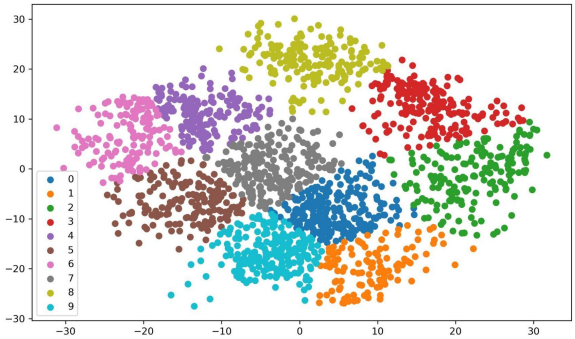
CSE506: Data Mining

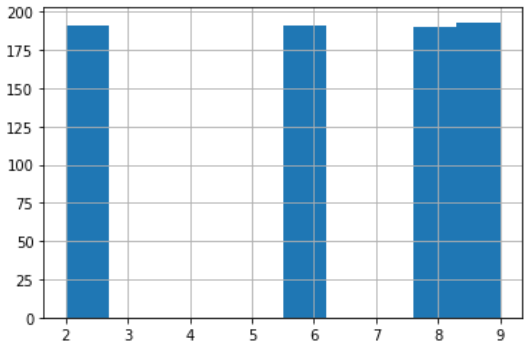
Assignment 3

Group 19.
Group Members: Ananya Kansal (2019458), Avishi Gupta (2019155), Jahnvi Kumari (2019469), Manvi Goel (2019472), Prachi Goyal (2019186)

DATASET

We selected the **digits dataset** for our anomaly detection tasks. It has 10 classes in it out of which we chose the digit classes **2,6,8,9** as our **target classes**. Then we selected few points from **other classes** to be introduced as **outliers**. These points were given labels from one of the 4 classes used for classification.





EDA measures -

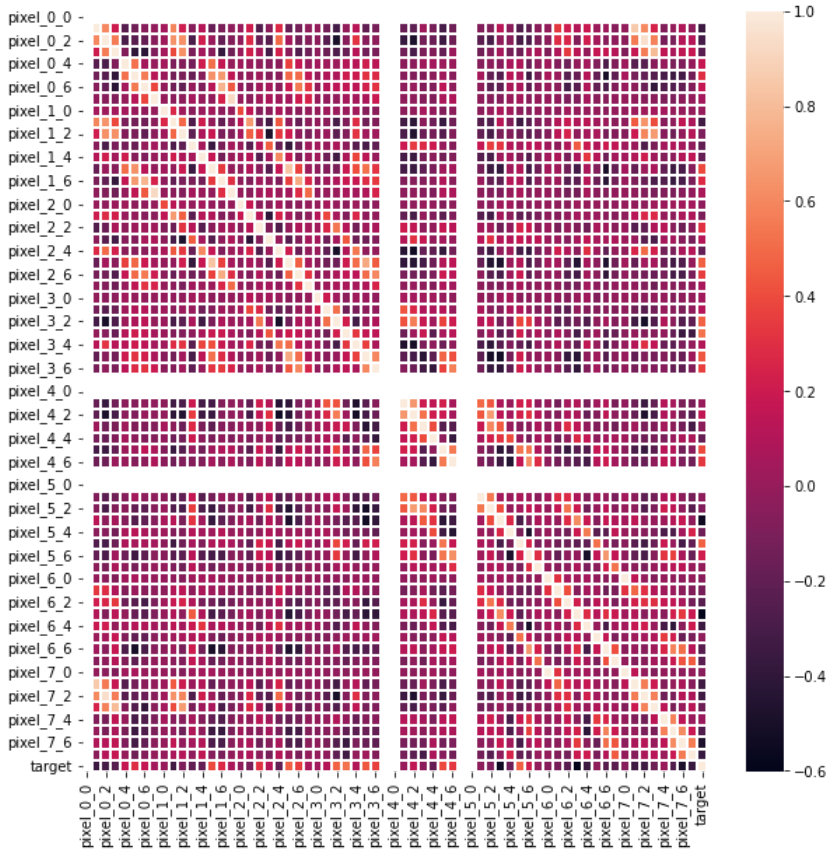
- Data.info() provides us with information about the data type of the features and if there are any null values.

```
Data columns (total 65 columns):
 #   Column     Non-Null Count  Dtype
---  ------     --------------  -----
 0   pixel_0_0  765 non-null    float64
 1   pixel_0_1  765 non-null    float64
 2   pixel_0_2  765 non-null    float64
 3   pixel_0_3  765 non-null    float64
 4   pixel_0_4  765 non-null    float64
```

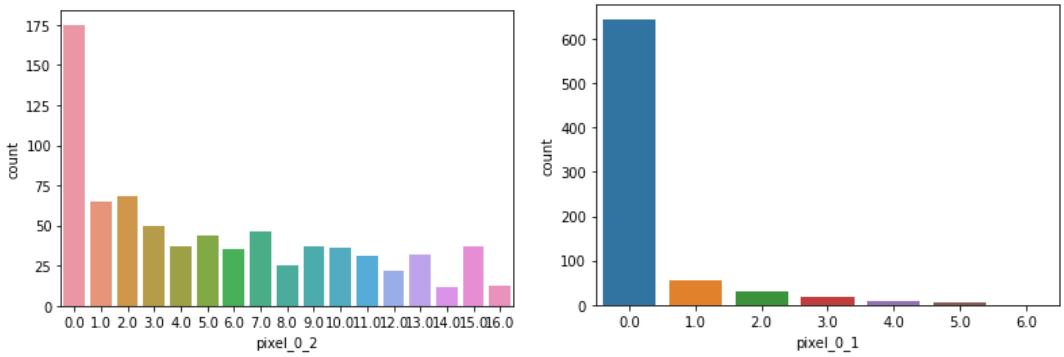
- Data.describe() gives us all the statistical information of the features like the minimum, maximum, median, mean values.

	pixel_0_0	pixel_0_1	pixel_0_2	pixel_0_3	pixel_0_4
count	765.0	765.000000	765.000000	765.000000	765.000000
mean	0.0	0.317647	5.354248	11.969935	10.856209
std	0.0	0.877147	4.932927	3.918059	4.644169
min	0.0	0.000000	0.000000	0.000000	0.000000
25%	0.0	0.000000	1.000000	10.000000	8.000000
50%	0.0	0.000000	4.000000	13.000000	13.000000
75%	0.0	0.000000	9.000000	15.000000	15.000000
max	0.0	6.000000	16.000000	16.000000	16.000000

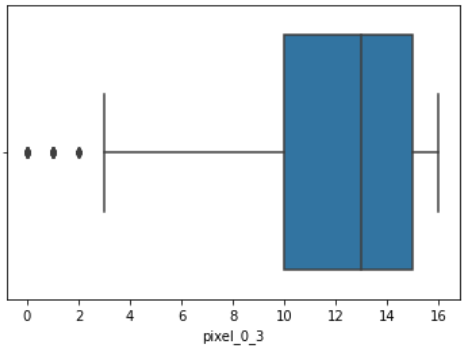
- The heatmap of the data represents the correlation between all the features and the label. We want to retain features having high correlation with the y label and we want to keep a subset of the features having high correlation among themselves. In the given heatmap we can see that pixel values 0, 40 and 50 have a single unique value and that is why we observe no variation. We can remove these pixel values because they give us no information for the classification task.



