CS 412

PROJECT PRESENTATION

By :-

Kumar Chandrasekhar (Graduate Student)

Kalkidan Sisay (Undergraduate Student)

Vinayak Kabra (Graduate Student)

TOPIC OF PROJECT:

CREDIT CARD FRAUD DETECTION

Goal

- Try to build a machine learning model that could accurately detect all the fraudulent transactions in an unknown dataset, thus potentially saving capital for customers.
- Use different machine learning techniques in order to achieve this goal.

Problem Description

Problem Description Link:

https://www.kaggle.com/mlg-ulb/creditcardfraud/home

- This is a classification problem with 29 input features of a particular credit card fraud detection and 2 output classes that tells us if a particular transaction is fraudulent or not.

Approaches

- We have tried Supervised as well unsupervised learning approaches.

Supervised Learning:

- Logistic Regression
- K-Nearest Neighbours
- Support Vector Machine
- Sequential Neural Network
- Gaussian Naive Bayesian

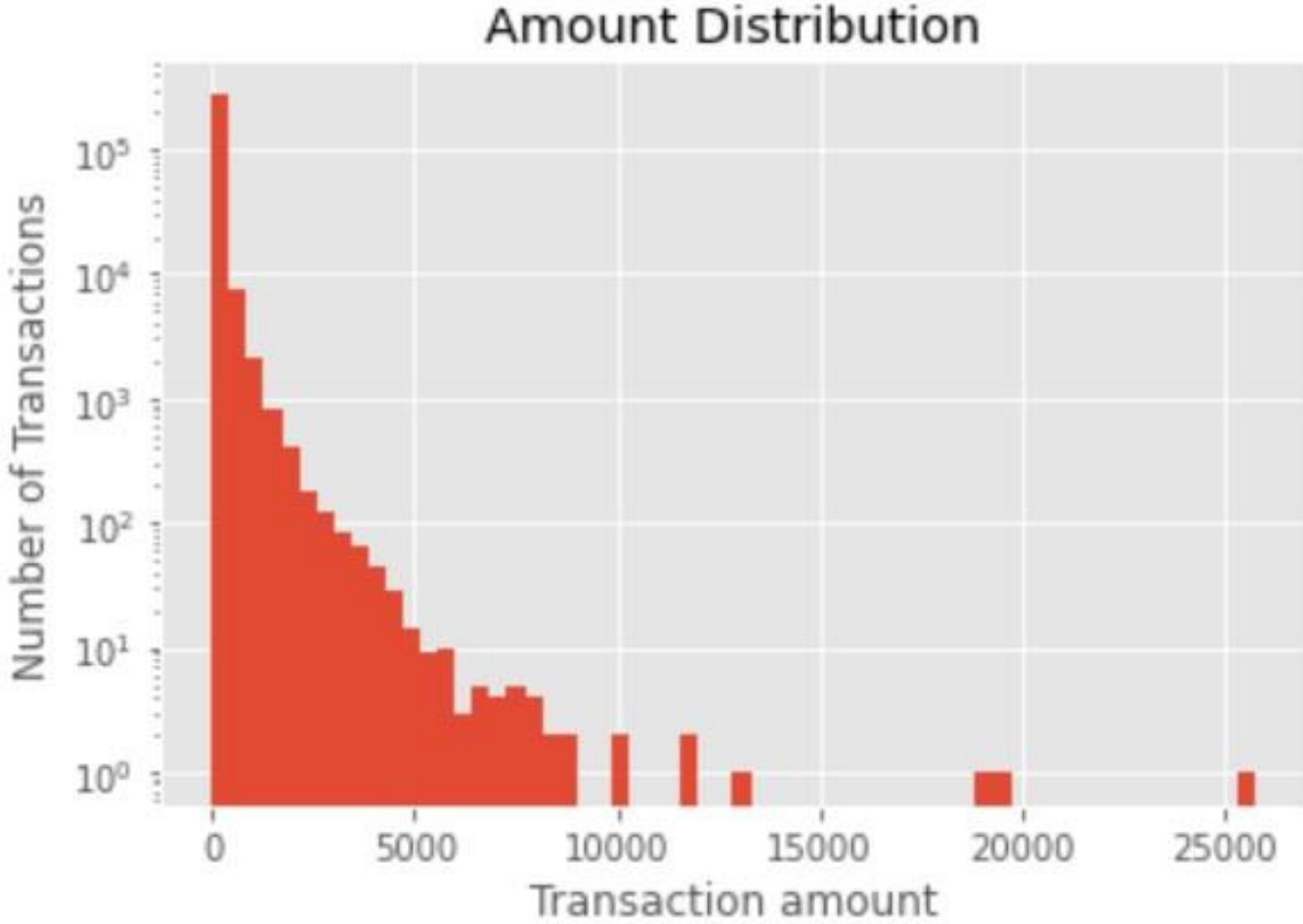
Unsupervised:

