# Introduction

A credit card is one of the most used financial products to make online purchases and payments. Though the Credit cards can be a convenient way to manage your finances, they can also be risky. Credit card fraud is the unauthorized use of someone else's credit card or credit card information to make purchases or withdraw cash. It is important that credit card companies are able to recognize fraudulent credit card transactions so that customers are not charged for items that they did not purchase. The dataset contains transactions made by credit cards in September 2013 by European cardholders. This dataset presents transactions that occurred in two days, where we have 492 frauds out of 284,807 transactions. The dataset is highly unbalanced, the positive class (frauds) account for 0.172% of all transactions.

# Module 1: Pipeline Process

In this section of the report, the methodology that has been adopted to achieve desired goals is discussed. Along with providing a good theoretical knowledge and understanding of the models and methods used, this section also discusses about the reasons that motivated to choose the specific methodology. The below subsections in this section explore the steps or methods employed to produce required results in this project by giving a detailed explanation of why these methods have been chosen and implemented.

Generally, the process of solving a problem in machine learning is divided into three or more stages. The first stage consists of performing Exploratory data analysis (EDA) followed by implementing the Machine Learning models, and finally evaluating the performance of models using certain metrics. This helps data scientists and analysts to produce desired results or expected outputs in solving a problem. The figure 1 shown below represents a flow chart that illustrates the proposed methodology. The below mentioned sections give an in-detailed explanation of the stages that have been adopted to implement this project.

Figure 1

*Stages of Pipeline Process*

Loading the Dataset

Performing Exploratory Data Analysis

Cleaning the Data

Model Building and Fitting on the Data

Machine Learning Models

Model Evaluation

Comparing the Models based on Accuracy scores

## Exploratory Data Analysis:

Exploratory data analysis often termed as EDA is the initial stage when addressing a problem using data science and machine learning technology. To put it in simple terms, it is the process of analysing the problem by exploring the data available that is related to the problem. Nowadays, data is being generated from many sources and most of the time, it is raw data, which means it contains many unnecessary factors that does not contribute to the problem. In most of the cases, the data available today may not always be structured, pure, or in ready to use form. Hence, it is essential to transform the data into required form according to the application by understanding and filtering it. This is where the exploratory data analysis stage comes into picture. For any problem to be solved using machine learning techniques, it is necessary to perform exploratory data analysis to gain better and true results. There are many advantages of performing EDA on the data being used, starting from knowing what the data represents to be able to use according to our requirements. By performing exploratory data analysis on a dataset, it becomes possible to gather various form of information of a particular dataset such as the number of columns, rows present, the type of data present in each column etc. In addition to this, EDA allows data scientists and analysts to find out if there are any missing values in the data and how they can be removed or filled according to their requirement. The process of exploratory data analysis also includes generating insights from the data using visualisations, i.e., it allows analysts to demonstrate the data in pictorial form. Some hidden trends present in the data can be observed, understood, and utilised according to the problem requirements by performing EDA. Exploratory data analysis is not only designed to examine the trends of a specific variable, but also to interpret the relationship between different variables present in the dataset. The correlation analysis allows the user to know how the variables are related to each other and how important they are in deciding the outcome of the problem. Some of the steps performed in this project as a part of EDA and the reasons these methods have been chosen are discussed below.

### Tools Required for EDA

There are various programming and non-programming tools that provide required functions and libraries to perform exploratory data analysis. Over the recent years, these tools have been expanded and developed various forms as per the required applications. Some of the tools that are extensively used for implementing EDA are Python, R, Microsoft Excel, etc. R and Python are the two programming languages that provide various kits and libraries which allow users to apply EDA effectively on variety of datasets. In this project, Python language has been used to perform exploratory data analysis. Python allows users to utilise some of the most famous libraries such as pandas, numpy, matplotlib, and seaborn. With the help of matplotlib and seaborn, data scientists and analysts produce visualisations which interpret the data. Visualisations include building charts such as boxplots, bar charts, histograms, violin plots, pie charts, etc. Additionally, correlation analysis among different variables in the dataset is represented pictorially using a graph called heatmap.

### Basic Statistical analysis

Statistical concepts hold an important role in designing and applying machine learning algorithms. Some of the concepts that help in understanding and solving data related problems are calculation of mean, median, minimum value, maximum value, first quartile, and third quartile. These statistical values provide a good description of all the columns present in the dataset. This helps in understanding the hidden patterns and trends present in the data.

### Univariate Analysis

The name Univariate analysis suggests that it is the process of analysing a single variable. Analysis of a single variable allows user to identify the trends present in that specific variable. It gives analysts and data scientists an opportunity to deep dive into understanding the importance of a variable. Univariate analysis includes visualising the variable data using box plots, violin plots, histograms, etc.

### Bivariate Analysis

Bivariate analysis is a type of analysis that consists of analysing two variables and interpreting the relation between them. With the help of bivariate analysis, the correlation among different variables present in the dataset can be understood. This analysis can be done between any two kind of variables such as numerical vs numerical, categorical vs categorical, categorical vs numerical. A bivariate analysis is usually performed by generating scatter plots, correlation coefficients, and regression analysis. By calculating the correlation coefficients between variables, the dependence of one variable on other can be measured.

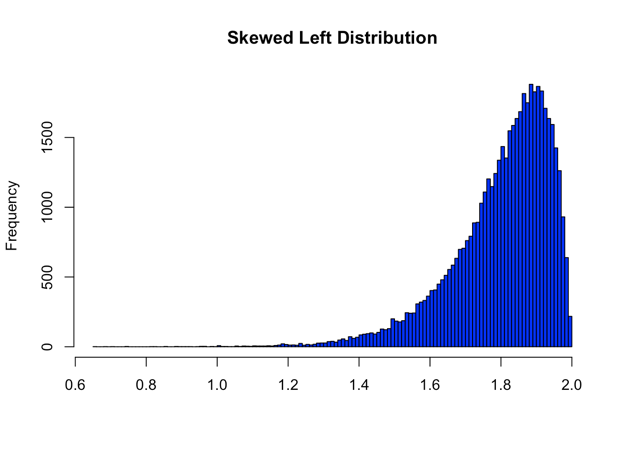
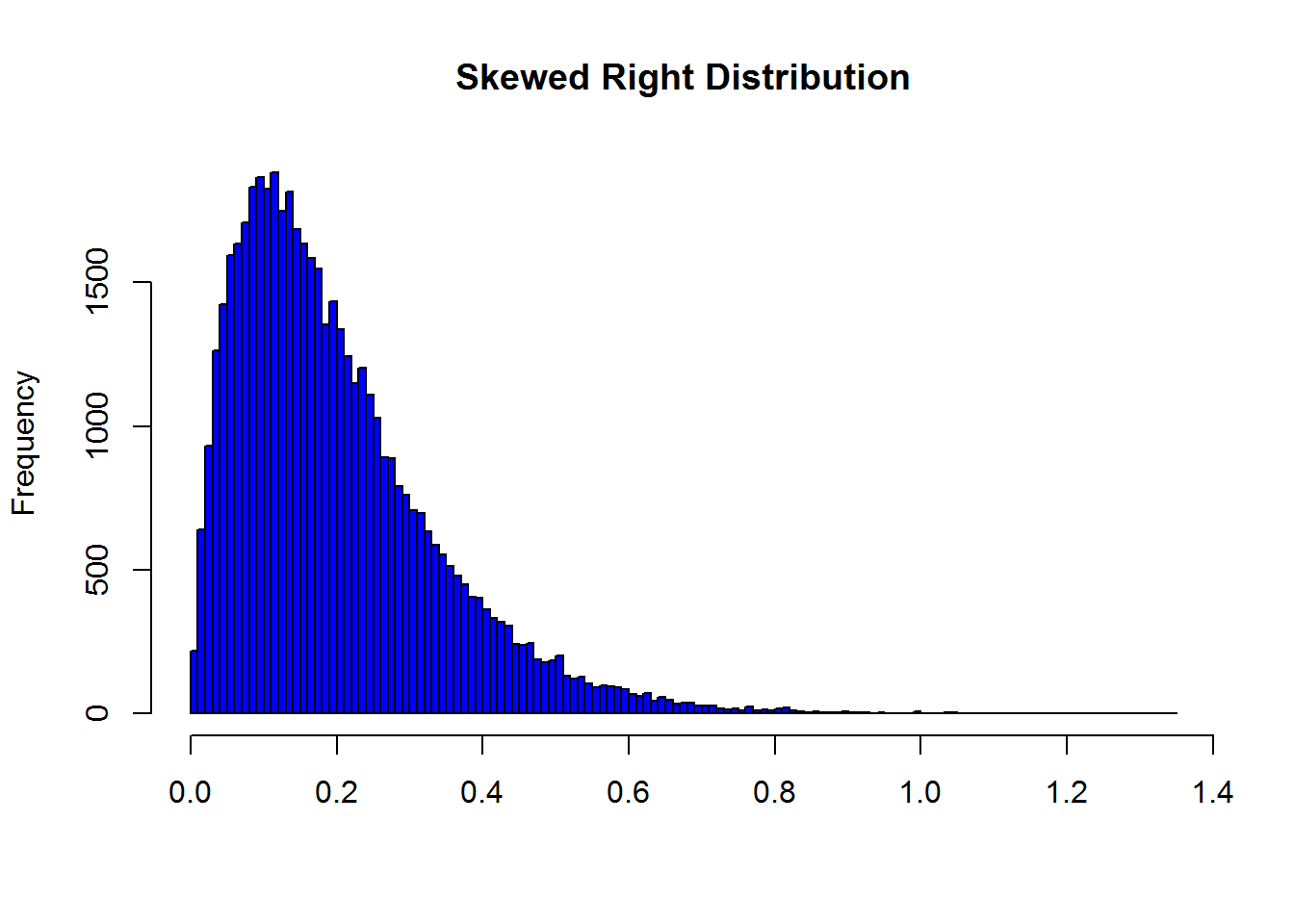
A brief description of the charts used in this project to perform exploratory data analysis are explained below.