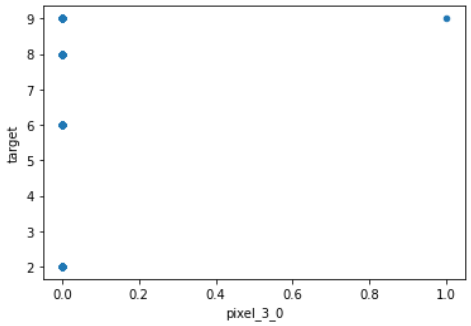
- Countplots make histograms of the all the unique values that the features take. The first countplot on the left shows a uniform distribution between the different values that a feature takes while the other is a more skewed distribution which shows that pixel 1 has value 0 for most of the data points.

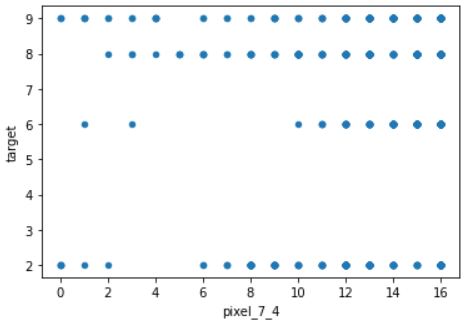


- Boxplots are an easy way to represent the mean, median, variance and percentile distribution of the data and even detect the outlier values in the feature values. In the plot below we see all the pixel values (<3) that pixel 3 takes are flagged as outliers as they lie outside the whiskers.

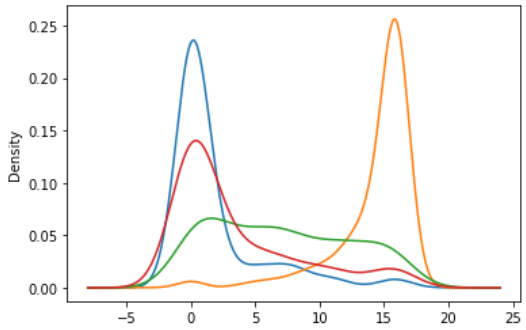


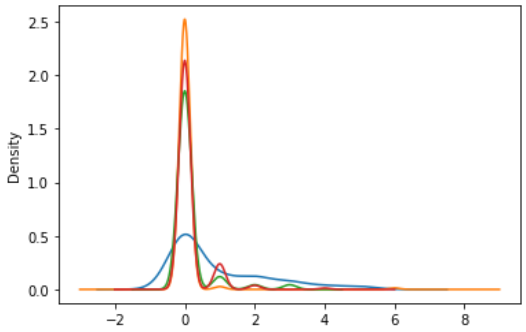
- Scatter plots are a helpful tool to look at the distribution of the feature values according to the target classes. This variation can help us detect features that can help distinguish between the various classes. For example by looking at the plot given below we can say that pixel 3 might not be a good feature to distinguish between classes since it takes the value 0 majoritatively for all classes whereas for pixel 74 it shows variation in pixel values with the variation in target class and hence might prove to be an important feature.





- We can also plot the distributions for the features to visually understand their statistical properties like variance, mean and type of the distribution. We can use the groupby function to plot the distributions of features according to the attribute that is used to group them. For example we can group the data according to their class labels and plot the separate distributions of one feature for all the classes. In the first diagram on the left we can see that the feature has different distributions corresponding to the target variable, hence it is a good feature to be used for classification whereas on the right the mode values of the pixel collide for all the target classes and less variation is observed, hence it may not act as a useful feature for classification.

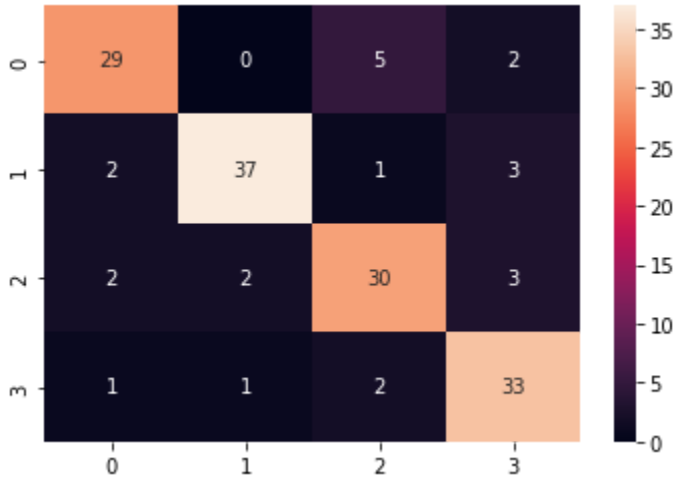




The Baseline model that we run for the classification task is Decision Tree classifier with its default parameter values and it gave us an accuracy of 86.92%. The classification report is as follows:

Class	precision	recall	f1-score	support
2	0.90	0.86	0.88	42
6	0.94	0.89	0.92	38
8	0.79	0.76	0.77	29
9	0.82	0.91	0.86	44
accuracy			0.86	153

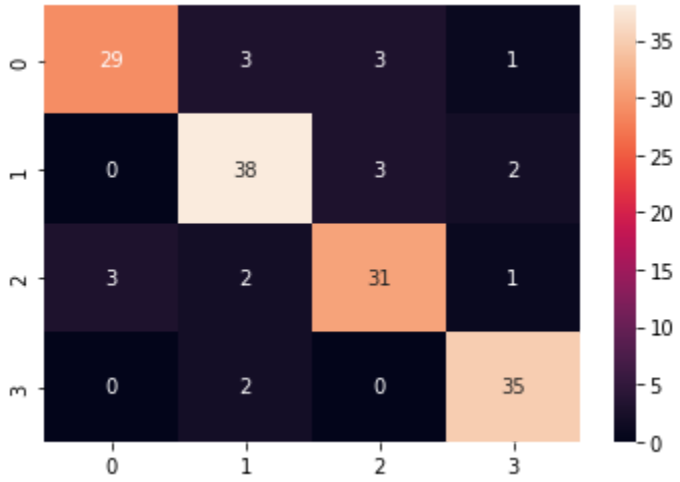
macro avg	0.86	0.85	0.86	153
weighted avg	0.87	0.86	0.86	153



3.c DIMENSIONALITY REDUCTION ALGORITHM

Dimensionality reduction techniques are used in reconstruction-based approach to anomaly detection. I used PCA to find the principal components and then measured the reconstruction error of each object. The objects with large reconstruction errors were classified as anomalies:

Confusion matrix

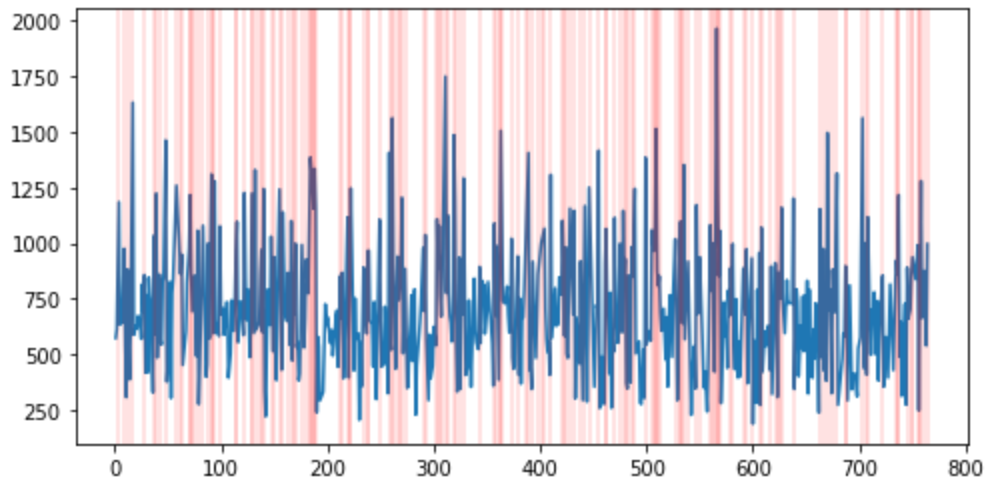


Classification reports of the best experiment

Accuracy = 0.8888888888888888

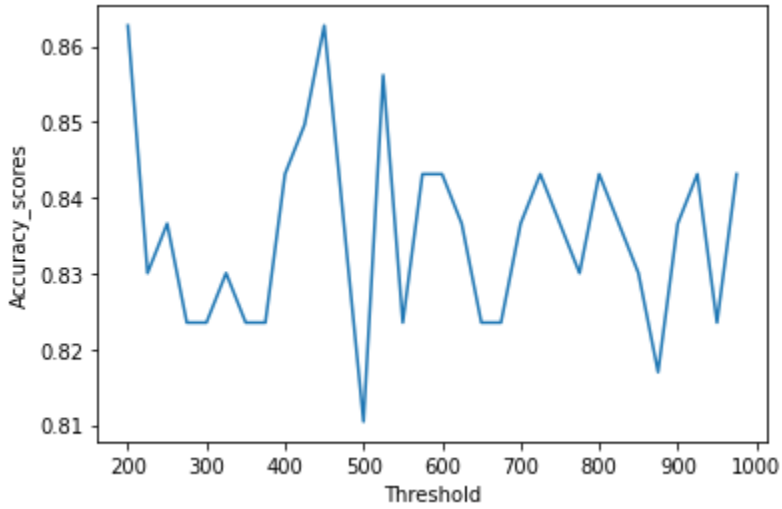
	precision	recall	f1-score	support
2	0.91	0.86	0.88	35
6	0.93	0.95	0.94	44
8	0.76	0.81	0.78	31
9	0.93	0.91	0.92	43
accuracy			0.89	153
macro avg	0.88	0.88	0.88	153
weighted avg	0.89	0.89	0.89	153

Reconstruction errors with the anomalous data points identified

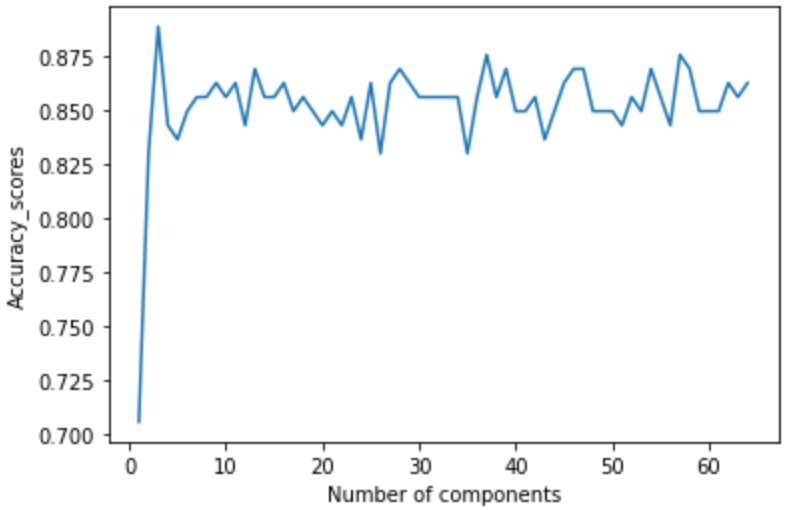


166 data points were removed from training to get a 89% accuracy.

Variation of accuracy with the threshold used



The accuracy is varying with a little fluctuating.

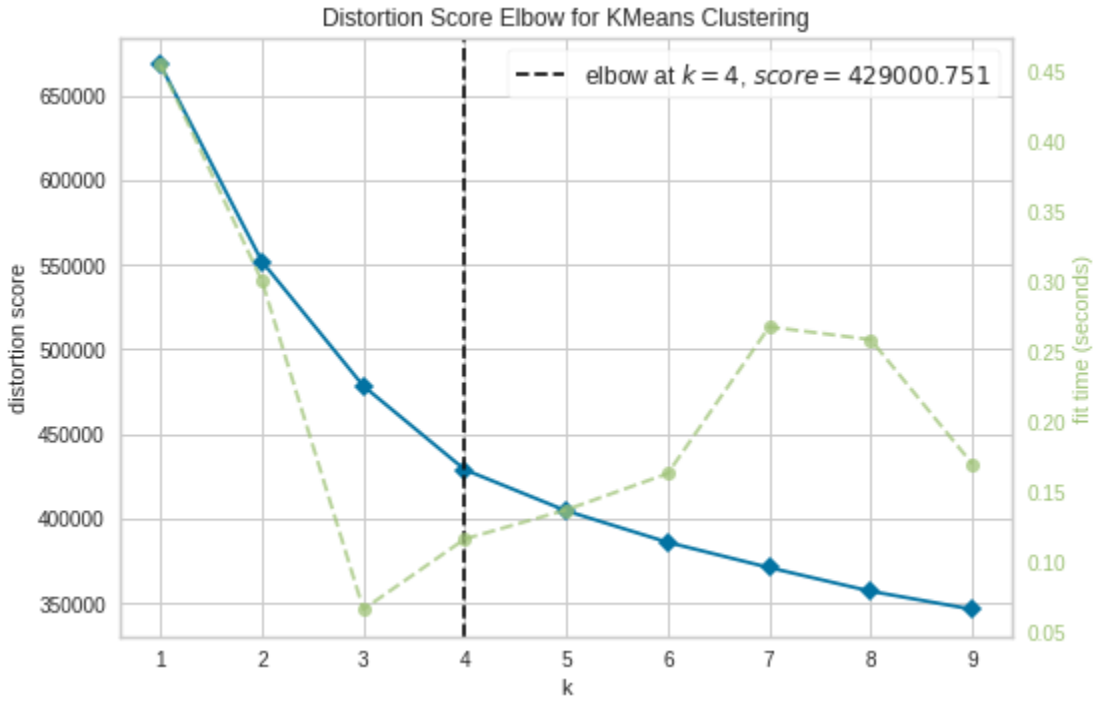


Analysis: Increasing the threshold keeping the number of components same increases the accuracy. This means that the reconstruction error is a good way to identify which points are anomalous

3.d CLUSTERING ALGORITHM

We use KMeans as the clustering algorithm. KMeans is a prototype based clustering algorithm, thus we can classify an object as outlier if it is not close enough to a cluster center.