- K-Means

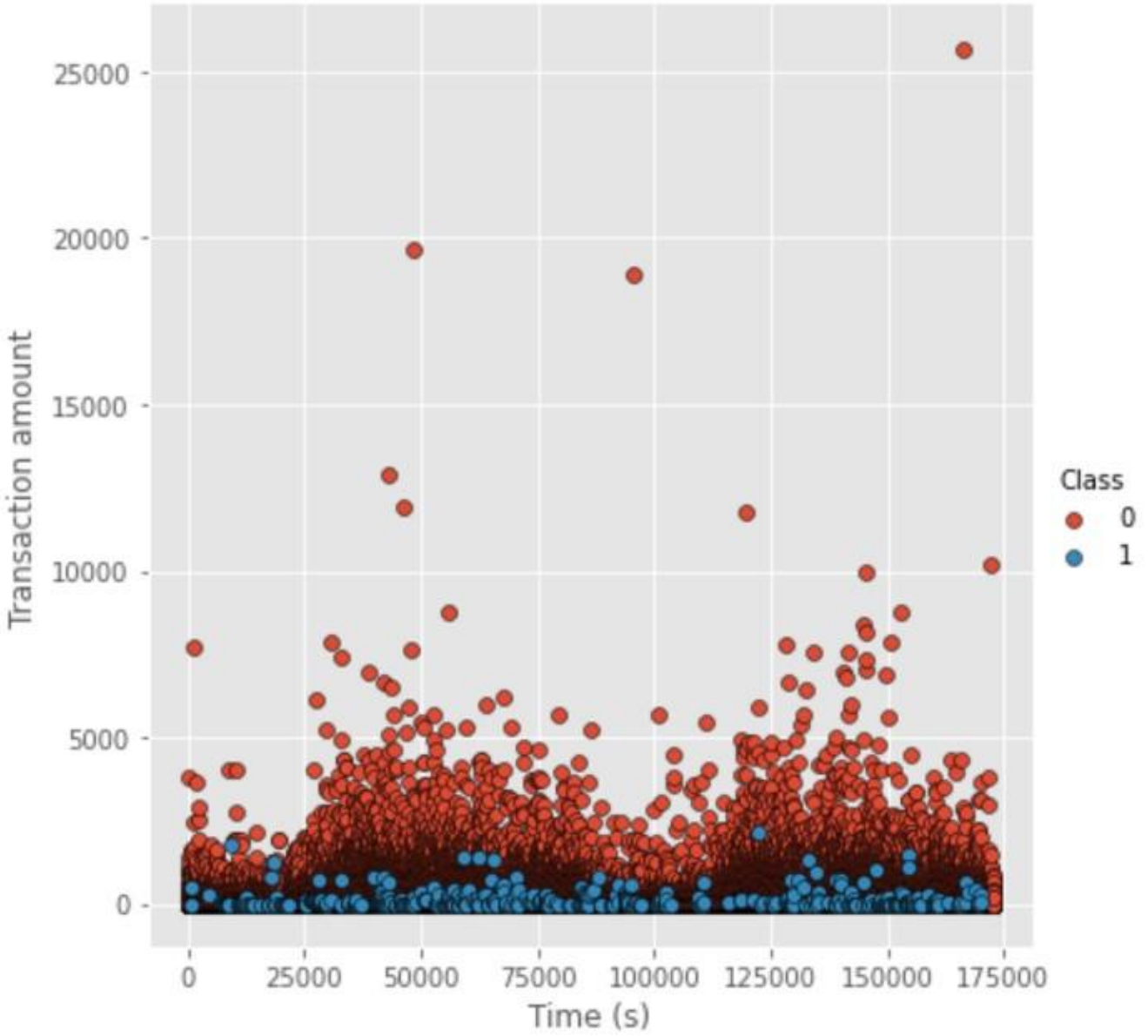
Datasets

- The dataset taken is PCA enabled data. Hence, no dimensionality reduction was performed. There are total of 29 input features and one output feature, which tells us if a particular transaction is fraudulent or not.
- The input features are V1,V2,V3….V28,Amount.
- The dataset is real data taken from an unknown organization and due to confidentiality of the data, the input features name has been masked as seen above, except for the last data which is the amount of the transaction performed.
- Initial data cleaning has been done, in order to remove column names and convert it into a feature vector that can be fed into the machine learning models.
- The ‘amount’ input feature lies in the range of 0-25700, whereas V1-V28 lies in the range of -150 to 150. Hence, normalization has been done to bring all the inputs to same range.