* **Histogram:**

A histogram is a graphical representation of the distribution of a variable’s value in the dataset. It is basically a collection of bar graphs also called as bins. Histograms are widely used in most of the data representation applications. Histograms provide information about how skewed the data is i.e., if the data is left skewed, right skewed, or normally distributed. It also gives information about the presence of outliers and their distribution in the data. Histogram has a feature to choose the bin size when plotting the chart. It is essential to choose an ideal bin size when plotting a histogram because with the variance in bin size, the shape of the histogram changes which holds importance in representation of the data. Below shown figure 2 shows a graphical illustration of left, right and normally skewed histograms.

Figure 2

*Left, Right, and Normally Skewed Histograms*

 Chart, histogram

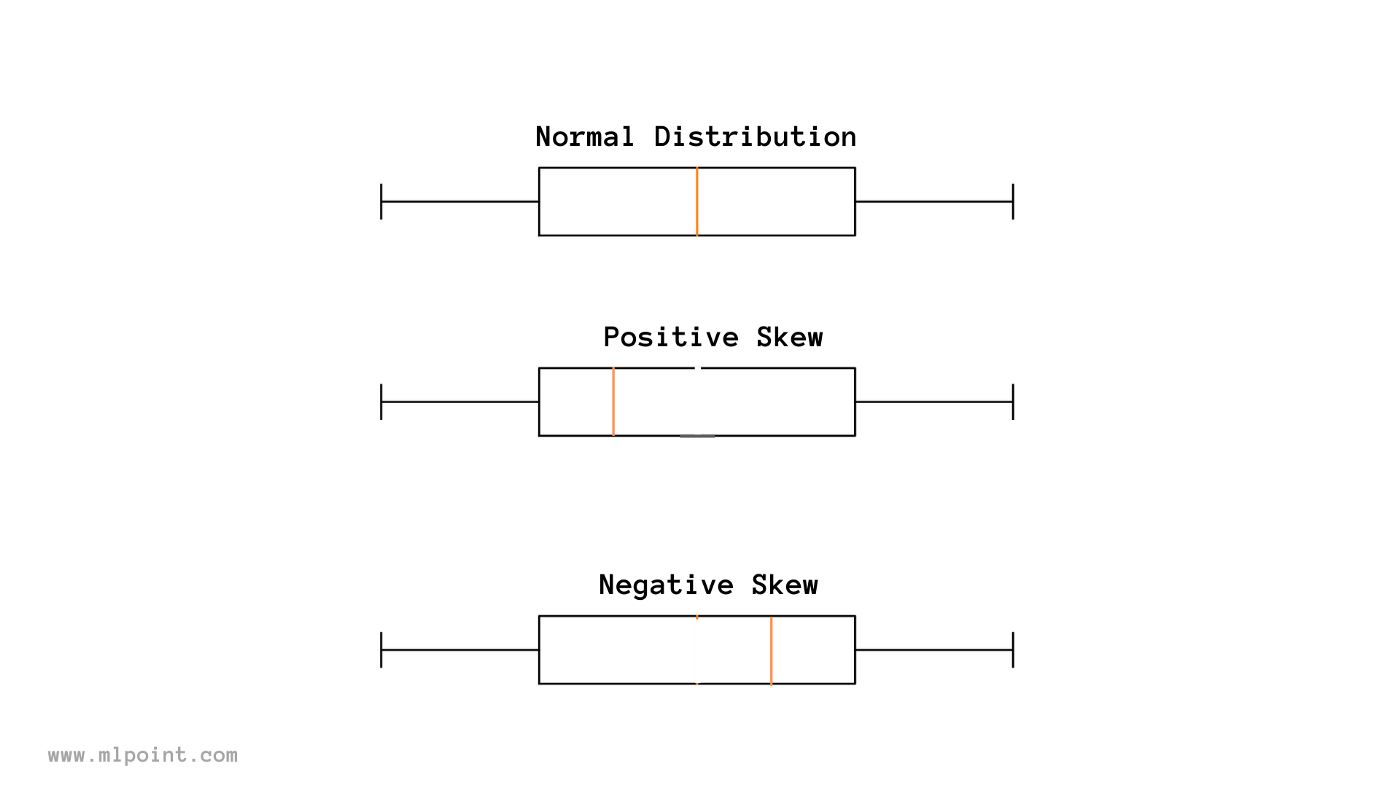
Description automatically generated

* **Boxplot:**

As discussed in the above sections, statistics play a vital role in interpreting various kinds of data. One of the most used graphs that explains the required information about the data is a boxplot. A boxplot is an illustrative representation of the concept of five number summary in statistics. A five number summary is a method where five different factors of a variable could be defined. The five factors that define a variable’s data and its pattern in the dataset are minimum value, first quartile, median value, third quartile, and maximum value of the variable. These five factors broadly describe a variable in a dataset. Boxplots are designed to present a visual depiction of the theory of five number summary in a graphical format. Boxplots are also called as Whisker plots because the structure of a boxplot is more like a box that shows the interquartile range, and the minimum, maximum values are connected by lines called as whiskers. Like the histograms, the fundamental aim of the boxplots is to make user understand how the data is distributed. Boxplots not only provide a clear view of the skewness of the data, but also exhibit comprehensible outlook about the outliers present in the data. The below shown figure represents normally, positively, and negatively skewed boxplots.

Figure 3

*Normal, Positive, and Negatively skewed Boxplots*



* **Pie Chart & Bar Graph:**

A pie chart is one of the basic visualisation charts that are used extensively in representing the data. The structure used in pie chart is basically a circle that is divided into different portions of areas called as pie. These areas represent the percentages of the data i.e., the size of the area is directly proportional to the percentages. More the percentage, more the area. Pie charts are used to demonstrate the quantity of a variable in a dataset.

Similar to a pie chart, bar graph is also a graphical illustration that is broadly used to represent the quantity of a variable in a data. But bar graph doesn’t use a circle to represent the data, instead it uses bars that describe the quantity of a variable. The higher the height of the bar, the higher the quantity. In this project pie chart and bar graphs are used to as a count plot that denotes the quantity or count of unique values present in a variable.

