SMAI Assignment-3 Report

Name: Akshat Maheshwari (20161024)

Question 1:

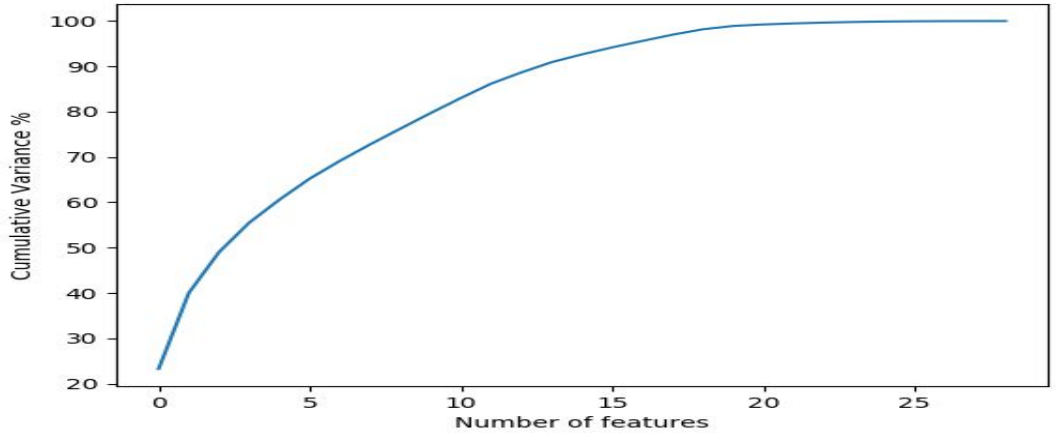
PCA (Principal Component Analysis):

Shape of original dataset: (24998, 29)
Shape of dataset after applying PCA: (24998, 13)

Now, in the PCA algorithm, I have reduced the dimensions of the dataset depending on the eigenvectors and eigenvalues. Now, as given in the question, we had to keep a tolerance of 10%, so that is how I got to select the number of features to be selected after dimensionality reduction (which is 13 here).

I calculated the eigenvalues and eigenvectors using the ***numpy.linalg.eig*** function of the numpy library. After that, I reverse sorted the eigenvalues and took the first 13 out of those values, and their corresponding eigenvectors.

The plot obtained that helped in the selection of the number of features is:



K-Means:

I implemented the K-means algorithm using the iterative method, and I have kept the maximum number of iterations as 10.

Maximum purity for the complete data: 0.8291863349067925

GMM:

I implemented the GMM clustering using the sklearn GaussianMixture model using the following function:

gmm = GaussianMixture(n_components=5, covariance_type='full', max_iter=10000, init_params='kmeans')

Purity array obtained: [0.8144523899134362, 0.9996699669966996, 0.8889329107237189, 0.3798808735936466, 1.0]
Purity for total data: 0.7776622129770382

Hierarchical Clustering:

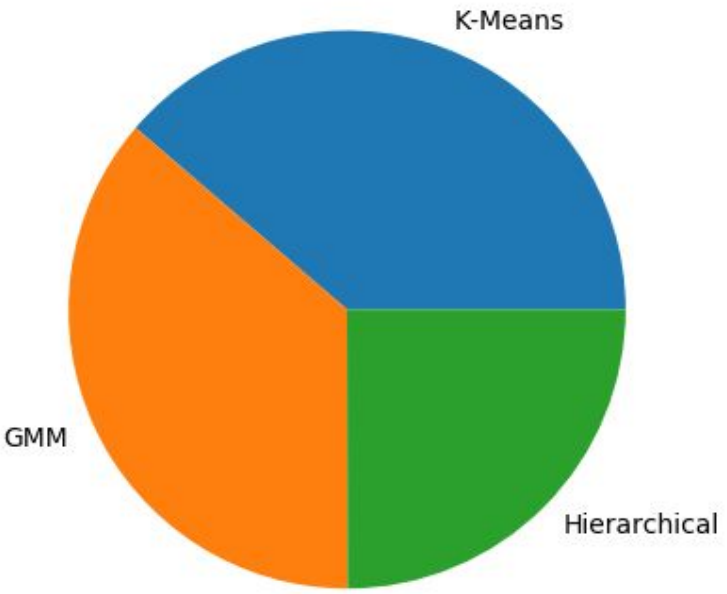
I implemented the Hierarchical clustering using the sklearn AgglomerativeClustering function:

