ML BOOTCAMP PROJECT REPORT

Winter of Code 7.0

AKARSH JAISWAL

24JE0881

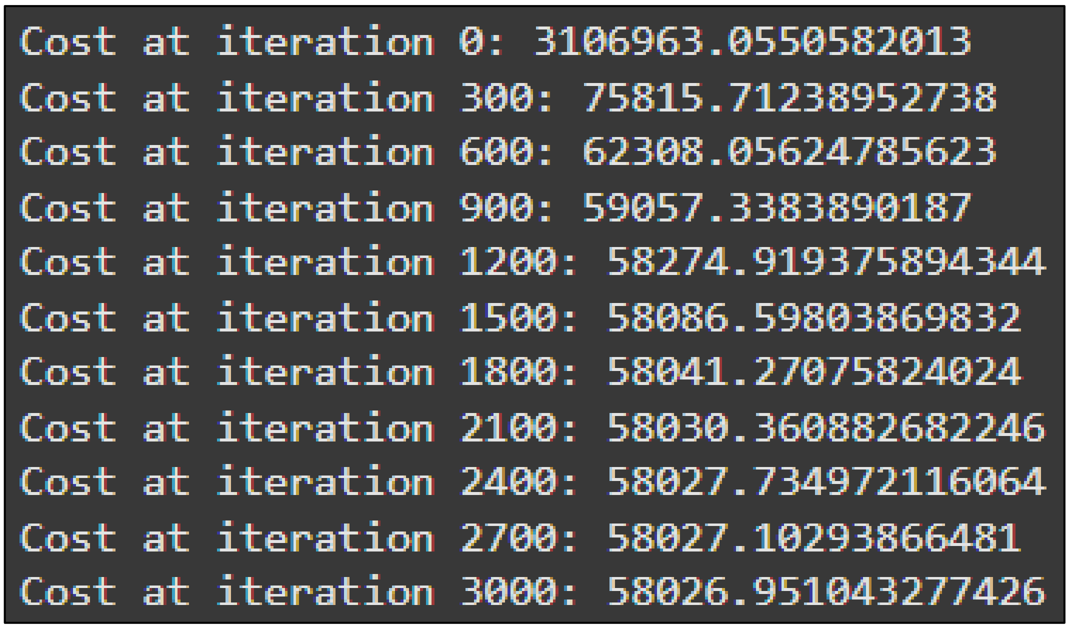
Project Report:

Implementation of linear regression, polynomial regression, logistic regression, softmax regression, K-Means Clustering and N-layer Neural Network with the help of numpy, pandas and matplotlib libraries only.

Linear Regression:

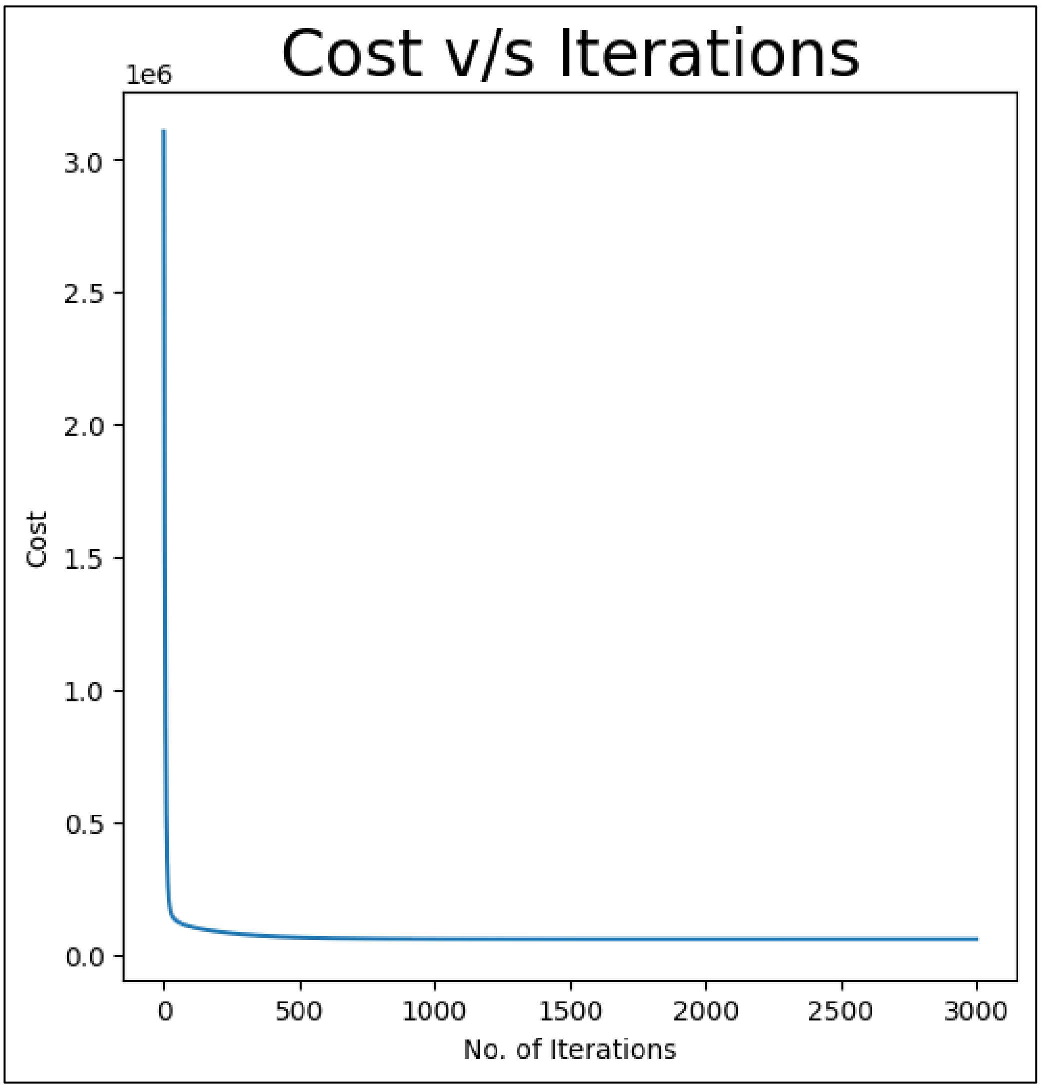
A linear regression model was employed for prediction, seeking to establish a relationship between independent and dependent variables. The dataset was imported from a CSV file having 25 features and 48000 training examples using the pandas library. After that I normalised the data to make each feature in a particular range by Z-score normalisation which also increased fast cost calculation and hurried convergence.

