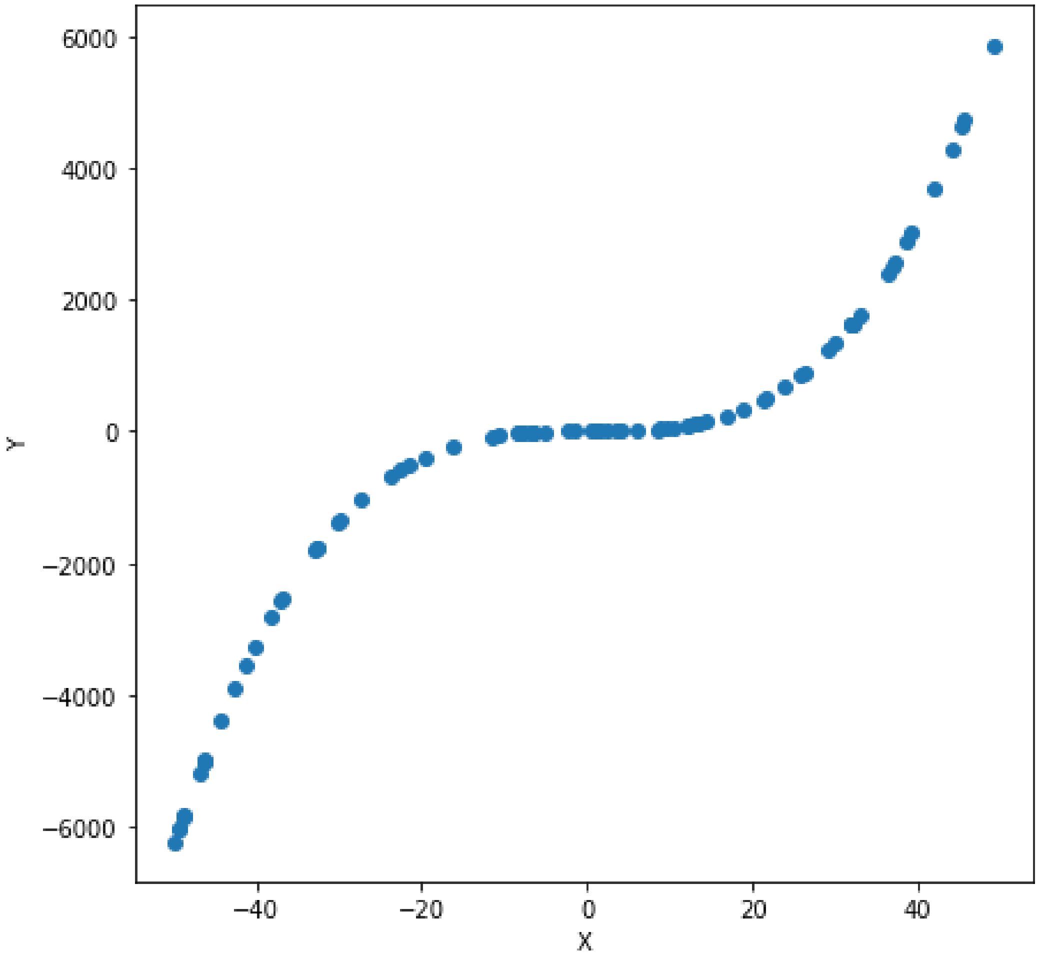
## Training Data Visualisation

We first load and visualise the training dataset to ensure that it is in an usable format:



### Test Data Visualisation

We then load and visualise the test dataset to ensure that it is in an usable format:



## Tabulation of Values

The following table contains the data for bias and variance for different function classes:

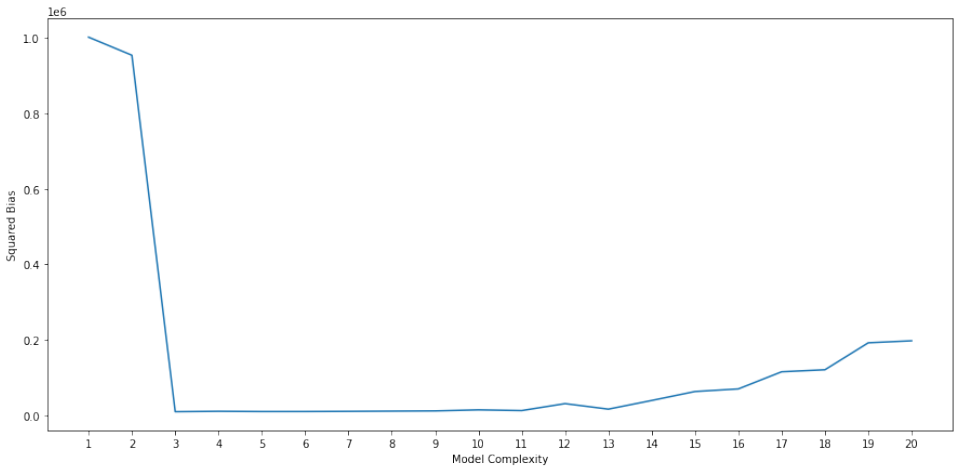
| Function Classes (Degree) | Bias | Variance |
|---|---|---|
| 1 | 819.7174357033888 | 25999.093009987868 |
| 2 | 810.763391396119 | 39105.83381326804 |
| 3 | 68.51085124220285 | 56095.89320972663 |
| 4 | 81.33971112338618 | 114907.29152950706 |
| 5 | 78.95841418892078 | 151434.02790058558 |
| 6 | 78.36480094220735 | 174226.74500308032 |
| 7 | 86.91619731151918 | 198849.50274648867 |
| 8 | 90.32513553789045 | 221555.66219639863 |
| 9 | 92.35486444098777 | 232275.80526448222 |
| 10 | 97.94987666578288 | 232807.77103703268 |
| 11 | 91.1035858738162 | 238575.67800096897 |
| 12 | 125.84095426141899 | 219780.32854004213 |
| 13 | 92.69346955553092 | 234838.9842927035 |
| 14 | 130.1744389208196 | 212545.22546633697 |
| 15 | 166.459419192714 | 221715.204658274 |
| 16 | 170.41737008473348 | 239357.37249339762 |
| 17 | 236.71466071433693 | 242993.45565667856 |
| 18 | 239.124904302733 | 269051.60410073324 |
| 19 | 304.8683524243953 | 270102.32868061244 |
| 20 | 305.4448944260228 | 299012.33395130053 |

## Degree of the polynomial versus bias and variance

### Bias

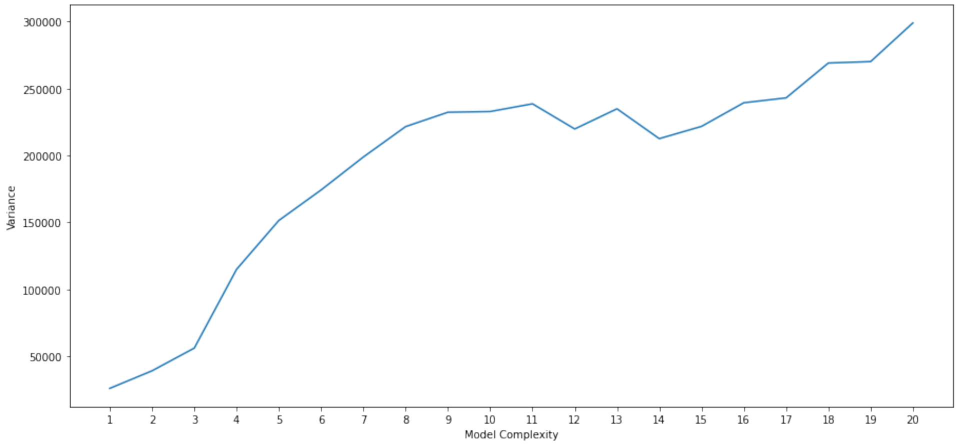
Clearly, when we run our code on all the models with all the function classes, what we observe is that when the degree of the chosen polynomial is low then the bias error is high. This bias error decreases as the degree of chosen polynomial increases. However, the bias error again starts on increasing as we move towards higher degree polynomials.