- The total transactions in the dataset is 284807, out of which, there are about 492 fraudulent transactions.
- Hence, in order to measure the accuracy, we have used the recall as accuracy measure, since overall accuracy will not give us sufficient information about the model.
- The Recall is given by (No. of actual fraudulent transactions/(No. of detected fraudulent transaction)
- The dataset has been split into training and test in the ratio 80 : 20.

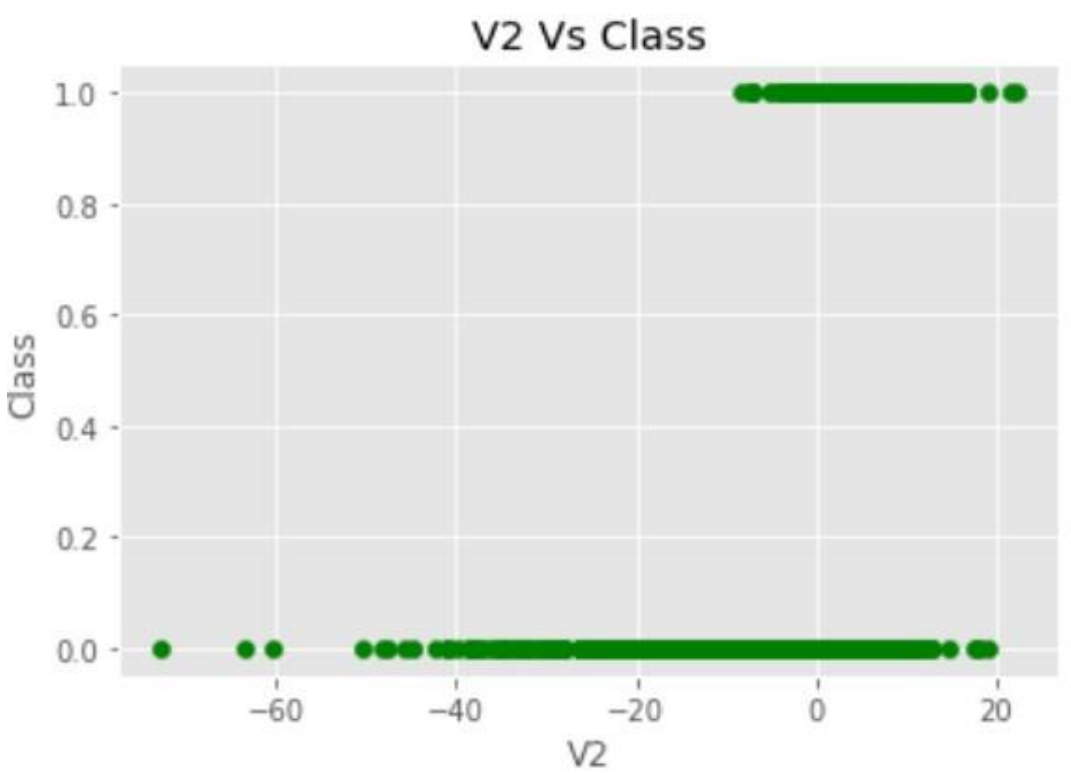

Amount Distribution
Number of Transactions
10^5
10^4
10^3
10^2
10^1
10^0
0
5000
10000
15000
20000
25000
Transaction amount