The dataset was split into 2 sets, one having 42000 training examples to train the model and the other having 6000 sets of training example to assess its generalization performance. The cost at various iteration values was calculated, and the results were printed at specific intervals. Different values of the hyperparameter alpha and no. of iterations were tested to enhance the model's accuracy.

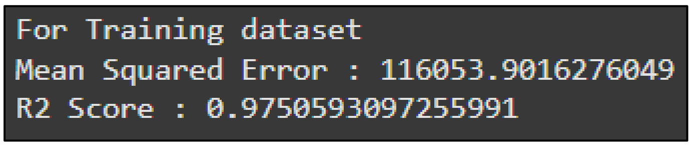


```
Cost at iteration 0: 3106963.0550582013
Cost at iteration 300: 75815.71238952738
Cost at iteration 600: 62308.05624785623
Cost at iteration 900: 59057.3383890187
Cost at iteration 1200: 58274.919375894344
Cost at iteration 1500: 58086.59803869832
Cost at iteration 1800: 58041.27075824024
Cost at iteration 2100: 58030.360882682246
Cost at iteration 2400: 58027.734972116064
Cost at iteration 2700: 58027.10293866481
Cost at iteration 3000: 58026.951043277426
```

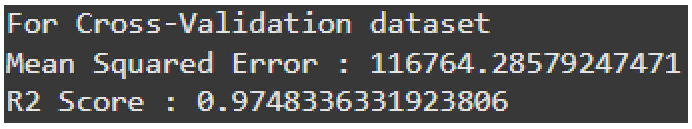
A graph was plotted, illustrating the relationship between the cost (calculated using the final chosen alpha = 1.9) and the number of iterations = 3000.



Finally, I calculated Mean Squared Error and R2 Score to measure the accuracy of the model.

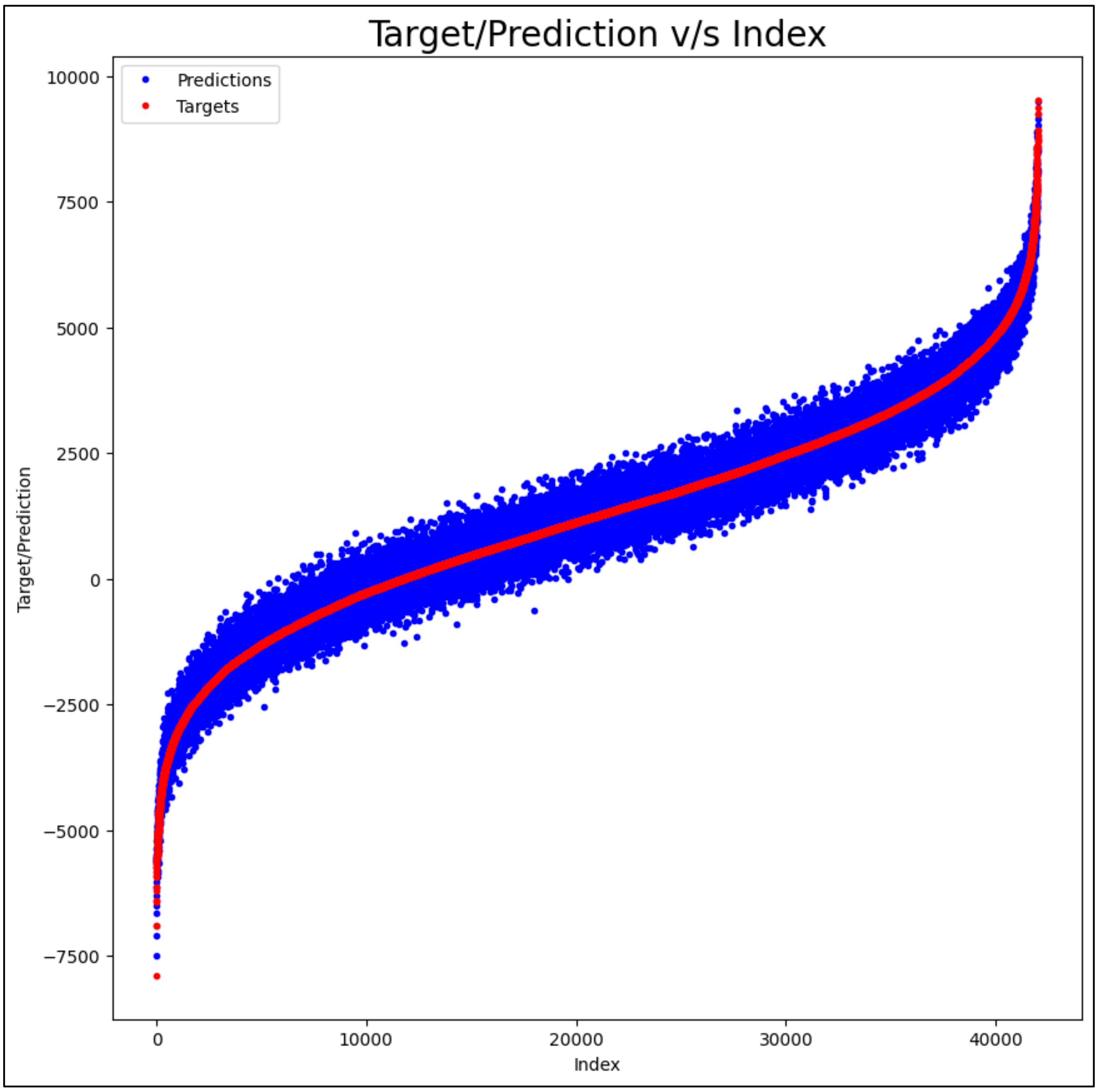


```
For Training dataset
Mean Squared Error : 116053.9016276049
R2 Score : 0.9750593097255991
```



```
For Cross-Validation dataset
Mean Squared Error : 116764.28579247471
R2 Score : 0.9748336331923806
```