We can see from the values the the bias is lowest at $degree = 3$, thus hinting at the fact that the data is best modelled by a degree 3 hypothesis.




### Variance

The variance on the lower function classes is low, and begins to increase as we keep on increasing the degree of the chosen polynomial. This general increase in variance is because the model becomes highly susceptible to minor variations in the training datasets in higher function classes.




# 3 Task 3: Calculating Irreducible Error

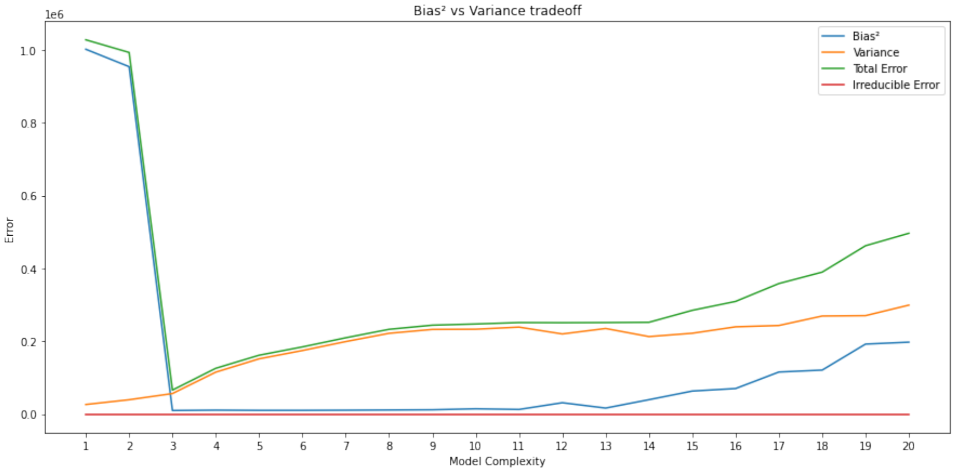
The following table contains the data for irreducible error for different function classes:

| **Function Classes (Degree)** | **Irreducible Error** |
|---|---|
| 1 | 1.16415322e-10 |
| 2 | 3.49245965e-10 |
| 3 | -8.73114914e-11 |
| 4 | -1.16415322e-10 |
| 5 | -8.73114914e-10 |
| 6 | 1.74622983e-10 |
| 7 | -6.69388101e-10 |
| 8 | 1.74622983e-10 |
| 9 | -5.82076609e-10 |
| 10 | 0.00000000e+00 |
| 11 | 2.32830644e-10 |
| 12 | 6.40284270e-10 |
| 13 | 3.20142135e-10 |
| 14 | 6.40284270e-10 |
| 15 | -1.74622983e-10 |
| 16 | -1.74622983e-10 |
| 17 | 0.00000000e+00 |
| 18 | -2.91038305e-10 |
| 19 | 1.45519152e-09 |
| 20 | -4.65661287e-10 |