### Correlation Analysis

Correlation test or analysis is conducted on a dataset to learn the relation between any two numerical or categorical variables present in the dataset. Correlation between two variables defines the dependability of one variable on the other. By performing correlation analysis, the user can understand the trends of a variable and predict their future behaviour. With this, the importance of certain variables in predicting the outcome variables can be known. Correlation is generally calculated and measured using a feature called correlation coefficient. The correlation coefficient value ranges from -1 to 1, with value 0 being least correlated and value 1 being highly correlated. Based on the correlation coefficient value, the concept of correlation is divided into three types. Correlation coefficient value 0 denotes that no correlation exists between the two variables. Whereas values 1 and -1 indicate that the two variables are perfectly positive correlated and perfectly negative correlated. This correlation is often represented in the form a graph called as Heat map. A heat map basically represents the correlation coefficient value of all the variables by using a range of colours. The darker the colour, the more the variables are correlated and vice-versa. Seaborn library provides state-of-art tools to plot this heatmap with option to choose colours of our choice.

### Data Cleaning (Missing or Null Values)

With the vast amount of data available, the probability of presence of missing values and outliers in a dataset is more. In many cases the data collected or gathered cannot be hundred percent accurate and complete. For example, in a data that has been gathered by conducting a questionnaire in a company, it is possible that some of the questions would not be answered by few employees based on their interests. In such cases, data engineers treat the unanswered questions i.e., the missing values are either replaced by zeroes or a term called Null value. The presence of these null values in a dataset can affect the data characteristics in different ways. So, when working on a dataset, it is essential to identify the missing values present in the data and clean the data, making it ready to use for machine learning models implementation. By following this procedure in the exploratory data analysis part, more effective results can be generated. There are different ways in treating the null values present in a dataset. Based on the type of data available, the process of filling null values is divided into three types- mean, median, mode. As discussed in the above sections, mean, median, and mode are some of the basic statistical concepts that play a vital role in solving a data-related problem. If the data available is in categorical form, then the missing values are filled by calculating mode of that column or variable. If the data is in numerical format, the null values can be treated by filling them with mean or median of that column. When the data consists of significant number of outliers, it is ideal to use the median value to fill the missing values. For data without outliers, both mean and median can be used to fill the null values. This way, the incomplete data could be made ready for applying the necessary models.

### Outlier Analysis

In general, outliers are the value that is considered to deviate from the normal distribution of the data. In every dataset, the way the data is distributed is examined to understand the trends it possesses. It is highly possible that a good number of outliers can be present in a dataset. Sometimes, these outliers can have a significant effect on the data distribution. So, it is important to identify these outliers and treat them accordingly. Various methods are available to identify the outliers present in a dataset. Outliers can be detected by few visualisation techniques such as boxplots, histograms, and violin plots. In few cases, scatter plots also provide a good knowledge of outliers present in a dataset. There are few statistical methods used to detect outliers such as the calculation of z-score and identifying the inter quartile range. After the outliers are detected, the dataset is treated by removing them, making it more effective for the implementation of machine learning algorithms.

## Module2: Machine Learning Algorithms

In recent years, Machine learning and data science techniques have become some of the most reached out trends in technological applications. Machine Learning is a concept of artificial intelligence and computer science where a machine is designed to learn as a human by using the data and building specific algorithms. The main aim of machine learning is to improve the accuracy of its models by applying them on various kinds of data. Today, data is found everywhere arou us, and it holds a lot of value if used appropriately. With variety of data being generated every day, more and more research is being conducted in developing machine learning techniques to solve many real-world problems. And these algorithms if implemented correctly, are producing state-of-the-art results. Machine learning is a combination of numerous concepts such as data analysis, statistics, probability, visualizations, etc. Over the past few years, data science and machine learning fields have seen a revolutionary growth in the way they are applied. Almost every sector today has applications of machine learning in solving different problems. Machine learning has been giving promising results to some of the complex issues being faced in the health sector. Many long-term diseases such as cancers and tumours are being detected and treated early with the help of machine learning algorithms. Basically, machine learning algorithms are divided into three types- Classification, Regression, and Clustering techniques. These techniques are further divided into many algorithms depending on the characteristics they possess. In this project, one of the most concerning long-term illnesses known as diabetes is predicted using machine learning techniques. The following sub-sections discuss various machine learning algorithms applied on the data to gain desired results in this project.

### Decision Tree:

Decision tree algorithm is known to be one of the most popular machine learning algorithms. It is a kind of supervised learning algorithm which uses previous data to make future decisions. Despite being a type of supervised algorithm, decision tree holds the ability to perform both classification and prediction problems. Depending on the type of target variable, decision tree learns to either classify or predict the output. The class of the dependent or the target variable is predicted or classified based on the independent variables available. For minute problems, decision tree can be used to visualise the decisions at every step. But, for real world data, it seems impossible to perform decision tree algorithm manually. So, machine learning provides a popular library called as scikit-learn which allows the implementation of several classification, regression, and clustering algorithms. To design a decision tree using scikit-learn library, there are few factors to be considered such as Criterion, Entropy, Gini gain, Gini index, Information gain, max\_depth, etc. These factors have been created to improve the accuracy results of decision tree. Different tree methods are available for application depending on the type of target variable, i.e., continuous, or categorical. The four types of decision tree methods are CART, CHAID, C4.5, and QUEST. The merits of using decision tree in this project are because of its ease of understanding and interpreting, its feature to simplify the relationship among dependent and independent variables. Decision tree algorithms are extensively used in the bank sector, health sector, customer analytics, and employee analytics.

### Logistic Regression:

Logistic Regression is one of the supervised learning algorithms in machine learning. Like Decision Tree algorithm, it works on the principle of classification technique. It uses the concept of probability to perform its operations by analysing the outcome when the target variable can be yes or no or in binary form (0&1). It has proved that Logistic Regression is one of the algorithms with most effective and accurate results. Unlike in Decision tree, the handling of missing values may not be possible in Logistic Regression.

### Random Forest Classifier:

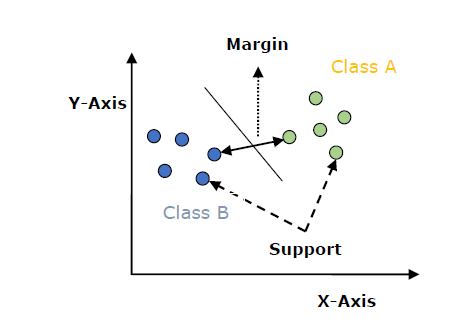
Random Forest is one of the widely used machine learning algorithms that has a proven history of generating productive results. It is a type of supervised learning algorithm that can be used as both classification and regression technique. Random Forest algorithm is a kind of ensemble models available which use multiple decision trees to form a single model. The name Random Forest suggests that it is a combination of random number of trees that forms into a forest. It outperforms the decision tree algorithm in things like over-fitting problem, and in improving accuracy. In Random Forest algorithm, the outcome is predicted by calculating the mean or average of all the decision trees used. For most of the machine learning algorithms, the accuracy could be increased by applying hyper parameter optimization technique, but Random Forest has the capability to produce best accuracy scores without using such technique. It is known for handling missing values better than decision tree algorithm.

### Support Vector Machine:

Support Vector Machine (SVM) is a most popular supervised machine learning algorithm. It is used for both classification and prediction problems. It can also be used as an unsupervised algorithm when the data is unlabelled. In case of unsupervised algorithm, support vector machine uses a clustering technique. Support Vector Machine uses a concept where the data is segregated into two different categories by drawing a line in between them. This line helps in deciding which category the new incoming data should be categorised as. Support vector machine is also applicable for image and text classification problems. For example, to identify whether a new image is a cat or dog, SVM can be trained on a dataset with many cat and dog images and by drawing a line between both these categories, the new image will be classified into category that shares more similarity.