I plotted a graph of Target/Prediction v/s Index to understand the model better.



Polynomial Regression:

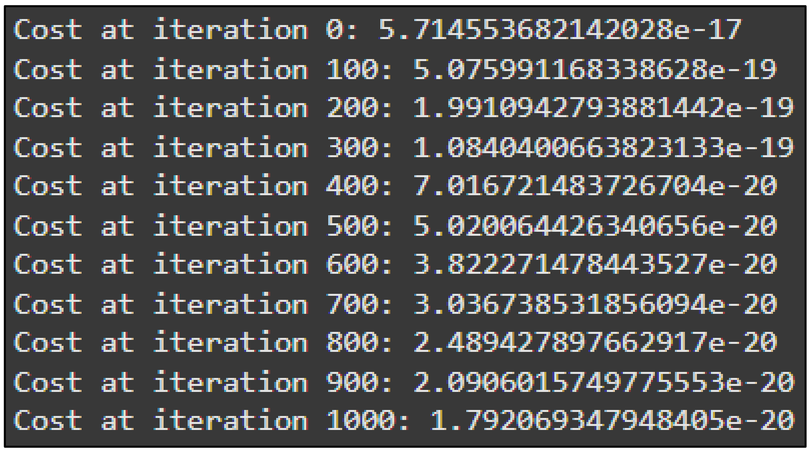
In this analysis, polynomial regression with feature engineering was employed using 5 existing features. Feature engineering was applied to create additional polynomial features, with degrees 1 to 8 by taking combination of each 5 features resulting in a varying number of new features. The cost at various iteration values was calculated, and results were monitored to identify the most suitable polynomial degree Data is normalised as before by Z score normalisation.

Again, the dataset was split into training, cross-validation and testing datasets in ratio 40000:4000:4000.

The sixth-degree polynomial works best when we consider both accuracy and efficiency. The chances of overfitting are fairly low because of the huge dataset. Hence, we will be using the sixth-degree model.

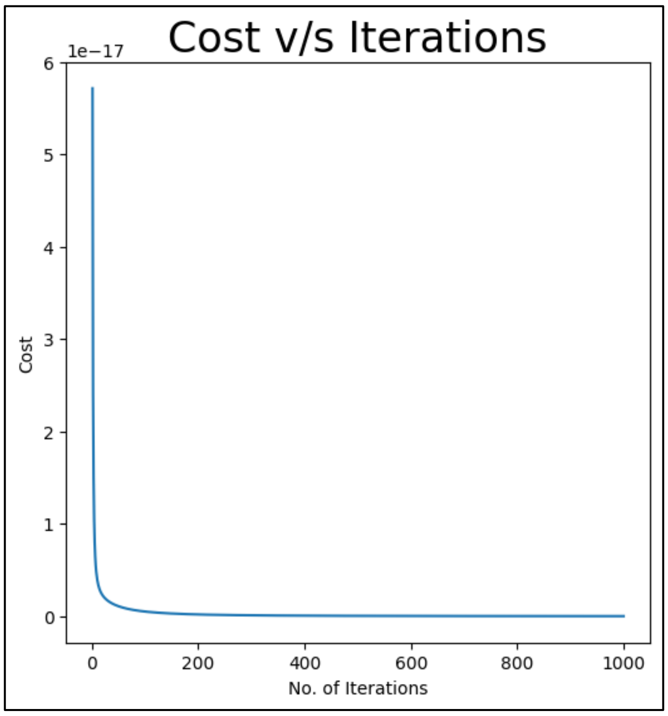
The R2 scores for training, cross-validation and testing datasets are 0.9996, 0.9972 and 0.9983 respectively.

The cost at various iteration values was calculated, and the results were printed at specific intervals. Different values of the hyperparameter alpha and no. of iterations were tested to enhance the model's accuracy.