Using this methodology, we train a KMeans model on the dataset.
To find the optimal number of clusters we make use of the Elbow Method.

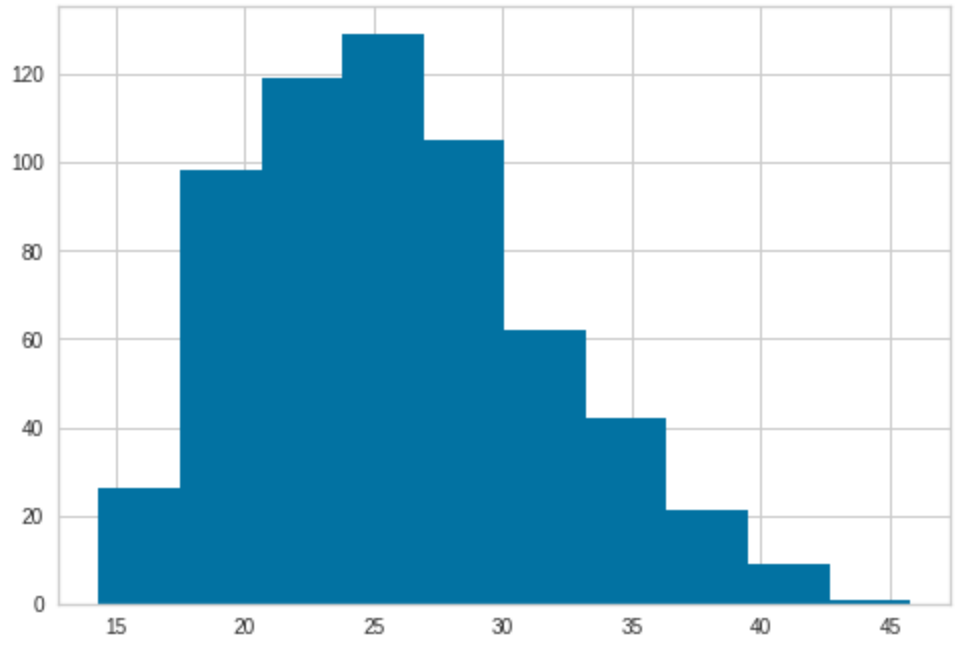


From the plot we can see that, the optimal number of clusters is 4.

We train the KMeans algorithm from sklearn library with 4 clusters.

SSE Score for Clustering = 428998.3932157522

We then find the distance of each object in the training data to the closest cluster center and plot the distance in form of a histogram

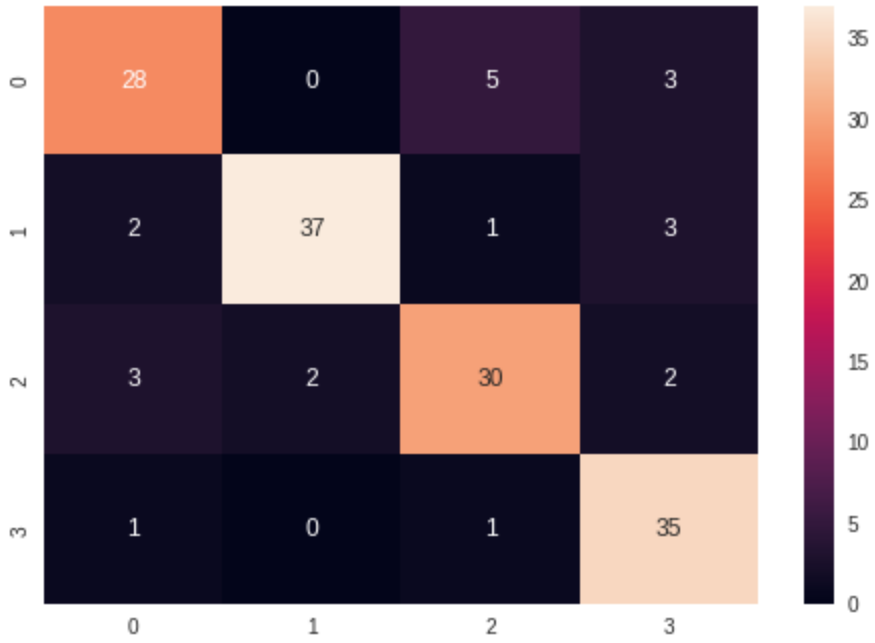


From the plot we can see that we have sufficient number of points till the distance 42, but after that the number of points is very less. Thus, we can consider 42 as threshold for defining anomalies.

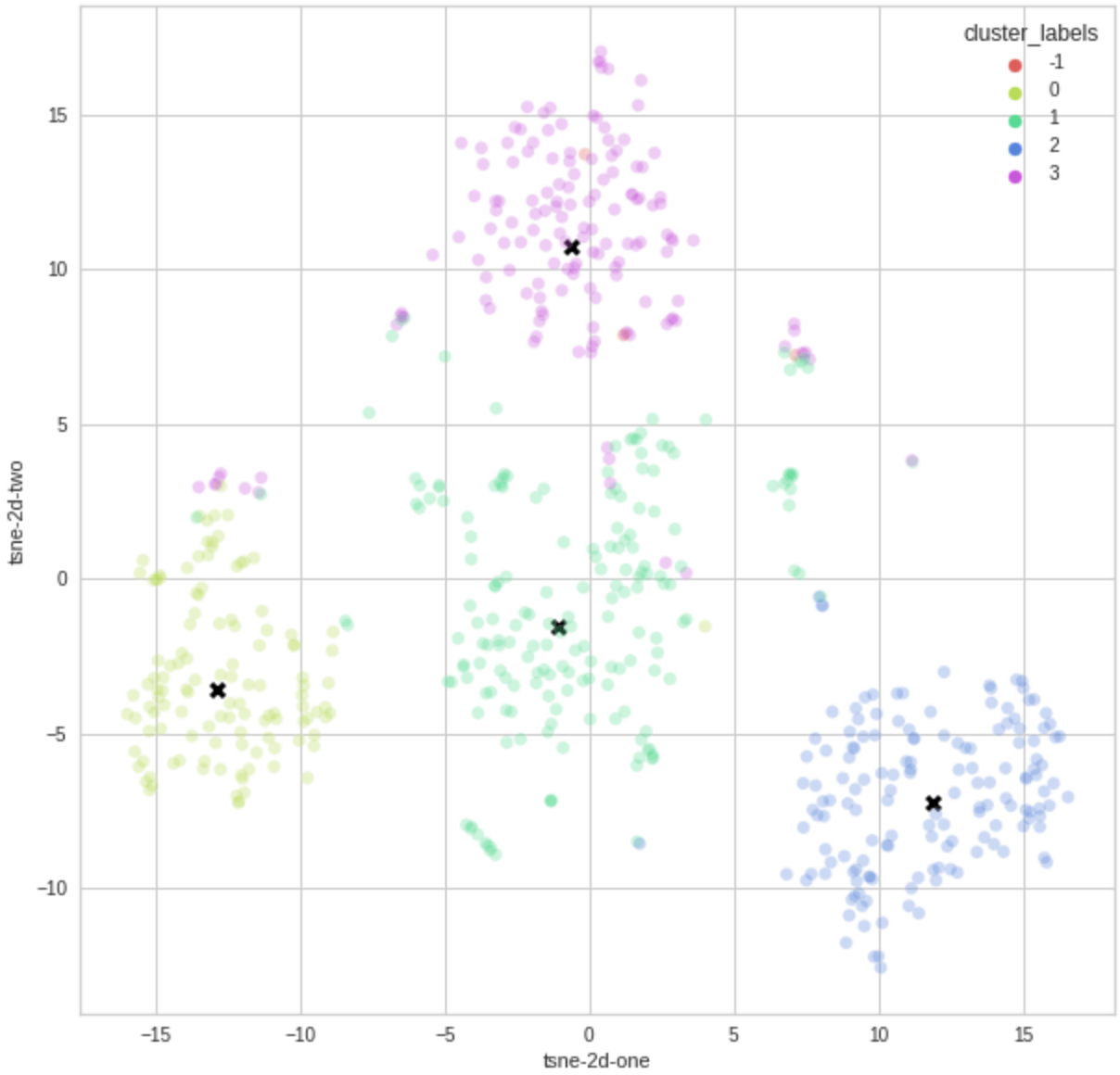
We drop all such outliers and train the decision tree again.
The new results are.
Classification Report