Figure 4

*SVM dividing data into classes A & B using hyperplane*

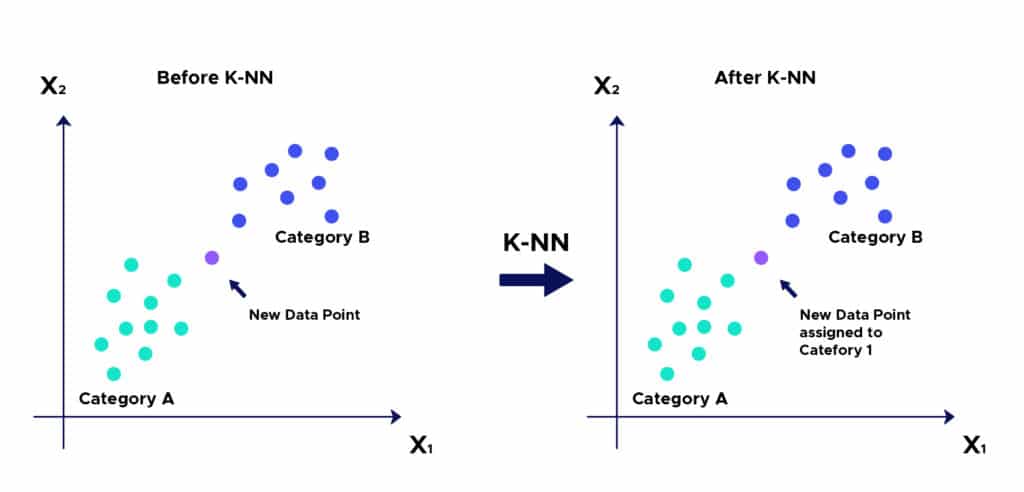


### K Nearest Neighbour:

K Nearest Neighbour usually referred as KNN is a type of supervised machine learning algorithm that is designed to solve both classification and regression problems. But in most of the cases, it is used as a classifier because of its features. It is known for its unique characteristic for being a non-parametric method in machine learning as it does not learn from the parameters in the data, but rather classifies the data based on the nearest point. It is also often referred to as a lazy learner among all the machine learning algorithms in view of the fact that it doesn’t train on the data when provided before. Instead, it stores the data when applied to it until any queries are performed on it. Assume the data is divided into two groups X and Y. Depending on the position of the new incoming data point, the distance is calculated to other nearby points. The point with the lowest Euclidian distance is selected as the target group (X or Y) of the new data point. The K in KNN denotes the number of points to be considered to calculate the distance. If K is 5, then distances between five new data points is calculated and the point with lowest distance is the chosen group or class. Choosing the K value can be experimental as there are no specific criteria followed in determining the K value. But it is recommended to choose the K value such that it’s not too low or too high. This can sometimes cause problems such as effect of outliers and over-fitting. The below shown figure 5 illustrates the working of KNN algorithm.

Figure 5

*Working of KNN algorithm*



## Module3: Evaluation Metrics

Developing machine learning models is based on the idea of constructive feedback. The main goal in implementing machine learning algorithms on different types of data is to understand how well a model performs on the data. Researchers found out various methods to calculate the performance of a machine learning model that are often called as Evaluation metrics. These metrics differ from model to model based on the problem to be solved. Evaluation metrics are not only used to compute a model’s performance, but they also allow the user to compare the model with other models. This way the user can decide which model performs best on the selected data. There are several most used evaluation or performance metrics such as Confusion matrix, Gini coefficient, F1 score, Gain and Lift charts, AUC – ROC, Kolmogorov Smirnov chart, Log Loss, Concordant – Discordant ratio, Cross-validation, and Root mean squared error. In this project, to analyse both machine learning and deep learning models performance, some of the standard metrics are used which are discussed below.

### Confusion Matrix:

To put in simple terms, a confusion matrix is a representation that generally consists of four different scores or values. These four values are True Negative (TN), True Positive (TP), False Negative (FN), and False Positive (FP). It basically represents the count of all these values. Confusion matrix is one of the most commonly used metrics to evaluate many classification algorithms.

Table 1

*Confusion Matrix*

|  |  |  |
| --- | --- | --- |
| Predicted Values | | |
| 0 | | 1 |
| Actual Values | 0 | TN | FP |
| 1 | FN | TP |

* **True Positive (TP):**

TP is the count of positive values that have been predicted correctly as positive. In this project, TP indicates the count of correctly predicted number of females with diabetes i.e., outcome 1 is predicted as 1.

* **True Negative (TN):**

TN is the count of negative values that have been predicted as negative by a model. In this project, TN denotes the number of correctly predicted females without diabetes i.e., outcome 0 is predicted as 0.

* **False Positive (FP):**

False positive is a number that denotes the count of negative values that are predicted as positive by a model. In our project, it represents how many females without diabetes (0) have been predicted as with diabetes (1). A False positive value sometimes is also termed as Type 1 error.

* **False Negative (FN):**

False negative is a value that indicates the count of positive values that are predicted as negative by a model. FN in our project indicates the number of times females with diabetes (1) predicted as females without diabetes (0). A False negative value sometimes is also termed as Type 2 error.

Based on the TP, TN, FP, FN values, various performance metrics can be calculated which are described below.

### Classification Report:

A Classification Report is kind of a tabular representation of various evaluation metrics available in machine learning. This can be implemented in python language using a famous library provided by Scikit-learn called as ‘classification report’. The classification report mainly consists of five metrics that are Accuracy, Precision, Recall, F1-Score, and Support. All these metrics are calculated based on the values provided by the Confusion matrix. A detailed explanation of the metrics displayed in a classification report is given below.

* **Accuracy:**

Accuracy of a model is calculated mathematically by dividing the total number of truly predicted values to the total predicted values. So, it is basically the ratio of the sum of True positive and True negative values to the sum of total values predicted.

Accuracy =

* **Precision:**

Precision, sometimes called as Positive predictive value is the ratio of actual positive values (TP) to the sum of total positive values (TP+FP) predicted by a model.

Precision =

* **Recall:**

Recall which is also known as Sensitivity is an evaluation metrics which is the ratio of correctly predicted positive values (TP) to the sum of actual positive values (TP+FN).

Recall =

* **F1-Score:**

F1-Score is considered as one of the important metrics in most of the cases due to its ability to use both Precision and Recall values. It is the weighted mean of both Precision and Recall values with its score ranging from 0 to 1, where 0 is the worst and value near to the 1 is considered as a good score. F1-Score is also called as F-measure.

F1-Score =

### AUC-ROC Score:

As discussed in the above section, confusion matrix offers several calculation parameters to analyse a model’s performance such as sensitivity, specificity, true positive, false positive. These parameters can be used in a different evaluation criterion known as AUC-ROC score which utilises visualisation as the method of representation. This score is mainly used in evaluating classification algorithms as it provides a clear understanding of a model’s ability to differentiate classes. Receiver Operating Characteristics curve (ROC) is a graphical representation metric that is formed by plotting a graph between True Positive rate (TPR) and the False Positive rate (FPR). Area Under Curve (AUC) is a metrics that is mostly used for binary classification problems. It computes the area under the ROC curve and this score determines the performance of the model in distinguishing both the classes. The higher the AUC score, the better the model’s performance.

# Module 4: Requirements

In this section of the report, a clear explanation of the dataset and the system features that are required to generate the end results is provided. It not only also discusses about the availability of the data used for this project, but also gives information about how the data is collected. This section describes how the system features play an important role in executing this project.

## 4.1 The Dataset:

A credit card is one of the most used financial products to make online purchases and payments. Though the Credit cards can be a convenient way to manage your finances, they can also be risky. Credit card fraud is the unauthorized use of someone else's credit card or credit card information to make purchases or withdraw cash. It is important that credit card companies are able to recognize fraudulent credit card transactions so that customers are not charged for items that they did not purchase. The dataset contains transactions made by credit cards in September 2013 by European cardholders. This dataset presents transactions that occurred in two days, where we have 492 frauds out of 284,807 transactions. The dataset is highly unbalanced, the positive class (frauds) account for 0.172% of all transactions.