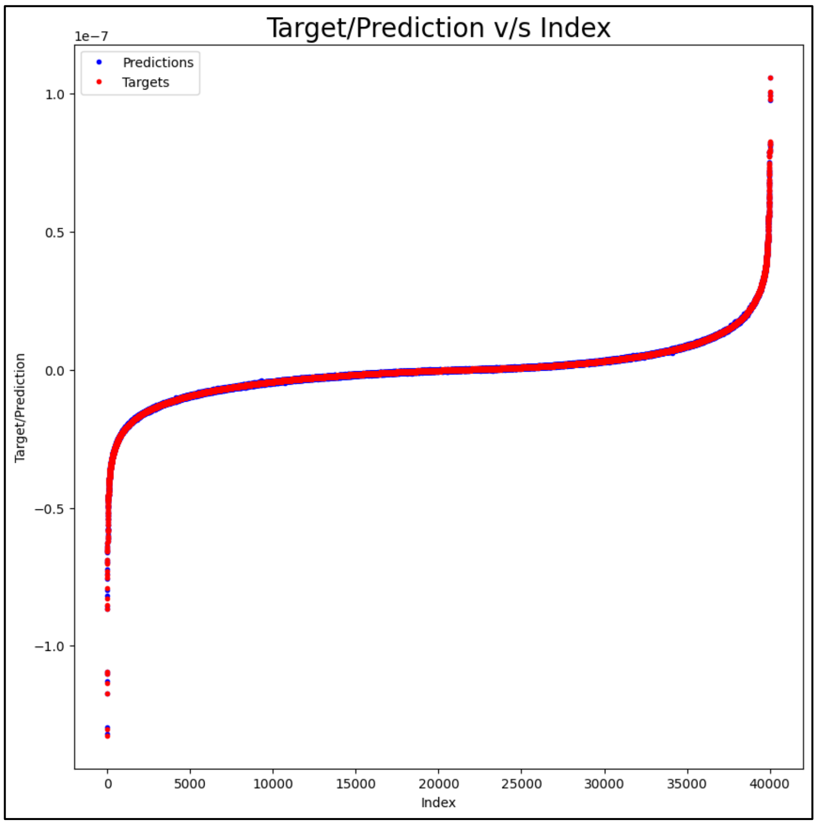


```
Cost at iteration 0: 5.714553682142028e-17
Cost at iteration 100: 5.075991168338628e-19
Cost at iteration 200: 1.9910942793881442e-19
Cost at iteration 300: 1.0840400663823133e-19
Cost at iteration 400: 7.016721483726704e-20
Cost at iteration 500: 5.020064426340656e-20
Cost at iteration 600: 3.822271478443527e-20
Cost at iteration 700: 3.036738531856094e-20
Cost at iteration 800: 2.489427897662917e-20
Cost at iteration 900: 2.0906015749775553e-20
Cost at iteration 1000: 1.792069347948405e-20
```

A graph was plotted, illustrating the relationship between the cost (calculated using the final chosen alpha = 0.12) and the number of iterations = 1000.



I plotted a graph of Target/Prediction v/s Index to understand the model better.



Logistic Regression:

A logistic regression model was employed for classification, aiming to predict the probability of each class based on independent variables. The dataset, containing 20 features and 48,000 training examples, was imported from a CSV file using the pandas library. The data was normalized using Z-score normalization to ensure all features had a mean of 0 and a standard deviation of 1, which facilitated faster cost computation and accelerated convergence during optimization.

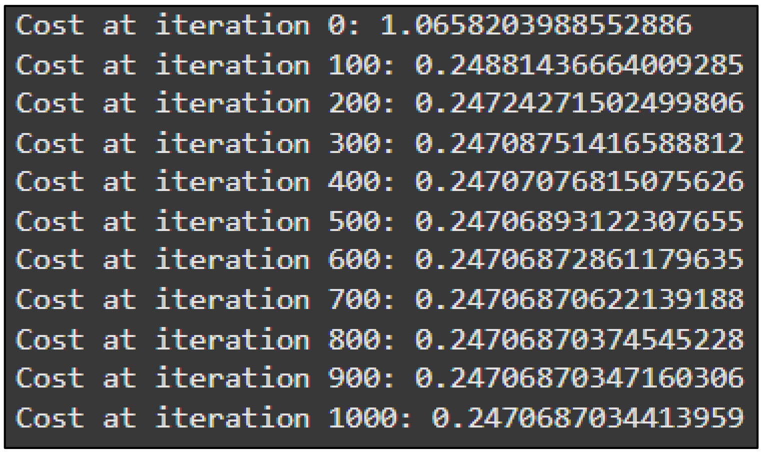
The dataset was split into training and cross-validation datasets in ratio 44000:4000.

The binary classification error and F1 score were 0.9345 and 0.893 respectively.

I also did some feature engineering to generate features of degree 2 and then carry out logistic regression but I came with a huge time consumption with little to no improvement. The F1 score for degree 2 features was 0.9029.

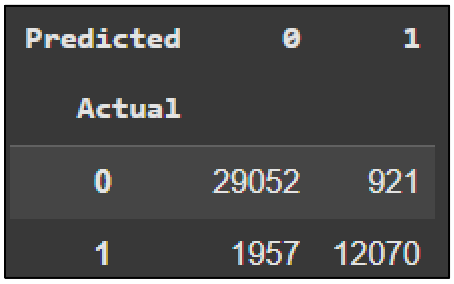
Finally, the binary classification error and F1 score for cross-validation dataset were 0.93675 and 0.896.

The cost at various iteration values was calculated, and the results were printed at specific intervals. Different values of the hyperparameter alpha and no. of iterations were tested to enhance the model's accuracy.