	precision	recall	f1-score	support
2	0.78	0.82	0.80	34
6	0.86	0.95	0.90	39
8	0.81	0.81	0.81	37
9	0.95	0.81	0.88	43
accuracy			0.85	153
macro avg	0.85	0.85	0.85	153
weighted avg	0.85	0.85	0.85	153

The confusion matrix.



We also plot the points using TSNE plots. The outliers are with label -1.



Analysis for Clustering

We can see that KMeans which is suitable for the dataset, is not only efficient but also good at identifying outliers in the data. From the TSNE plot we can see that many points that are at a distance from two cluster centers or lie between two clusters are mostly identified as outliers which is true for our dataset.

The same can be said using empirical measures where all the metrics improved as the model became more robust towards outliers. We also notice an increase in classwise accuracy which comes from improved learning by the model.

Shortcomings of the Method and How to fix them.

The performance of clustering algorithms such as KMeans is heavily dependent on the number of clusters used as well as the presence of outliers in the data. Also, clustering algorithm needs to be selected depending on the data. The number of outliers are also dependent on the threshold chosen for anomaly score.
Keeping these shortcomings in mind, we use Elbow Method to calculate the number of clusters, which comes as 4, which is also the true number of clusters in the data. For the second problem, we see that the data used has globular data which is suitable for prototype based clustering, hence the chosen method as KMeans. We specifically make use of a histogram with multiple bin sizes to calculate the threshold.

Learnings.
From this exercise we were able to get more insights on different functions of clustering algorithm and setting a threshold to define anomaly classes.

3.e CLASSIFICATION ALGORITHM

We have used **One-Class Support Vector Machine** for outlier detection. We removed the outliers using the one-class support vector machine and then trained and tested the dataset using a vanilla decision tree classifier.
We compare results for vanilla decision tree before and after removing anomalies:

Before removing anomalies:

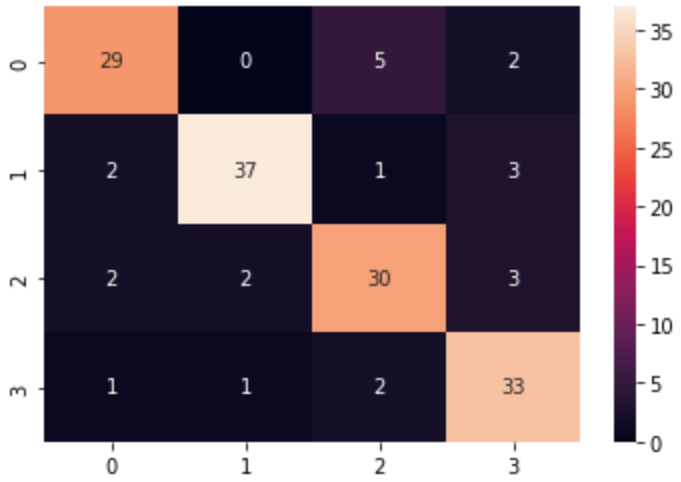
Accuracy score: 0.8431372549019608 (shown in screenshot attached below).
Classification report (shown below).

```
0.8431372549019608
              precision    recall  f1-score   support

           2       0.81      0.85      0.83        34
           6       0.86      0.93      0.89        40
           8       0.81      0.79      0.80        38
           9       0.89      0.80      0.85        41

    accuracy                           0.84       153
   macro avg       0.84      0.84      0.84       153
weighted avg       0.84      0.84      0.84       153
```

Confusion matrix:



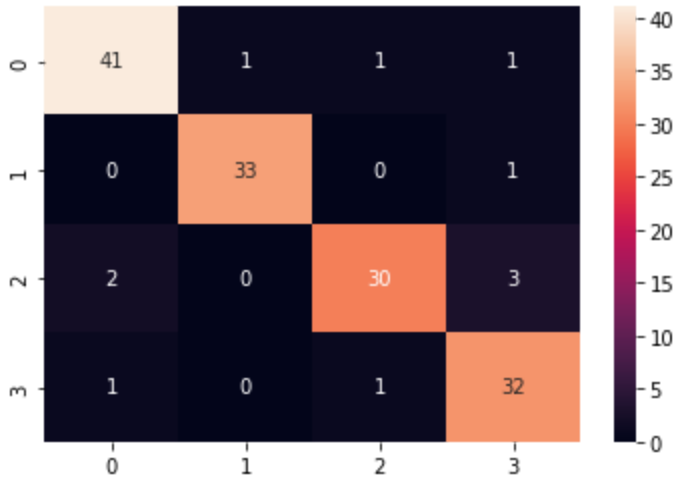
After removing anomalies using one-class SVM:

Accuracy score: 0.9251700680272109 (shown in screenshot attached below).
Classification report (shown below).