Hierarchy = AgglomerativeClustering(n_clusters=5, affinity='euclidean', linkage='single')

Purity array obtained: [0.5345496739086945, 1.0, 1.0, 1.0, 1.0]
Purity for total data: 0.534624154898587 8

My device was giving Memory Error, so I have added this link to my google colab notebook: https://colab.research.google.com/drive/1HD-LIoVvHTkvszMwinJ2JztNVIA9DMt7

The pie chart for the purities of the different classifiers is:



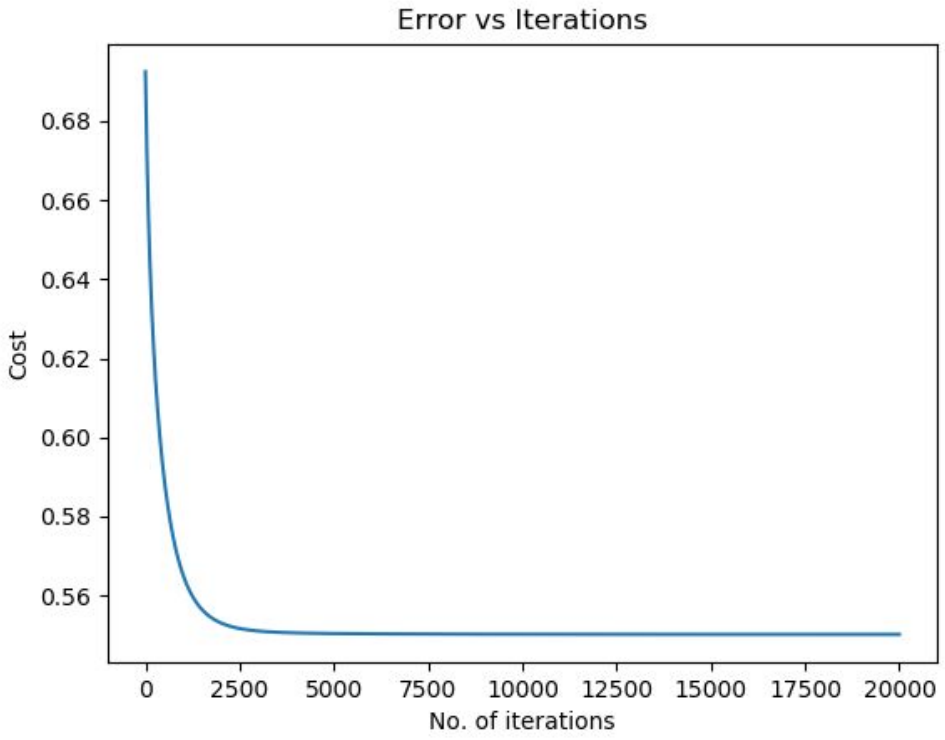
PCA for Categorical Data:

PCA is designed for continuous variables. It tries to minimize variance (=squared deviations). The concept of squared deviations breaks down when we use categorical data. So for the answer, yes, PCA can be used for the categorical data but it won't be that effective for categorical data.

Question 2:

Logistic Regression:

- Accuracy: 0.9444444444444444
- Precision: 0.9879518072289156
- Recall: 0.9534883720930233
- F1 Score: 0.963855421686747
- Cost(y-axis) vs the Number of iterations(x-axis) graph:

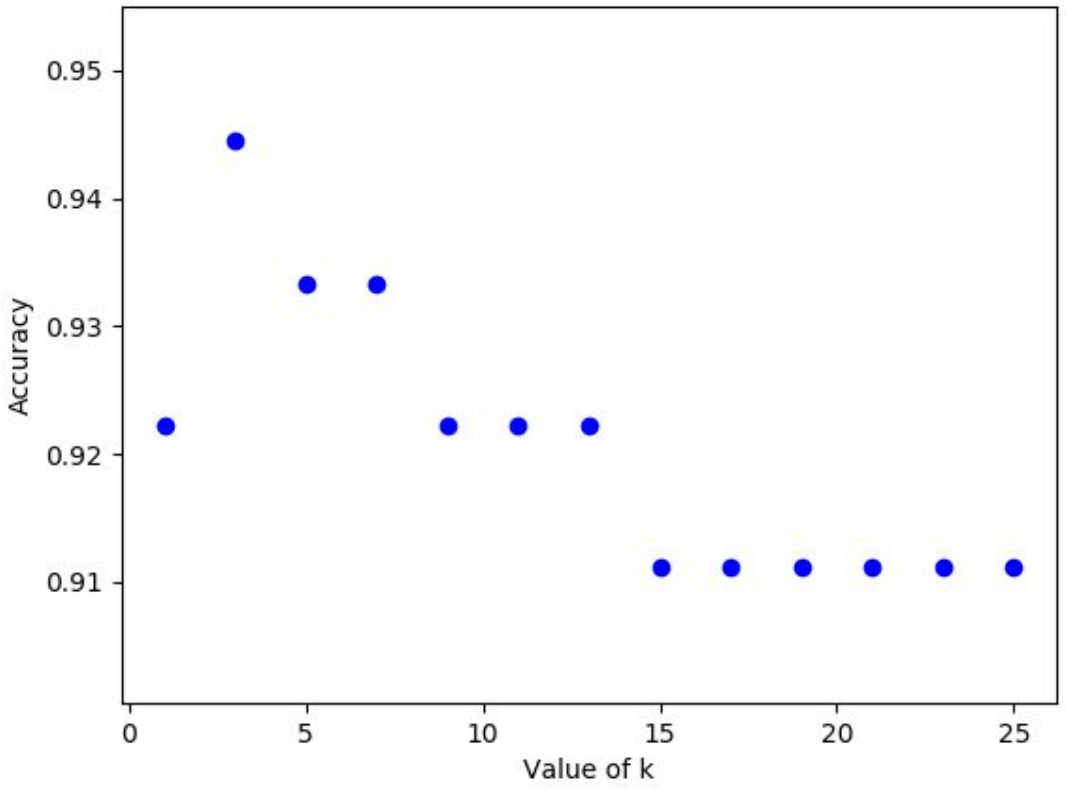


The values keep on changing because I have randomly shuffled the data, and on every new run, we get a different set of values in the test and train datasets.

KNN Algorithm:

- Precision : 0.968
- Recall : 0.968
- F1-score : 0.968

- Accuracy: 0.9411764705882353
- Accuracy(y-axis) vs K-values(x-axis) graph:



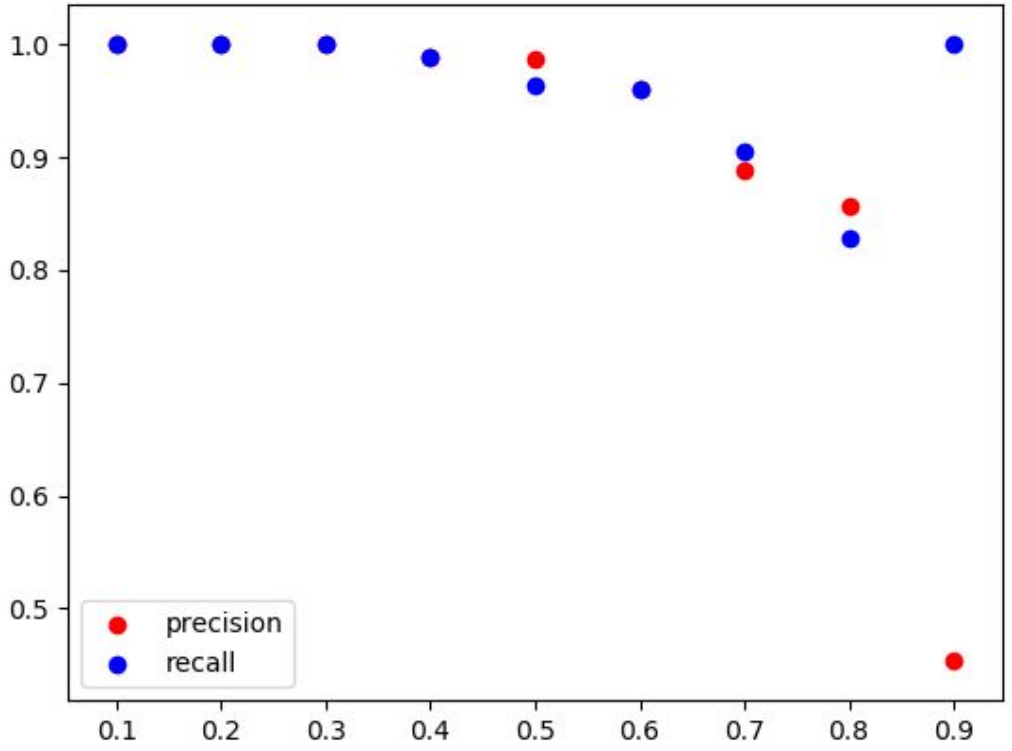
In this case also, like in the logistic regression, the values may change in different iterations because of the random shuffling of the dataset.