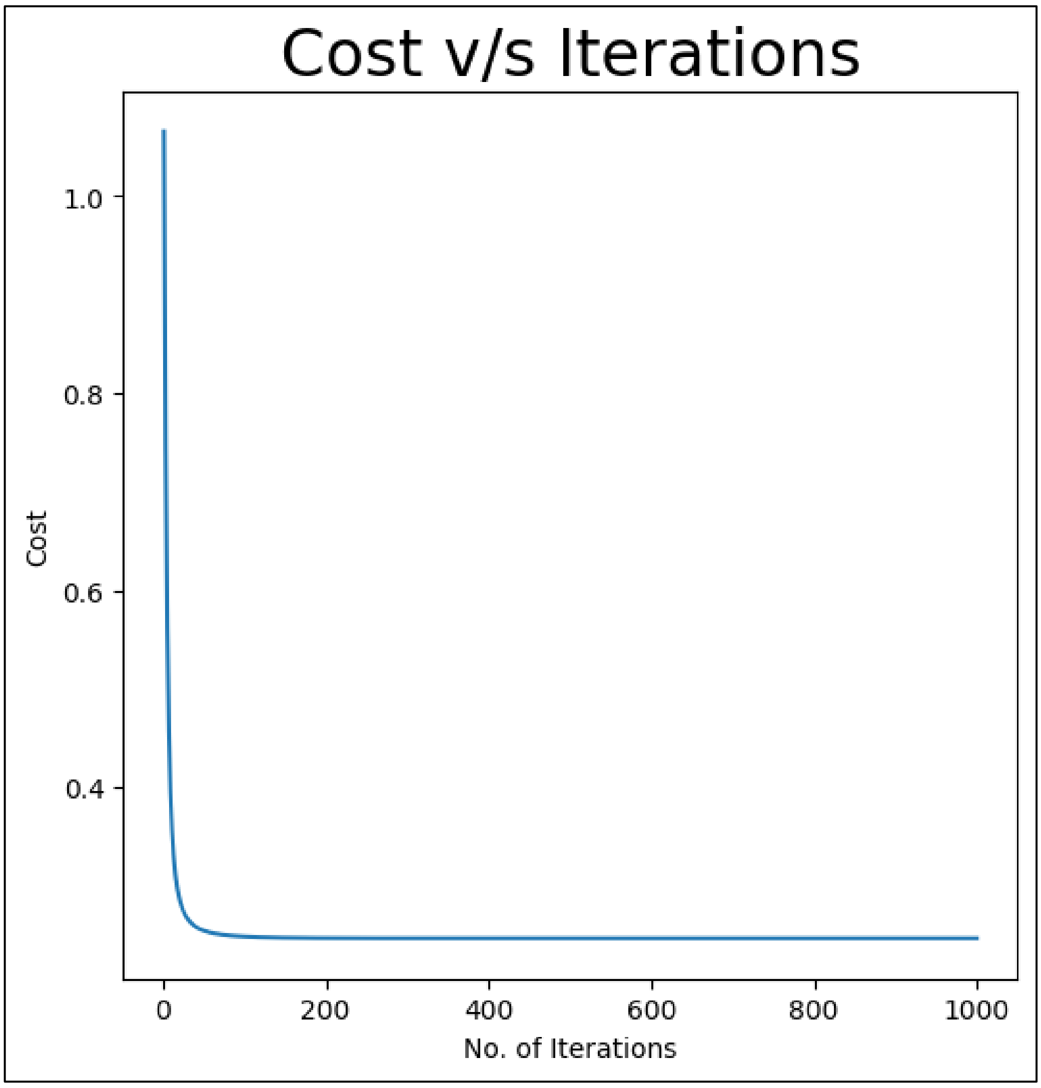
```
Cost at iteration 0: 1.0658203988552886
Cost at iteration 100: 0.24881436664009285
Cost at iteration 200: 0.24724271502499806
Cost at iteration 300: 0.24708751416588812
Cost at iteration 400: 0.24707076815075626
Cost at iteration 500: 0.24706893122307655
Cost at iteration 600: 0.24706872861179635
Cost at iteration 700: 0.24706870622139188
Cost at iteration 800: 0.24706870374545228
Cost at iteration 900: 0.24706870347160306
Cost at iteration 1000: 0.2470687034413959
```

A confusion matrix was also made to understand the model better.



Predicted	0	1
Actual		
0	29052	921
1	1957	12070

A graph was plotted, illustrating the relationship between the cost (calculated using the final chosen alpha = 0.5) and the number of iterations = 1000.



Softmax Regression:

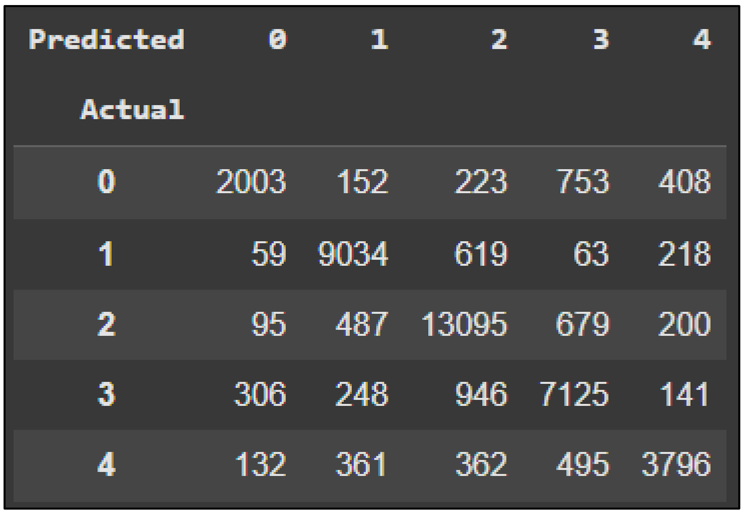
A softmax regression model was employed for multi-class classification, aiming to predict the probabilities of each class across multiple categories. The dataset, comprising 20 features and 48,000 training examples, was imported from a CSV file using the pandas library. Z-score normalization was applied to scale the data, ensuring each feature had a mean of 0 and a standard deviation of 1, which improved numerical stability, expedited cost calculations, and enhanced convergence during training.

The dataset was split into training and cross-validation datasets in ratio 44000:4000.

The multi-class classification error and micro F1 score for the training dataset were 0.8346 and 0.8346 respectively.