0.9251700680272109

	precision	recall	f1-score	support
2	0.93	0.93	0.93	44
6	0.97	0.97	0.97	34
8	0.86	0.94	0.90	32
9	0.94	0.86	0.90	37
accuracy			0.93	147
macro avg	0.93	0.93	0.92	147
weighted avg	0.93	0.93	0.93	147

Confusion matrix:



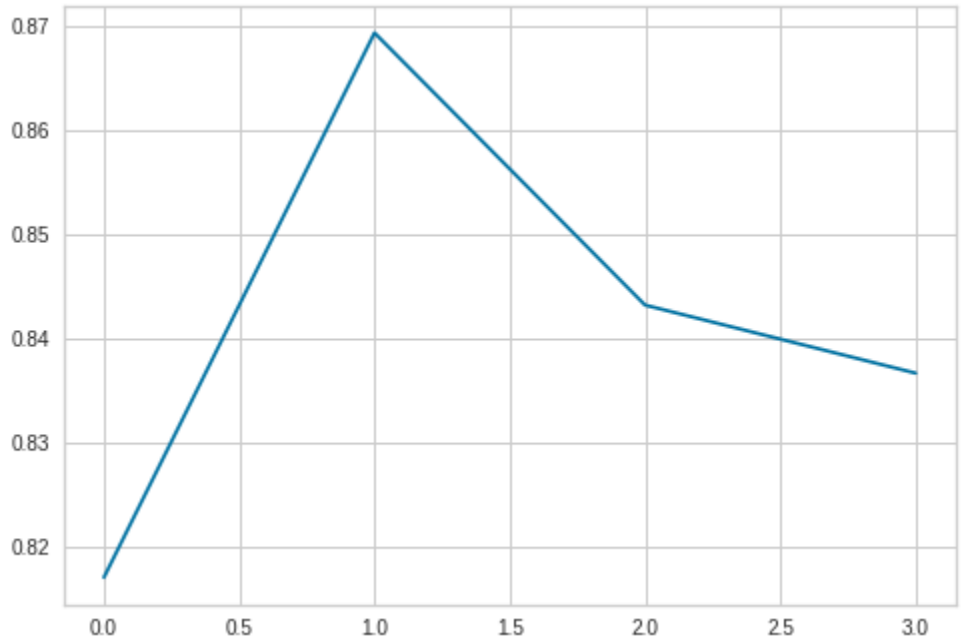
Analysis: We notice the accuracy score calculated using the Decision Tree classifier after using the one-class SVM to remove outliers is better as compared to using the Decision Tree without removing outliers. This shows that removing outliers reduces the chances of misclassification. Before removing anomalies, the shape of data was: (765, 66), while shape after removing anomalies was: (735, 66).

Shortcomings: Vanilla decision tree trained on dataset containing anomalies gives poor test accuracy as it overfits the data. Thus, accuracy of the classifier improves after removing anomalies.

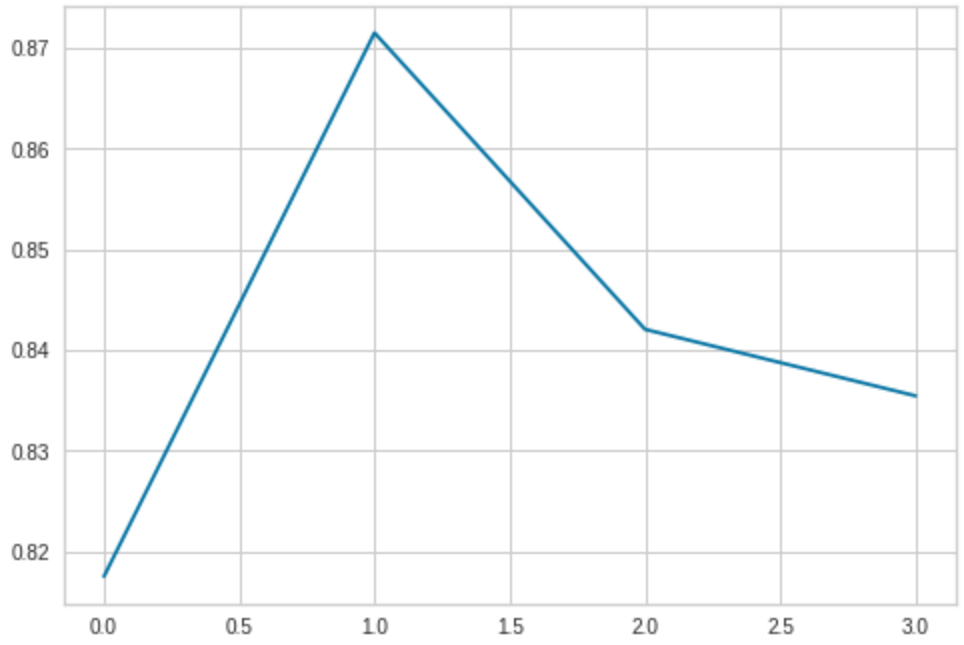
3.f INFOGRAPHICS

We plot all the four models across various evaluation metrics to see the change or improvement of each model and to see which method performs better with respect to which metrics.