Comparison between Logistic regression and KNN:

- KNN is faster than Logistic regression because of the greater number of iterations.
- Accuracy of KNN > Accuracy of Logistic Regression (in this case)

Threshold Value:

- Threshold values vs precision/recall graph:



- Criteria for threshold value: There are multiple ways to select the threshold value and it depends on the objective of the regression model. Below is one reason:
    - When we don't want predictions to be biased, i.e., we want both the true positives and the true negatives to be high, we look for a threshold value which maximizes both sensitivity and specificity.
    - When we want the accuracy of our model to be high, in that case, we choose the cross validation method to determine the best threshold value for higher precision.
- Criteria I chose: I want to maximize the accuracy of my model, so I chose the threshold value accordingly.

Question 3:

One vs All Logistic regression:

There are 7 types of output labels {3, 4, 5, 6, 7, 8, 9} and below are the accuracies for each output variable:

- For output label 3, accuracy is 1.0
- For output label 4, accuracy is 0.9977298524404086
- For output label 5, accuracy is 0.8229284903518729
- For output label 6, accuracy is 0.7082860385925085
- For output label 7, accuracy is 0.9614074914869466
- For output label 8, accuracy is 1.0
- For output label 9, accuracy is 1.0

These also keep on changing on different runs of the code because of the random shuffling of the data.

One vs One Logistic Regression:

In one vs one we have to do nC2 combinations of the output labels. Out of those nC2, we have to sort out the maximum outputted labels. Those will be used for the output labels.

- Accuracy in round 0 is: 0.7185840707964601
- Accuracy in round 1 is: 0.8132387706855791
- Accuracy in round 2 is: 0.8702702702702703
- Accuracy in round 3 is: 0.8736263736263736
- Accuracy in round 4 is: 0.9875776397515528
- Accuracy in round 5 is: 0.9875776397515528
- Accuracy in round 6 is: 0.6522388059701493
- Accuracy in round 7 is: 0.9375
- Accuracy in round 8 is: 0.9463869463869464
- Accuracy in round 9 is: 0.9950980392156863
- Accuracy in round 10 is: 0.9950980392156863
- Accuracy in round 11 is: 0.9103448275862069
- Accuracy in round 12 is: 0.9094076655052264
- Accuracy in round 13 is: 0.9924812030075187
- Accuracy in round 14 is: 0.9924812030075187
- Accuracy in round 15 is: 0.8163265306122449

- Accuracy in round 16 is: 0.9285714285714286
- Accuracy in round 17 is: 0.9285714285714286
- Accuracy in round 18 is: 0.92
- Accuracy in round 19 is: 0.92
- Accuracy in round 20 is: 0.9

Comparison between One-v-One and One-v-All:

- One vs one is slower than one vs all logistic regression, because of the fact that the number of iterations in one vs one is quite large than one vs all.
- In this case, one vs one was more accurate.