## 4.2 Tools and Libraries Required:

To apply the machine learning or deep learning techniques effectively, different libraries and tools are available. These tools come with their own advantages and disadvantages depending on the area of application. Some of the most used programming languages in machine learning include Python, R, SQL, etc. Whereas depending on the languages used, various tools or software are available in the market such as Jupyter Notebook, Matlab, Rstudio, etc. All these software tools come with a set of libraries that are applicable in different ways. The programming language, software, and libraries that have been used to implement this project are discussed below

### Python:

Python is an open-source programming language that has become immensely popular in recent years for application in various fields. It is known for its ability to write programs in user-friendly language. Python has a wide range of applications such as in web development, Data Science, Internet of Things, Artificial Intelligence etc., Recently, python has been known for its usage as a primary language in implementing machine learning, and deep learning technologies. Starting from data collection to data analysis, and visualisations, python has become extremely popular for its ease of use. Python also allows users to perform all kinds of analysis on vast and variety of datasets. In this project, Python has been used as the main programming language to implement necessary machine learning tasks and achieve the expected results. In this project the version of python language used is 2.7.16. Python provides and allows users to analyse the data using various libraries which are discussed in the further sections below.

### Jupyter Notebook:

Jupyter Notebook is an open-source, integrated development environment that is available free both online and offline. Jupyter Notebook possess the quality of allowing to write, share, and execute already built codes. It allows users to write codes in different programming languages such as python, R, Scala, Julia, etc., While there are many available platforms to write and execute programs in python, Jupyter Notebook is considered one of the most widely used platform for implementing python due to its ease of availability, ability to perform user interface visualisations etc., In this project, python language along with Jupyter notebook has been used to accomplish the data analysis and machine learning tasks required to solve the problem.

### Pandas and NumPy:

Pandas and NumPy are some of the most successful and extensively used open-source libraries in data pre-processing stage of analysing a dataset. Pandas provide an excellent option to load data in a Data Frame format. It allows users to perform multiple actions on rows and columns in a Data Frame. Whereas NumPy is known for its ability to allow users to perform different numerical operations on data. These numerical functions include arithmetic operations, descriptive statistics etc.,

### Seaborn and Matplotlib:

Data visualisation is one of the popular methods of representing the data. It is a crucial component of data science that helps people comprehend and interpret complex data. Matplotlib and Seaborn are the foundations of data visualisation using Python. Matplotlib is mostly used to plot two dimensional graphs of arrays and create statical interferences such as histograms, box plots, violin plots, pie charts, bar charts, etc., Seaborn library is known as an extended version of matplotlib library. It is used to perform univariate and bivariate analysis by creating beautiful visualisations. It is also used in generating pair plots, heatmaps, time forecasting, etc.

### Scikit-learn:

Scikit-learn library is the most significant and core part of data science that is used to build and implement machine learning algorithms. It has a wide variety of features to choose ranging from ML algorithms to their metrics. All types of machine learning models are built in python by importing scikit-learn library. Additionally, scikit-learn library provides different tools to perform operations such as feature extraction and model evaluation.

# Module5: Analysis

In this section of the report, various tasks performed as a part of data analysis are discussed. It includes visualising the data, univariate, and bi-variate analysis, correlation analysis and plotting heatmap. Some of the mostly used graphical representations in interpreting the data such as histograms, count plots, pair plots, box plots, violin plots are presented in this section.

## 5.1 Descriptive Statistics:

Table 4

*Statistical Parameters calculated*

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Time** | **V1** | **V2** | **V3** | **V4** | **V5** | **V6** | **V7** | **V8** | **V9** | **...** | **V21** |
| **count** | 284807 | 2.85E+05 | 2.85E+05 | 2.85E+05 | 2.85E+05 | 2.85E+05 | 2.85E+05 | 2.85E+05 | 2.85E+05 | 2.85E+05 | ... |
| **mean** | 94813.86 | 1.76E-12 | -8.25E-13 | -9.65E-13 | 8.32E-13 | 1.65E-13 | 4.25E-13 | -3.05E-13 | 8.78E-14 | -1.18E-12 | ... |
| **std** | 47488.15 | 1.96E+00 | 1.65E+00 | 1.52E+00 | 1.42E+00 | 1.38E+00 | 1.33E+00 | 1.24E+00 | 1.19E+00 | 1.10E+00 | ... |
| **min** | 0 | -5.64E+01 | -7.27E+01 | -4.83E+01 | -5.68E+00 | -1.14E+02 | -2.62E+01 | -4.36E+01 | -7.32E+01 | -1.34E+01 | ... |
| **25%** | 54201.5 | -9.20E-01 | -5.99E-01 | -8.90E-01 | -8.49E-01 | -6.92E-01 | -7.68E-01 | -5.54E-01 | -2.09E-01 | -6.43E-01 | ... |
| **50%** | 84692 | 1.81E-02 | 6.55E-02 | 1.80E-01 | -1.98E-02 | -5.43E-02 | -2.74E-01 | 4.01E-02 | 2.24E-02 | -5.14E-02 | ... |
| **75%** | 139320.5 | 1.32E+00 | 8.04E-01 | 1.03E+00 | 7.43E-01 | 6.12E-01 | 3.99E-01 | 5.70E-01 | 3.27E-01 | 5.97E-01 | ... |
| **max** | 172792 | 2.45E+00 | 2.21E+01 | 9.38E+00 | 1.69E+01 | 3.48E+01 | 7.33E+01 | 1.21E+02 | 2.00E+01 | 1.56E+01 | ... |

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **V22** | **V23** | **V24** | **V25** | **V26** | **V27** | **V28** | **Amount** | **Class** |
| 2.85E+05 | 2.85E+05 | 2.85E+05 | 2.85E+05 | 2.85E+05 | 2.85E+05 | 2.85E+05 | 2.85E+05 | 284807 |
| -3.41E-13 | -5.72E-13 | -9.73E-13 | 1.46E-12 | -6.99E-13 | -5.62E-13 | 3.33E-12 | -3.52E-12 | 88.349619 |
| 7.35E-01 | 7.26E-01 | 6.24E-01 | 6.06E-01 | 5.21E-01 | 4.82E-01 | 4.04E-01 | 3.30E-01 | 250.120109 |
| -3.48E+01 | -1.09E+01 | -4.48E+01 | -2.84E+00 | -1.03E+01 | -2.60E+00 | -2.26E+01 | -1.54E+01 | 0 |
| -2.28E-01 | -5.42E-01 | -1.62E-01 | -3.55E-01 | -3.17E-01 | -3.27E-01 | -7.08E-02 | -5.30E-02 | 5.6 |
| -2.95E-02 | 6.78E-03 | -1.12E-02 | 4.10E-02 | 1.66E-02 | -5.21E-02 | 1.34E-03 | 1.12E-02 | 22 |
| 1.86E-01 | 5.29E-01 | 1.48E-01 | 4.40E-01 | 3.51E-01 | 2.41E-01 | 9.10E-02 | 7.83E-02 | 77.165 |
| 2.72E+01 | 1.05E+01 | 2.25E+01 | 4.58E+00 | 7.52E+00 | 3.52E+00 | 3.16E+01 | 3.38E+01 | 25691.16 |

The table 4 shown above is the output obtained when the statistical description operation has been performed on the dataset. This is achieved by using a function in python called “.describe()”. As mentioned in the table 4, different statistical parameters such as count, mean, median, standard deviation, minimum, max values, 25th, and 75th quartile values of each variable are calculated. In the table 5 shown below, the type of data every variable consists is obtained by applying a function on the dataset called “.info()”.