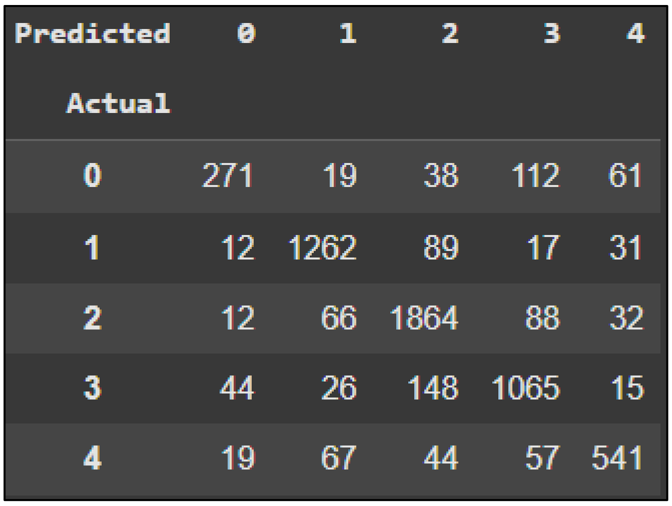
The multi-class classification error and micro F1 score for the cross-validation dataset were 0.8338 and 0.8338 respectively.

The confusion matrix for the training dataset was:



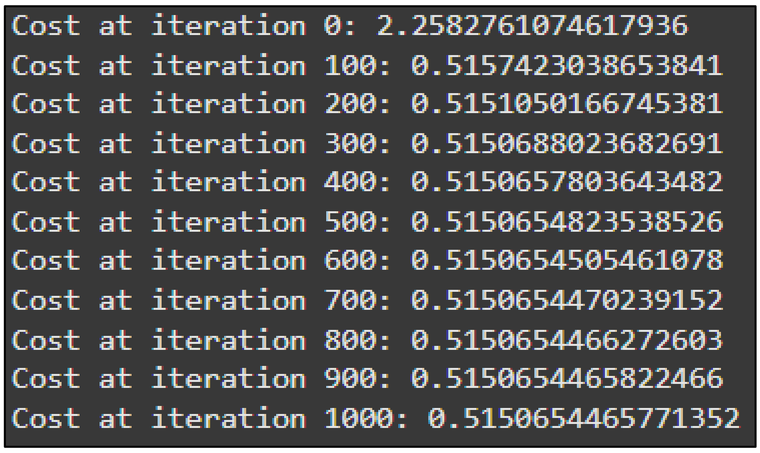
Predicted	0	1	2	3	4
Actual					
0	2003	152	223	753	408
1	59	9034	619	63	218
2	95	487	13095	679	200
3	306	248	946	7125	141
4	132	361	362	495	3796

The confusion matrix for the cross-validation dataset was:



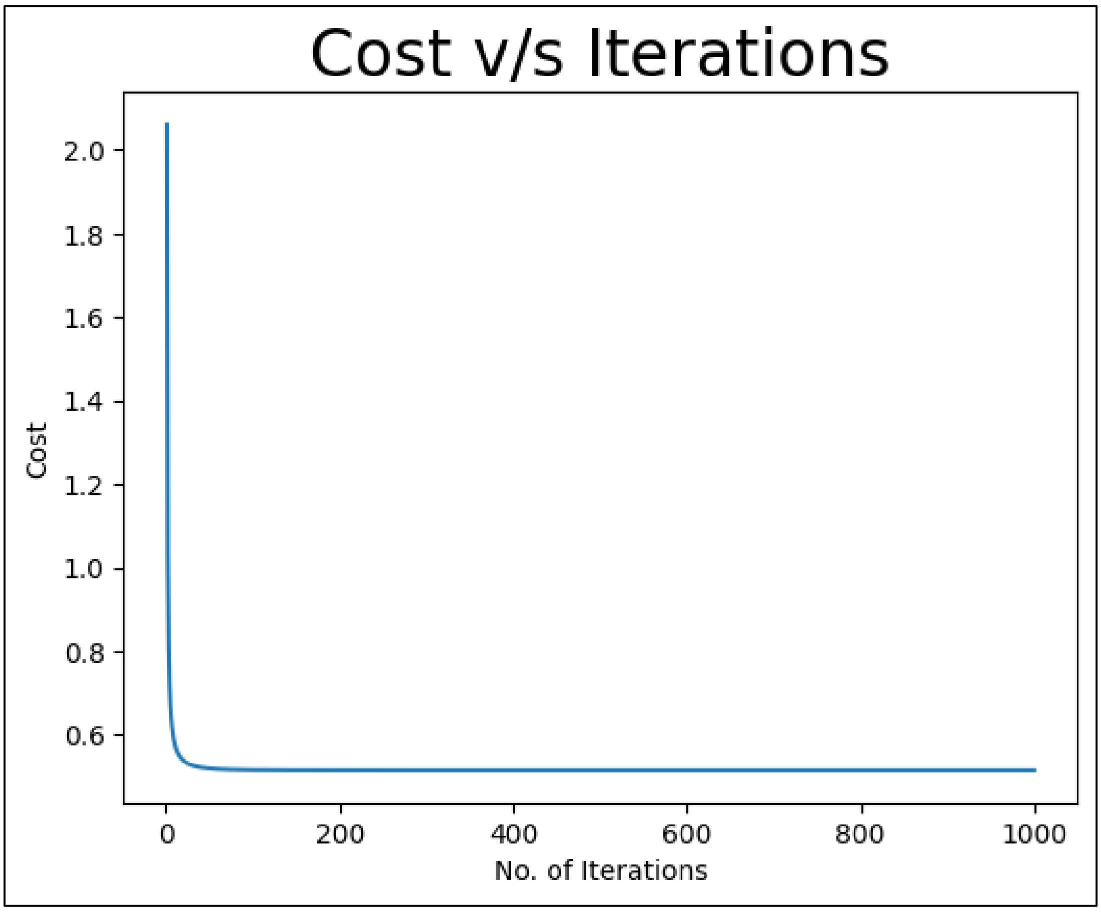
Predicted	0	1	2	3	4
Actual					
0	271	19	38	112	61
1	12	1262	89	17	31
2	12	66	1864	88	32
3	44	26	148	1065	15
4	19	67	44	57	541