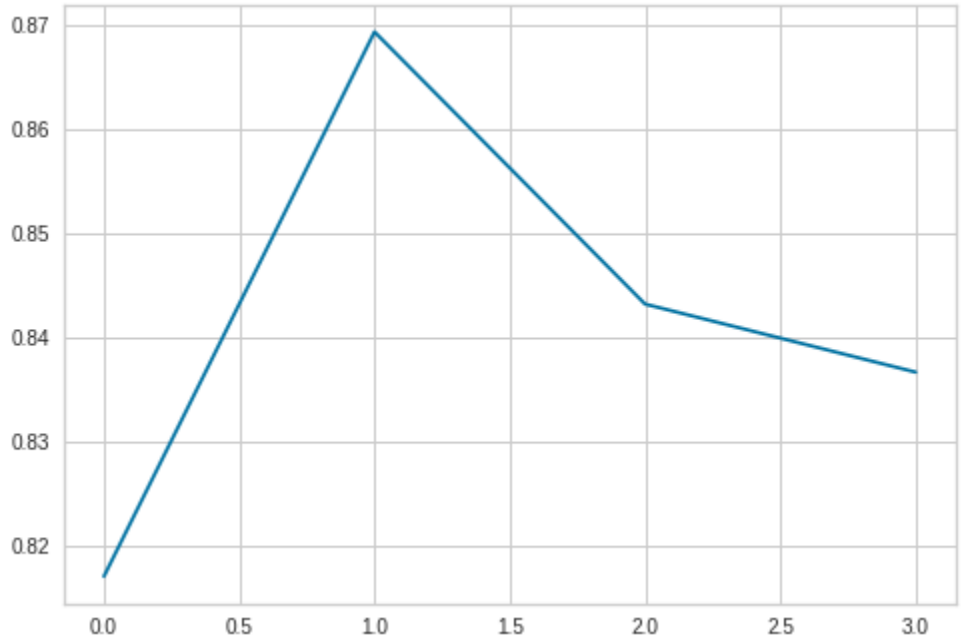
Accuracy



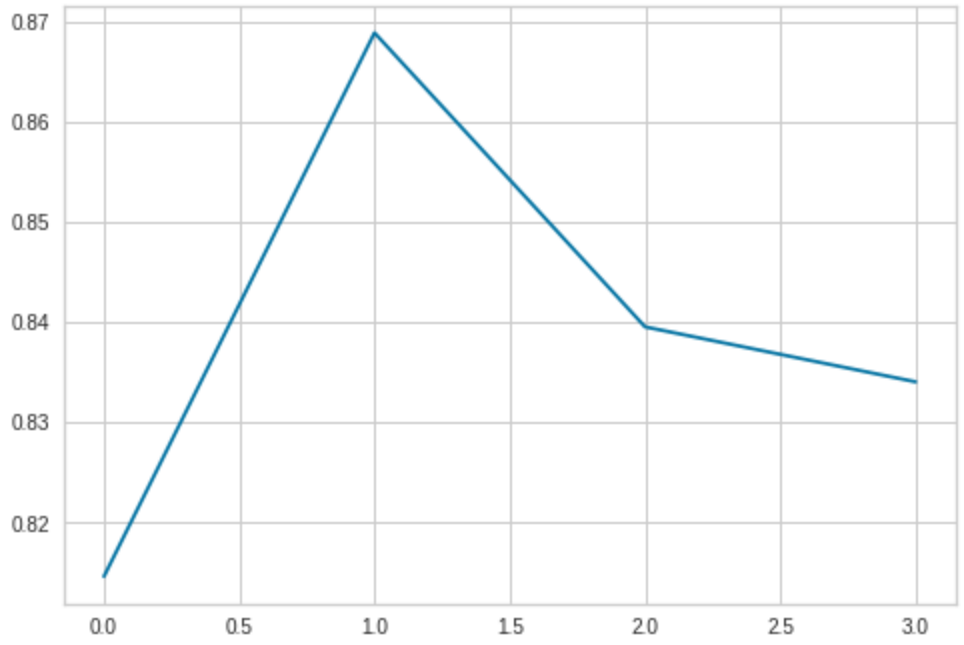
Balenced Accuracy



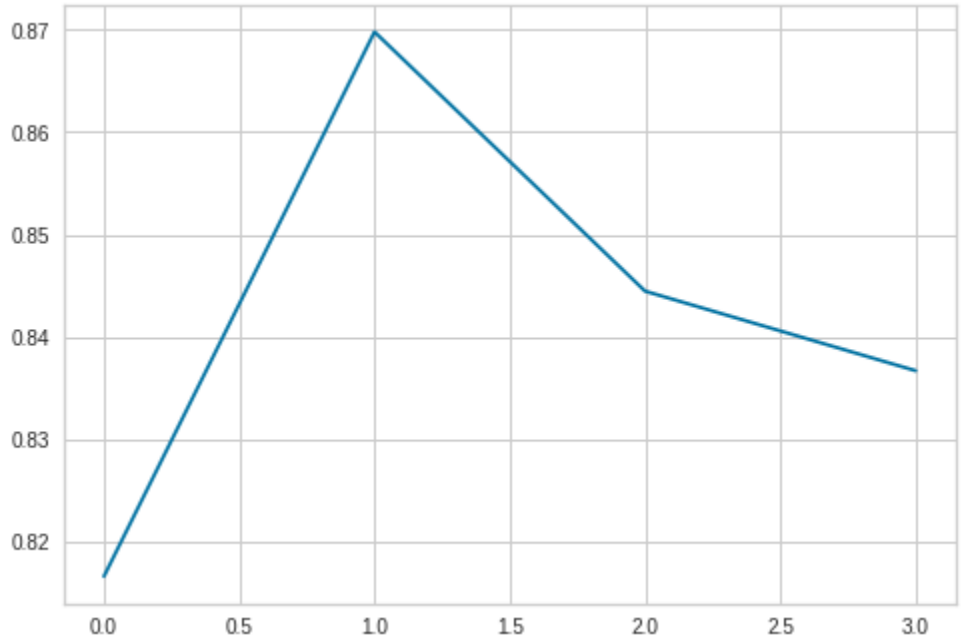
F1-Score (Micro)



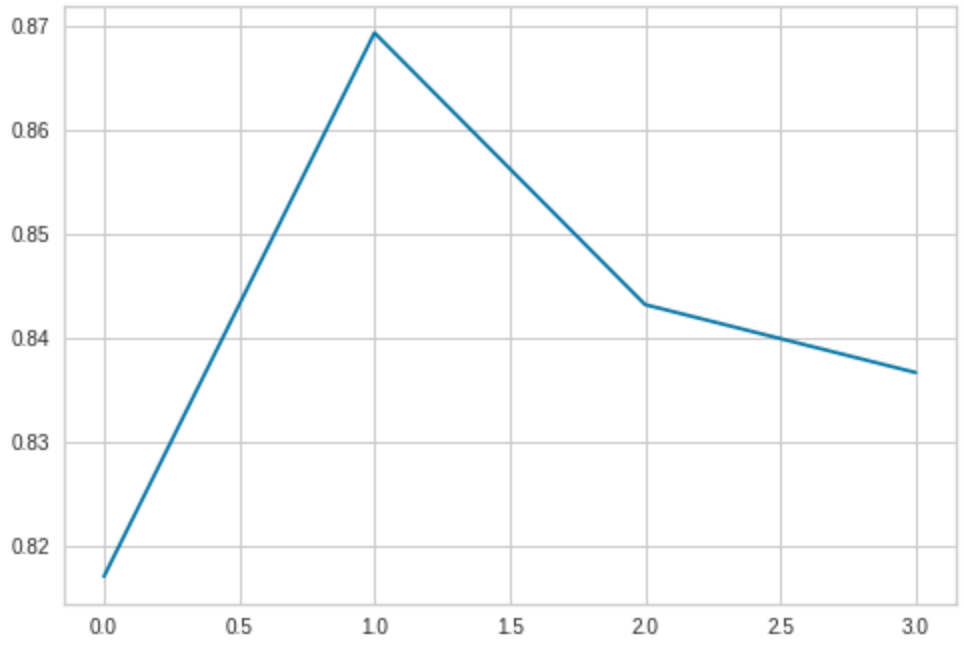
F1-Score (Macro)



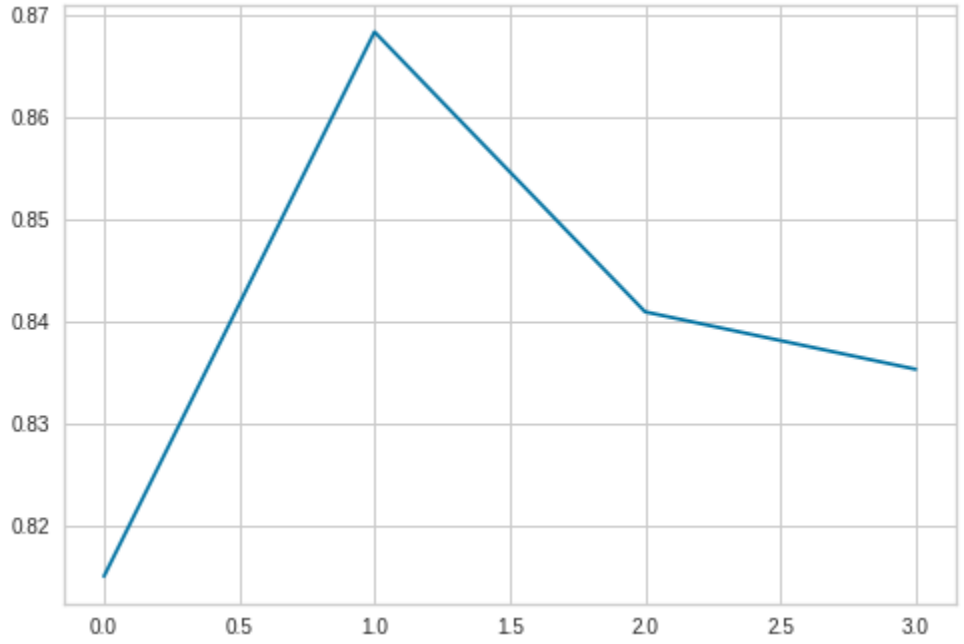
F1-Score (Weighted)