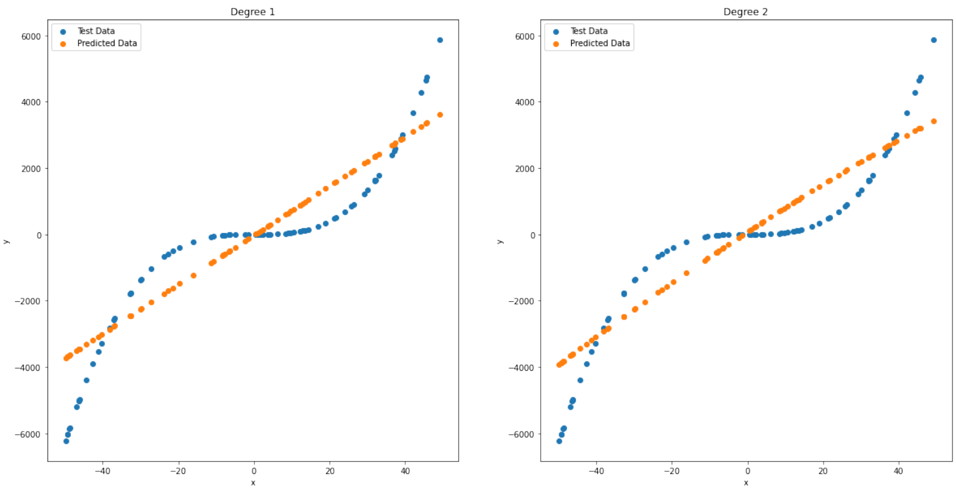
Again, as it can been clearly seen from the above table, the irreducible error remains almost constant throughout. This happens because this error is irrelevant of the underlying model and has to do with the inherent noise in the problem itself.

The value of irreducible error is found to be very close to 0. This means that the data provided to us contains very less noise (noise here refers to data precision, accuracy, other factors over which we cannot have any control whatsoever)

# 4 Task 4: Plotting $Bias^2$ $Variance$ graph

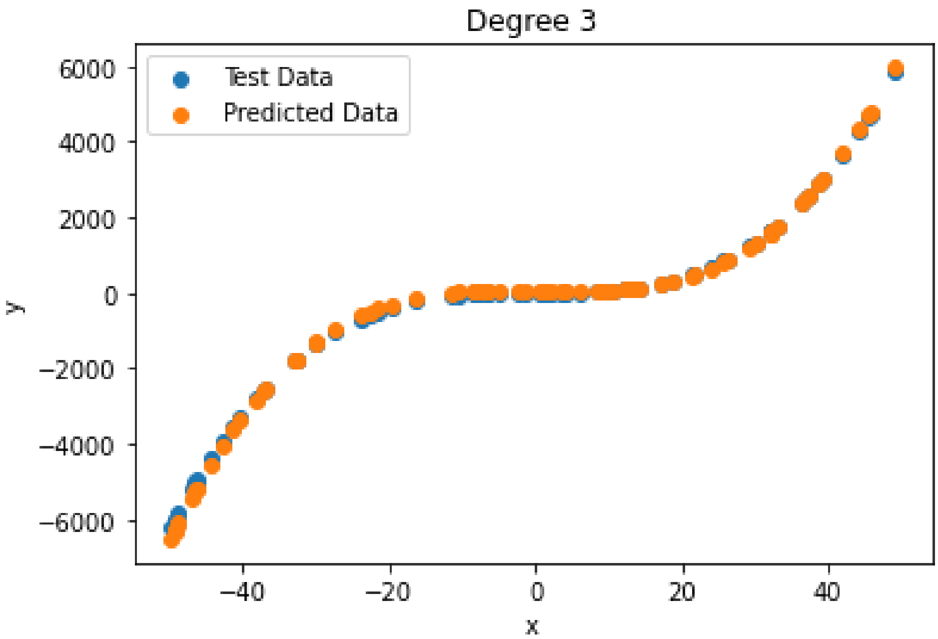



For $degree = 1, 2$ :




**Underfitting** is the case when a statistical model is not complex enough to accurately capture all the relationships between the dataset's target features and variables. From the $bias^2 - variance$ graph, we can see that in this area the bias is very high and the variance is very low.