The cost at various iteration values was calculated, and the results were printed at specific intervals. Different values of the hyperparameter alpha and no. of iterations were tested to enhance the model's accuracy.



```
Cost at iteration 0: 2.2582761074617936
Cost at iteration 100: 0.5157423038653841
Cost at iteration 200: 0.5151050166745381
Cost at iteration 300: 0.5150688023682691
Cost at iteration 400: 0.5150657803643482
Cost at iteration 500: 0.5150654823538526
Cost at iteration 600: 0.5150654505461078
Cost at iteration 700: 0.5150654470239152
Cost at iteration 800: 0.5150654466272603
Cost at iteration 900: 0.5150654465822466
Cost at iteration 1000: 0.5150654465771352
```

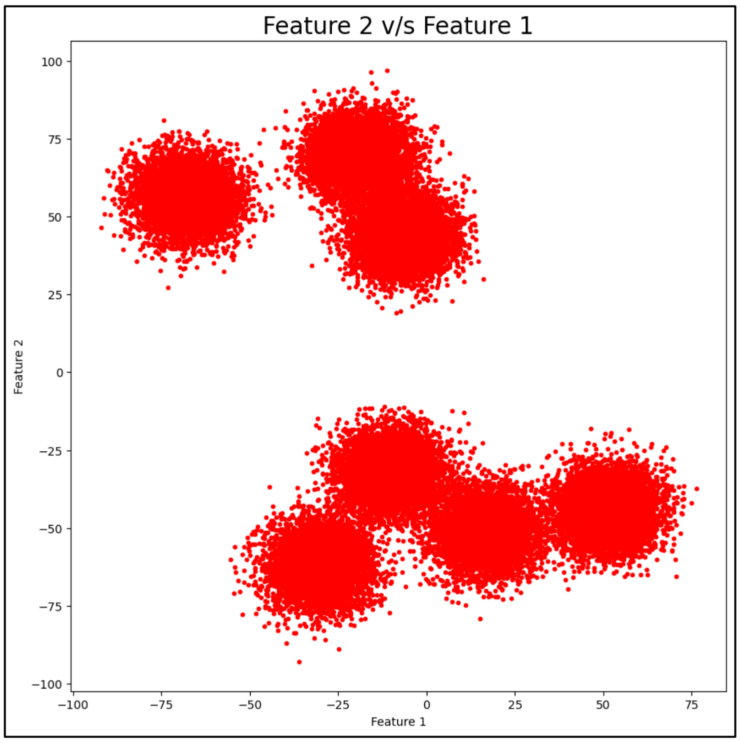
A graph was plotted, illustrating the relationship between the cost (calculated using the final chosen alpha = 1.5) and the number of iterations = 1000.



K-Means Clustering:

A k-means clustering algorithm was utilized for unsupervised learning, aiming to group the dataset into distinct clusters based on feature similarity. The dataset, containing 6 features and 40,000 samples, was imported from a CSV file using the pandas library.

I first plotted the data along all possible pairs of features to get an idea of the number of clusters and to find the best pair to represent the data. A pair which showed a lot of variance was (feature_2, feature_1).



The number of clusters were varied from 2 to 10 and their costs were recorded.

For each value of the number of clusters, we chose the best of 30 random initializations to ensure we have a global minimum for each instead of a local minimum.

The cost for each value of number of clusters is listed from 2 to 10 is listed here:

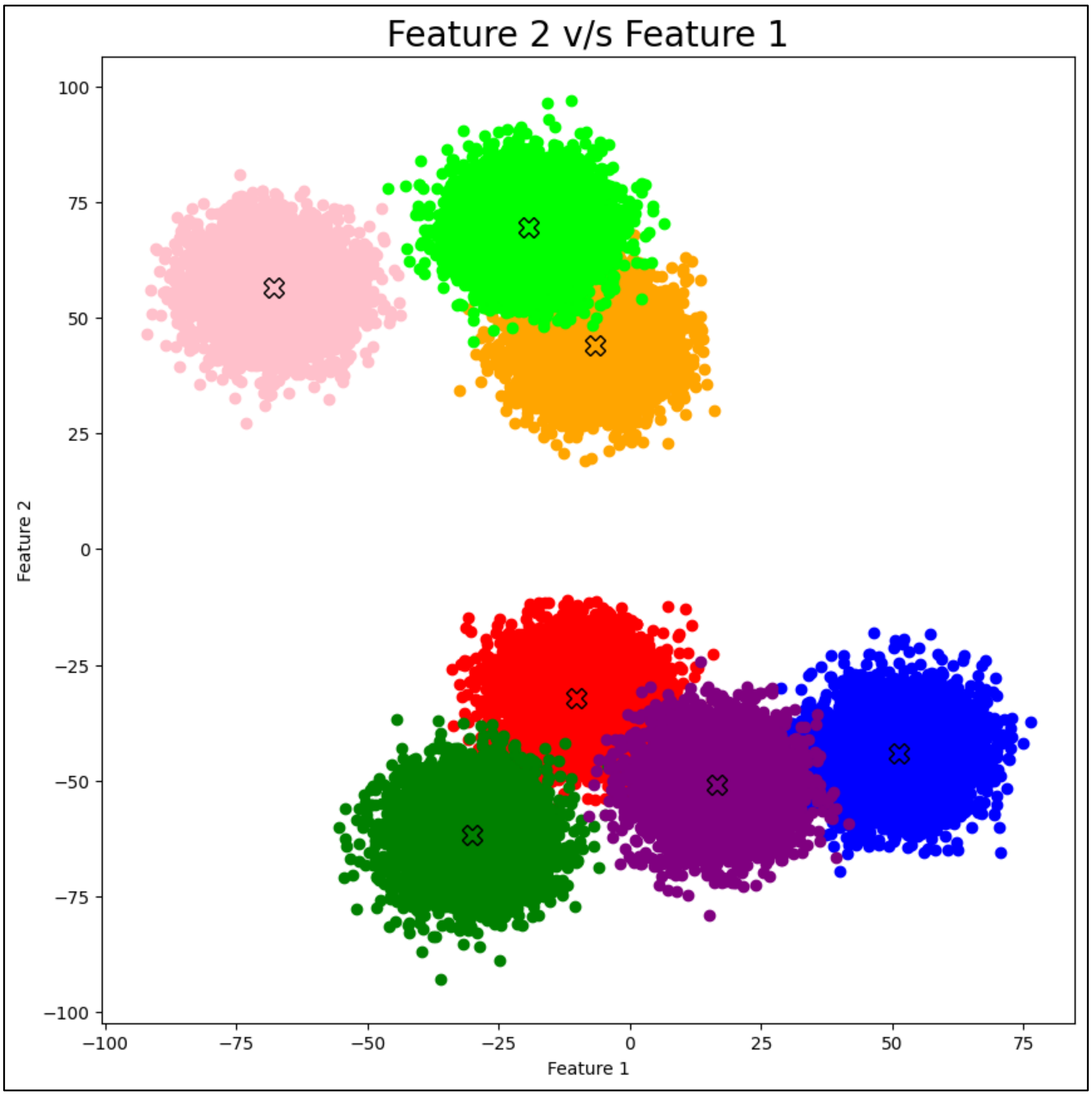
8105.989827442513, 4925.311450065913, 2559.254957914058,