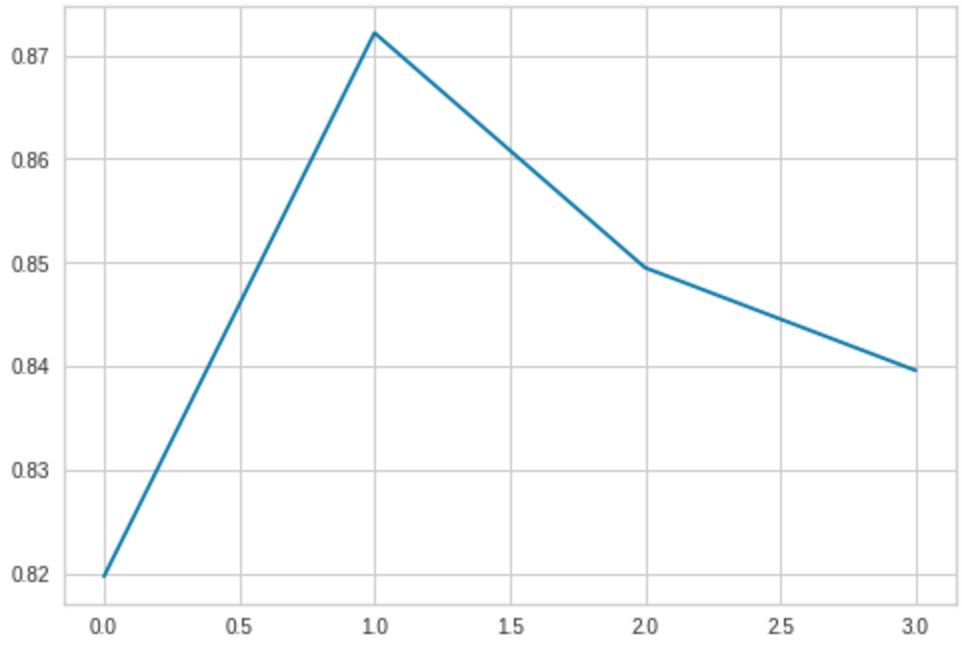
Precision (Micro)



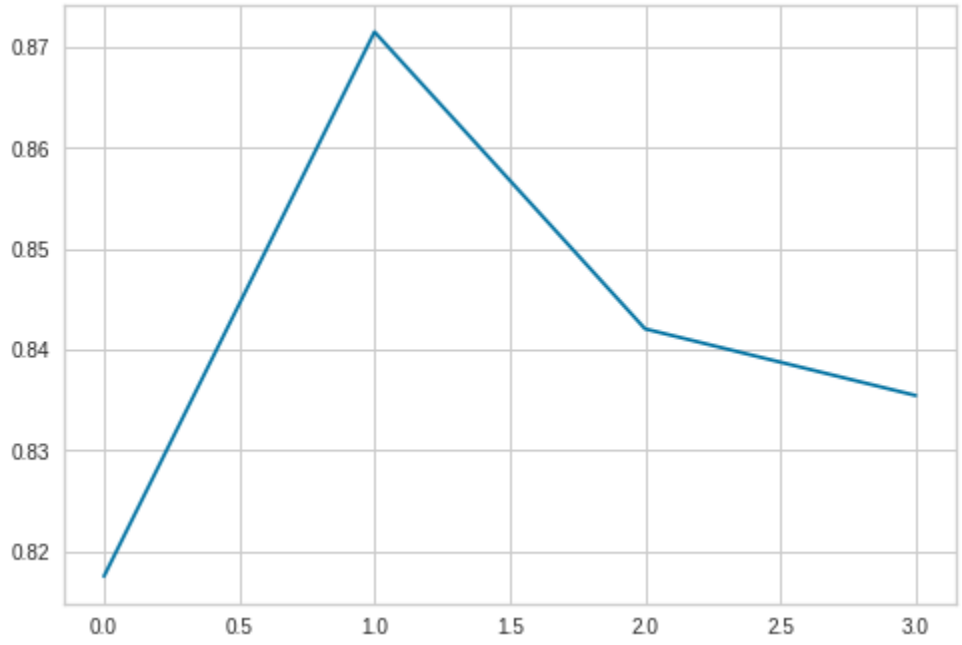
Precision (Macro)



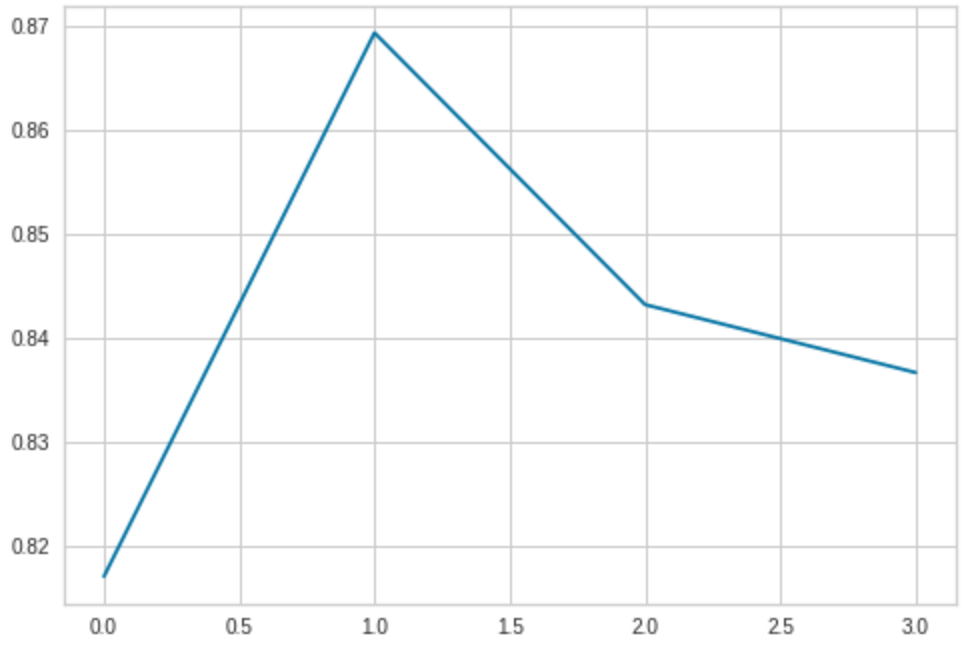
Precision (Weighted)



Recall (Macro)



Recall (Weighted)



We notice a similar trend in the plots where all the models outperform the baseline model but we can notice that some models outperform others which depends both on the dataset chosen and the algorithm chosen in the category.
One thing to note is that clustering algorithm does not improve the performance by much, this can be attributed to the fact that very samples were chosen as outliers by the algorithms due to selected threshold.
In one-class SVM, 30 outliers are removed - which is high as compared to other algorithms.Thus resulting in best performance among selected algorithms.