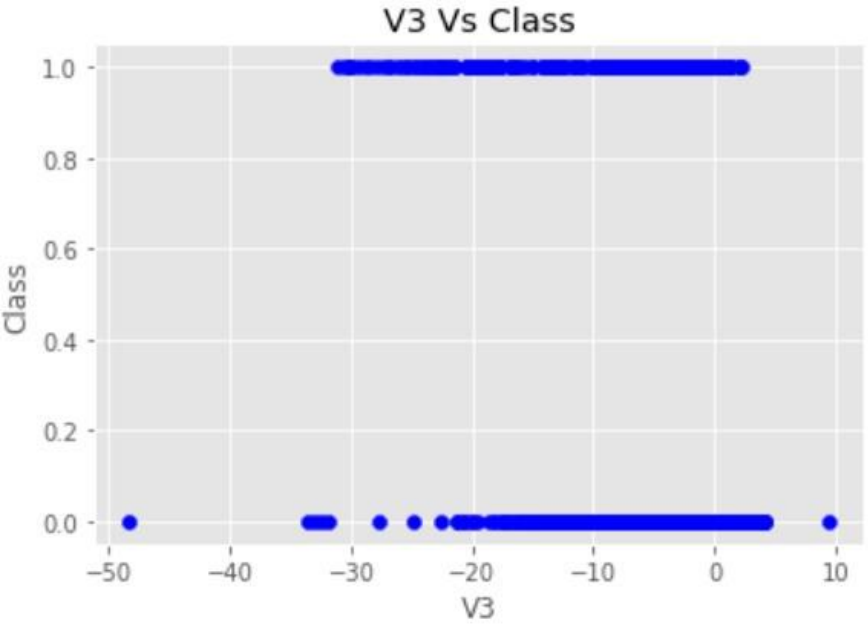
Transaction Amount VS Time

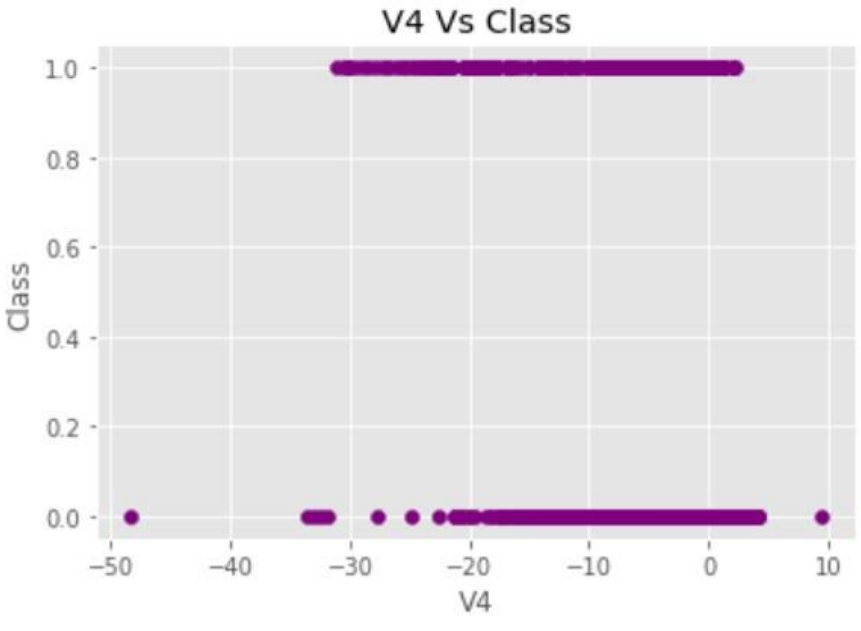


Class Vs V


V1 Vs Class
Class
V1
1.0
0.8
0.6
0.4
0.2
0.0
−50
−40
−30
−20
−10
0


V2 Vs Class
Class
V2
1.0
0.8
0.6
0.4
0.2
0.0
−60
−40
−20
0
20


V3 Vs Class
Class
V3
1.0
0.8
0.6
0.4
0.2
0.0
−50
−40
−30
−20
−10
0
10


V4 Vs Class
Class
V4
1.0
0.8
0.6
0.4
0.2
0.0
−50
−40
−30
−20
−10
0
10

Libraries used

We have used **Pandas** for data cleaning and parsing

Sklearn for training the data and to use the machine learning models

Numpy- for array operations

Keras- For implementing Sequential Neural Network

Logistic Regression

- We have used default sigmoid function as activation function in this logistic regression.

- The input is 29x(number of data) input feature vector.
- The entire dataset has been divided into training and test data in the ratio 3:1.
- The model was best trained when the hyper parameter max_iter was set to 4000.
- The overall accuracy achieved was 99.92 percent. But, we consider the percentage of fraudulent detection detected correctly as an accuracy measure(**Recall**), we get **67.5%** as accuracy for this model. This accuracy was before normalizing the data.
- After **normalizing**, the overall accuracy was 99.94 and fraudulent accuracy is **80%**
- The accuracy decreases when the iterations(**max_iter**) is increased further.

Logistic Regression

When K-fold Cross Validation was performed with 10 splits of the data.

The Recall for Logistic regression without normalization was same i.e 70%

But, when the data was normalized, the K-fold Cross Validation Recall reduced from 80% to 62%.