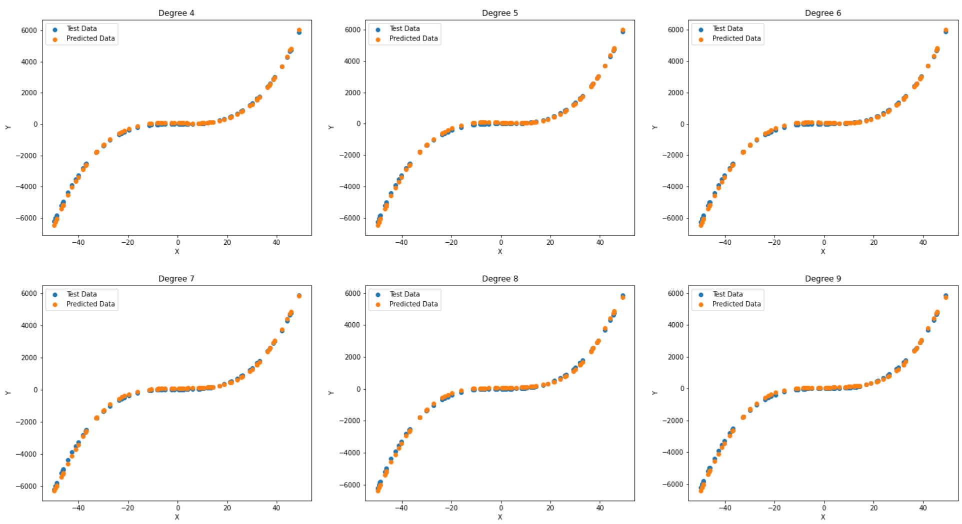
For $degree = 3$, the model behaves ideally and plots a graph as follows:

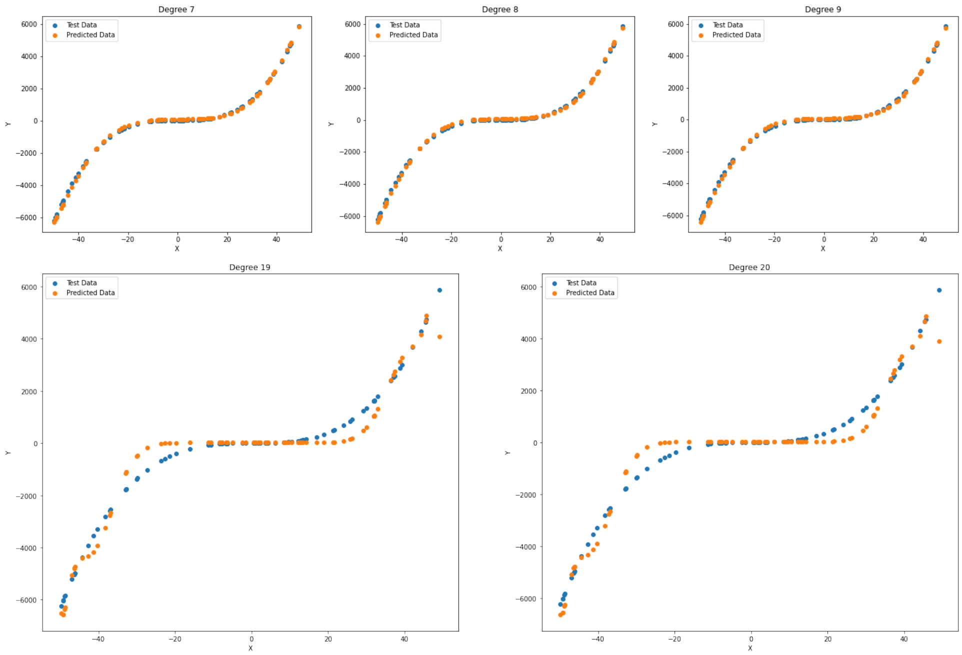



This is the ideal case for our model, and the true approximation for our function, mostly because the true function is actually cubic. The function can be said to neither underfit nor overfit. We can see from the $bias^2 - variance$ graph that the bias drastically drops here, and the variance is low too at the same time.

**Overfitting** is the case when a statistical model is so complex and memorized towards a particular to a dataset that it cannot be used for any other model. For higher function classes, our model starts suffering from high variance due to its inability to generalize well beyond the training data, our model **overfits**.

For $degree > 4$ plots:






# 5 Conclusion

From the graphs, it is evident that an increase in degree after 3 does not bring much benefit in terms of bias, but just increases the variance. Thus the function class of 3 is the best suited model for this problem.