Table 5

*Data Types of Variables*

0 Time 284807 non-null float64

1 V1 284807 non-null float64

2 V2 284807 non-null float64

3 V3 284807 non-null float64

4 V4 284807 non-null float64

5 V5 284807 non-null float64

6 V6 284807 non-null float64

7 V7 284807 non-null float64

8 V8 284807 non-null float64

9 V9 284807 non-null float64

10 V10 284807 non-null float64

11 V11 284807 non-null float64

12 V12 284807 non-null float64

13 V13 284807 non-null float64

14 V14 284807 non-null float64

15 V15 284807 non-null float64

16 V16 284807 non-null float64

17 V17 284807 non-null float64

18 V18 284807 non-null float64

19 V19 284807 non-null float64

20 V20 284807 non-null float64

21 V21 284807 non-null float64

22 V22 284807 non-null float64

23 V23 284807 non-null float64

24 V24 284807 non-null float64

25 V25 284807 non-null float64

26 V26 284807 non-null float64

27 V27 284807 non-null float64

28 V28 284807 non-null float64

29 Amount 284807 non-null float64

30 Class 284807 non-null int64

## Target Variable:

In every problem, the variables present in a dataset can be divided into two types- independent and dependent variables. As mentioned in the section 4, the dependent variable considered in the present dataset is the Class variable. It is important to explore the target variable to understand more about what the algorithms need to classify or predict. So below mentioned is some of the analysis performed on the Class variable. The table 6 shown below demonstrates the properties of the Class variable. We can see from the table that the Class variable has two types of classes 0 and 1. The transactions is represented with 1 is fraud and without Non fraudulent with 0. The dataset has a higher percentage of non- fraudulent compared to Fraudulent Transactions.

Table 6

*Classes in Output variable*

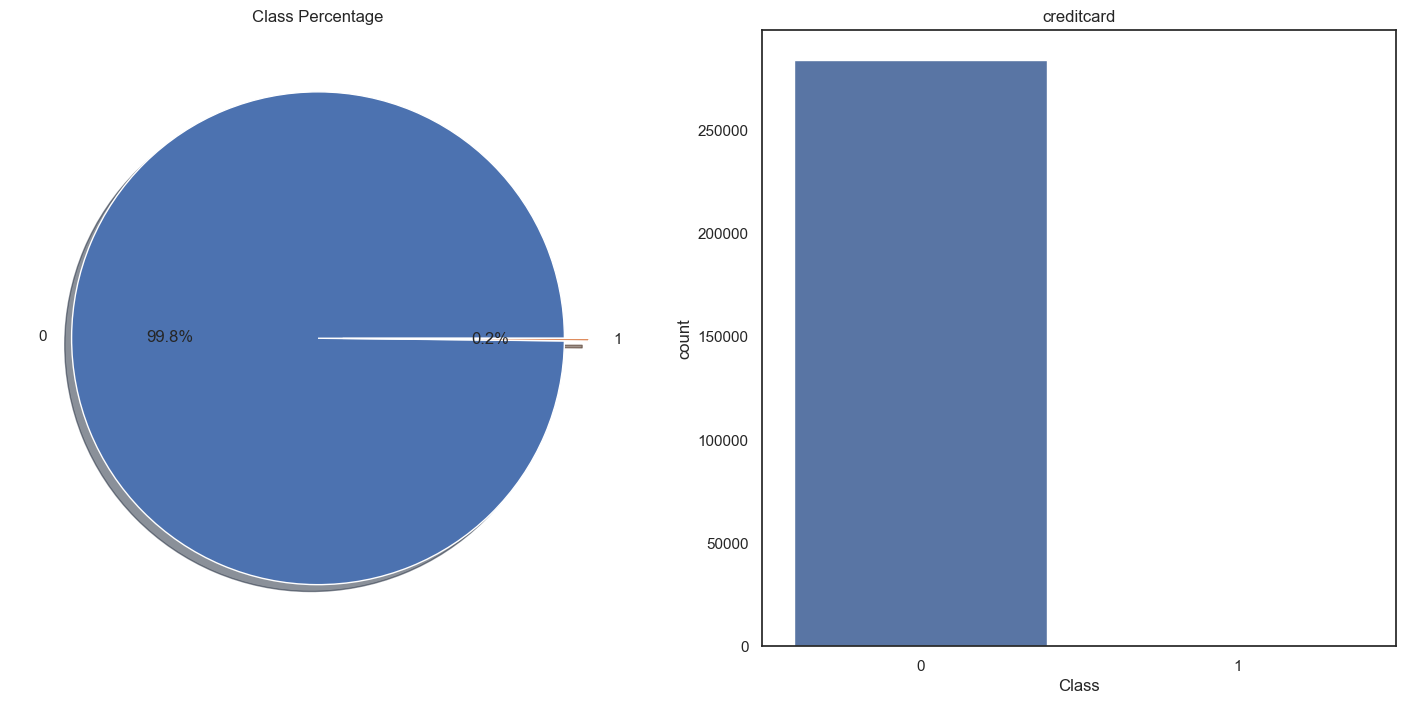
|  |  |  |
| --- | --- | --- |
| **Target Variable: Class** | | |
| Type of class | 0 (non-fraudulent) | 1 (fraudulent) |
| Number of values | 284315 | 492 |
| Percentage of values | 99.827% | 0.172% |

### Visualising Target variable:

The figure 6 mentioned below are a representation of the pie chart and bar graph of Class variable. These visualisations are obtained by importing the matplotlib library.

Figure 6

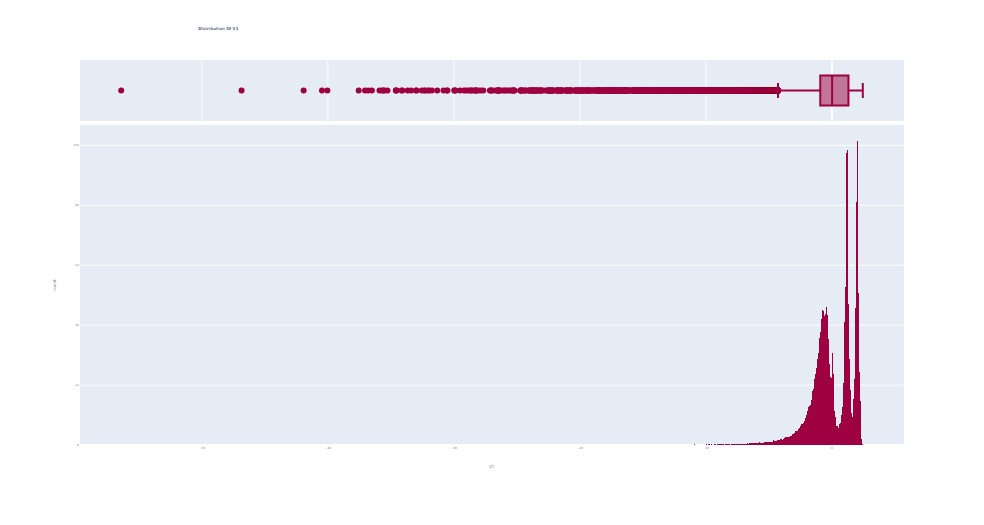
*Pie and Bar charts of Class Variable*



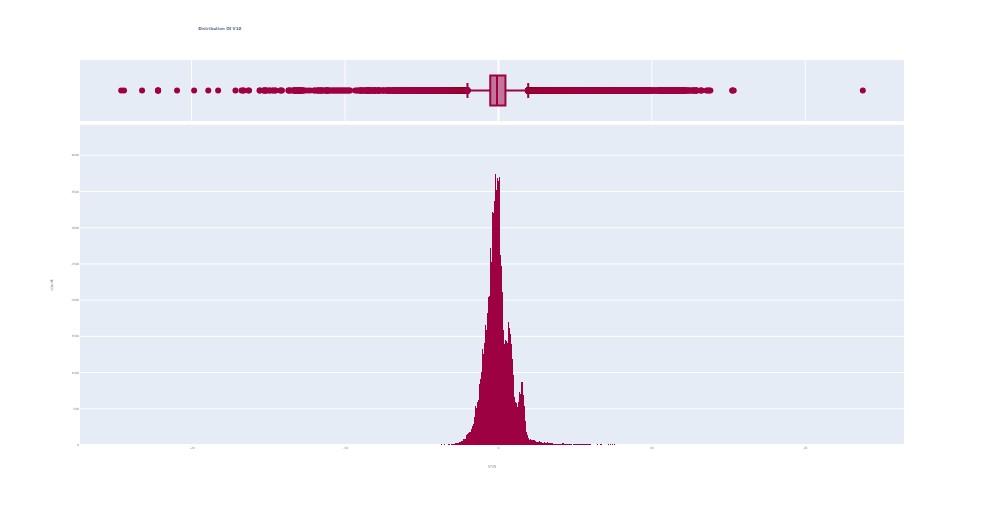
## Univariate Analysis:

As a part of exploratory data analysis, univariate analysis has been performed on all the variables in the dataset. By performing univariate analysis, the trends, and variations a variable possess can be understood. In the figures mentioned below, both boxplot and histogram used to visualise the variables is shown.