1515.0571262156063, 624.4838337935834, 293.74261334088044,

288.9623614389889, 284.2979559391002, 279.73879743896646

After observing the costs and graphs, we come to the conclusion that 7 clusters is the best for this dataset.

Hence, the final graph is:



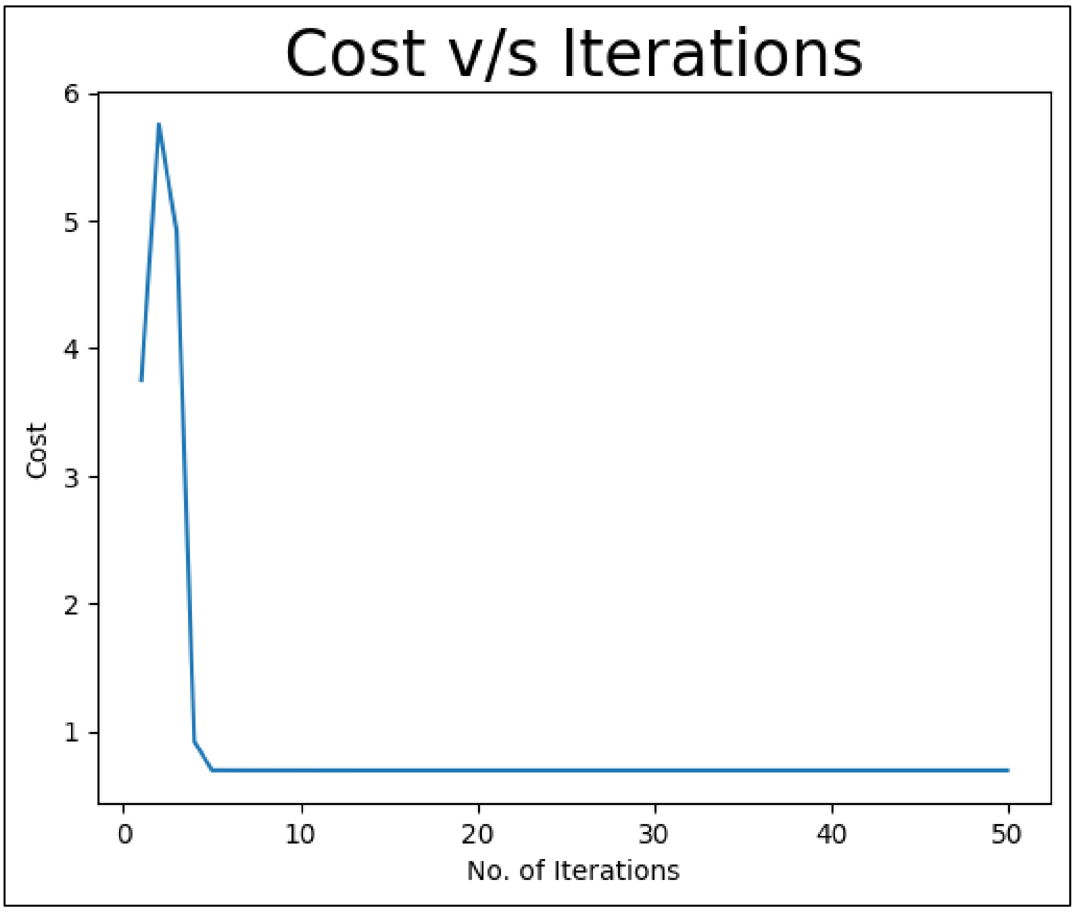
N-Layered Neural Network:

The dataset is loaded from a CSV file containing features and target values and converted into NumPy arrays and scaled numerical features by dividing with max value that is 255.

Then I used Z-score normalization to normalize the data.

I used a 3-layer neural network with units (128, 32, 1) to predict the binary labels.

A graph was plotted, illustrating the relationship between the cost (calculated using the final chosen alpha = 0.9) and the number of iterations = 50.


Cost v/s Iterations
Cost
No. of Iterations