K-Means

- Here, we took only the input feature vector and tried classifying it into 2 clusters.
- The number of iteration was 10000, for which we receive best possible accuracy.
- The output class columns were removed since we are using an unsupervised model here, and only X was considered for training the model
- Without Normalization, below are the results:
- We got overall accuracy of 47%% upon training.
- However, the Recall(% of fraudulent transactions) was 73.33%
- Precision was very very less.
- When we did K-fold cross validation, the recall(the accuracy measure) reduced to 47% from 73%
-
- Increasing or decreasing the maximum iteration decreases the accuracy.

K-Means

With Normalization,

- We got overall accuracy of 47% which is similar to what we got without normalization.
- However, the Recall(% of fraudulent transactions) was 83.33% as compared to 73.33% without normalization
- Precision was very very less.
- When we did K-fold cross validation, the recall(the accuracy measure) reduced to 56% from 83%, but still, this is better than 43% without normalization.

Support Vector Machine

- The model was best trained when the hyper parameter max_iter was set to 1000.
- Each iteration takes longer time in SVM compared to other models.
- The kernel used in this SVM is of linear type.
- The overall accuracy achieved was 98 percent. But, we consider the percentage of fraudulent detection (**Recall**)detected correctly as an accuracy measure, we get 65.83% as accuracy for this model, when the dataset is not normalized.
- With K-Fold cross validation,the overall accuracy decreases to 53% and the Recall decreases to 40% without normalization.

Support Vector Machine

- With normalized data, the fraudulent accuracy (Recall)increases to 80 percent from 65%. The overall accuracy stays above 99% as compared to 53% without normalization.
- With K-fold cross validation, the recall for normalized data is 78%, better than 40% in the previous case.

Sequential Neural Network

We have implemented 3 layer sequential feed-forward neural network.

The first layer consists of 29 nodes, the second layer has 8 nodes and the final layer has got a single node corresponding to the output class.

We have used ‘relu’ function as the activation function in the first two layers, and sigmoid function in the last layer.

We have chosen the batch size as 10000, and number of epochs as 20.

We chose ‘binary cross entropy’ as loss function and ‘adam’ optimizer in this model.

We received 99 percent overall accuracy on this model for this particular dataset and 84.76%. recall for normalized data , 73% accuracy for dataset without normalization.

K-Nearest Neighbors

- 29 features were used for this method, dropping Time and Class.
- The output class columns were removed since we are using an unsupervised model here, and only X was considered for training the model
- We got overall accuracy of 99.91% upon training.
- Accuracy for fraudulent transactions(**Recall**) was equal to 63.994%
- With Normalization, there is slight decrease in accuracy. (From 99.94% to 99.66%)
- One of the drawbacks of KNN despite its high accuracy is, it takes alot of time to execute.

Gaussian Naive Bayesian

- We are taking 27 features available from 29, dropping Time and Class.
- Training and Testing Dataset are in the ratio 7:3
- We are getting accuracy around 97.82%.
- The test size does not affect the accuracy score.
- However, the test size should be positive and smaller than the sample size.
- On increasing or decreasing dataset, there is very small change in accuracy.
- Accuracy for Fraudulent Transactions(Recall) is 82%

Gaussian Naive Bayesian

- After Normalization, Accuracy increases by more than 1.5% i.e from 97.9% to 99.83%.
- Although, without Normalization, overall Accuracy of the model increases to 99.85%.
- There was also slight improvement in fraudulent transactions detected as it increased from 81.44% to 81.9%

Conclusion

- The Sequential neural network gave the best recall with normalized data.
- KMeans performs poorly on this dataset as compared to supervised learning.
- Logistic Regression and KNN had almost same accuracy and were next best methods.
- But Fraudulent Detection was better in Logistic Regression than KNN.
- Among supervised learning methods, accuracy of SVM is the least and it also takes long time to train.
- Significant impact was noticed with normalization and K-fold cross validation on the model accuracy.
- Also, the max_iter had a significant impact on the model accuracy.

Conclusion

	Logistic Regression	K-Means	SVM	Sequential Neural Network	K-NN	Gaussian Naive Bayes
Accuracy (Without Normalization)	**99.9%**	98.03%	98.79%	99.83%	99.91%	99.85%
Recall (Without Normalization)	64.35%	42.89%	65.83%	73.33%	59.34%	**81.90%**
Accuracy (With Normalization)	**99.94%**	69.26%	**99.94%**	99.82%	99.84%	99.81
Recall (Normalization)	80%	51.91%	78.75%	84.76%	-	-