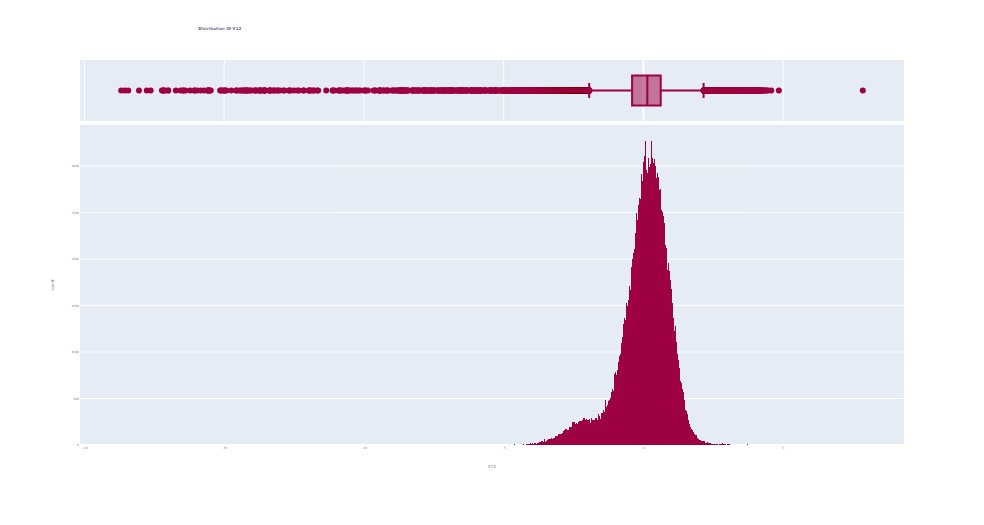
Distribution of V1:



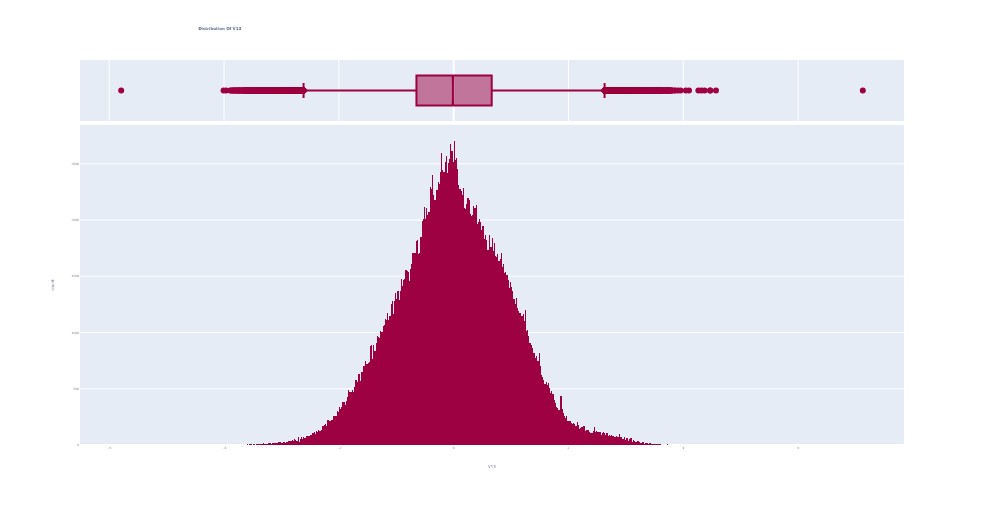
Distribution of V10:



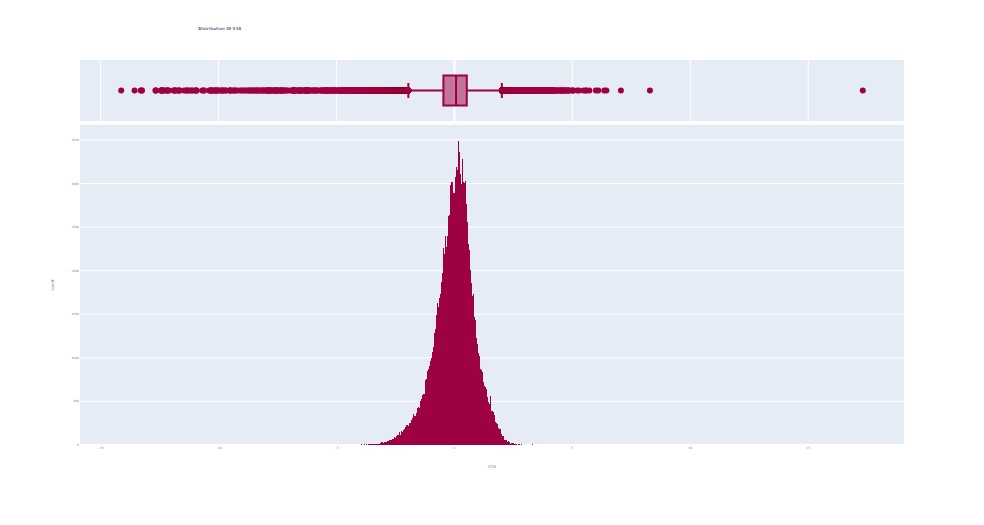
Distribution of V12:



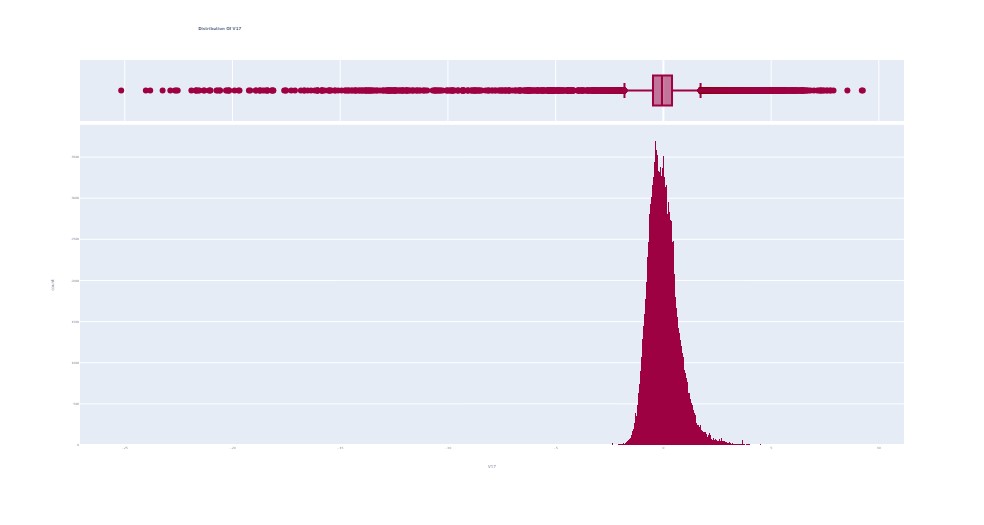
Distribution of V13:



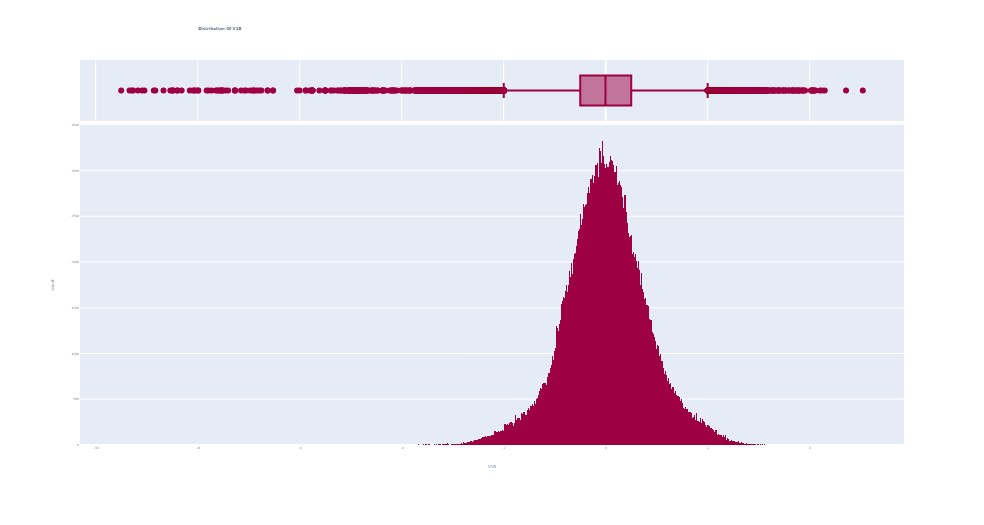
Distribution of V16:



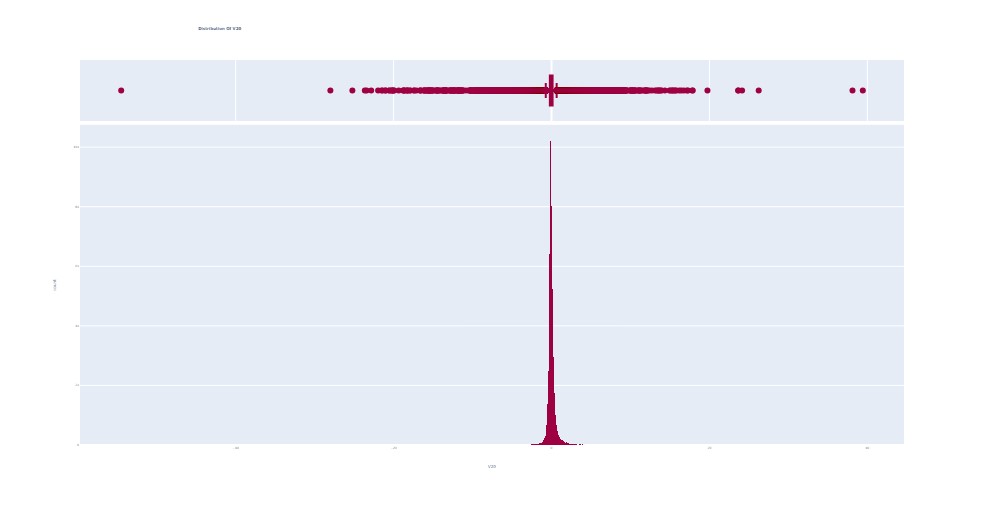
Distribution of V17:



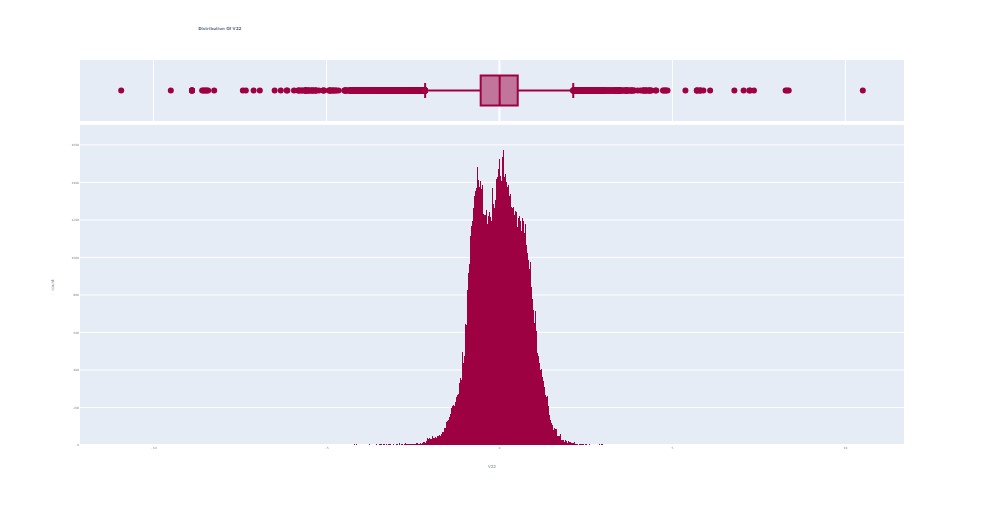
Distribution of V18:



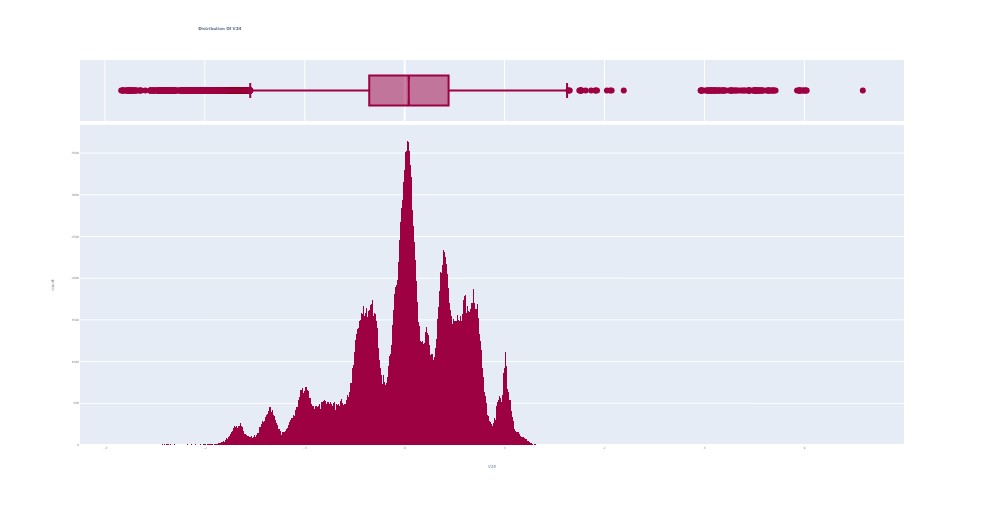
Distribution of V20:



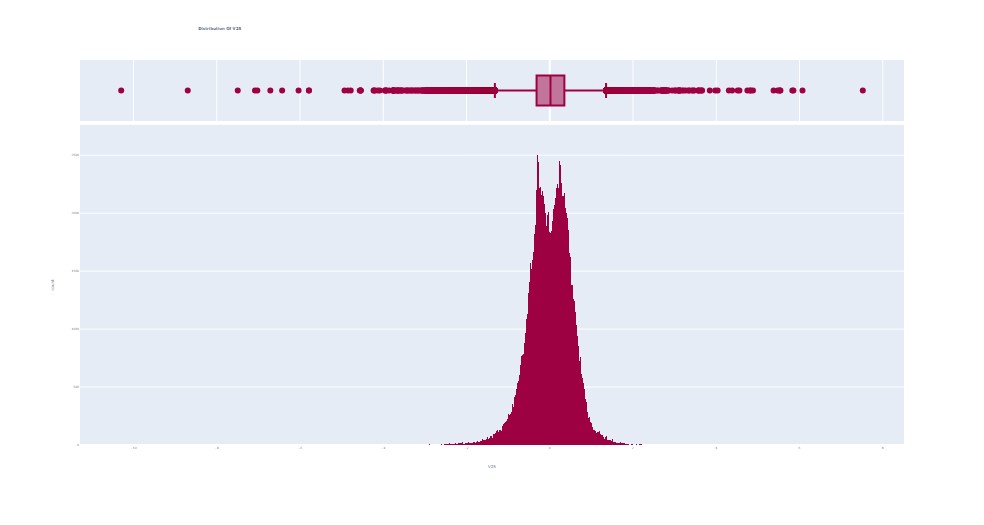
Distribution of V22:



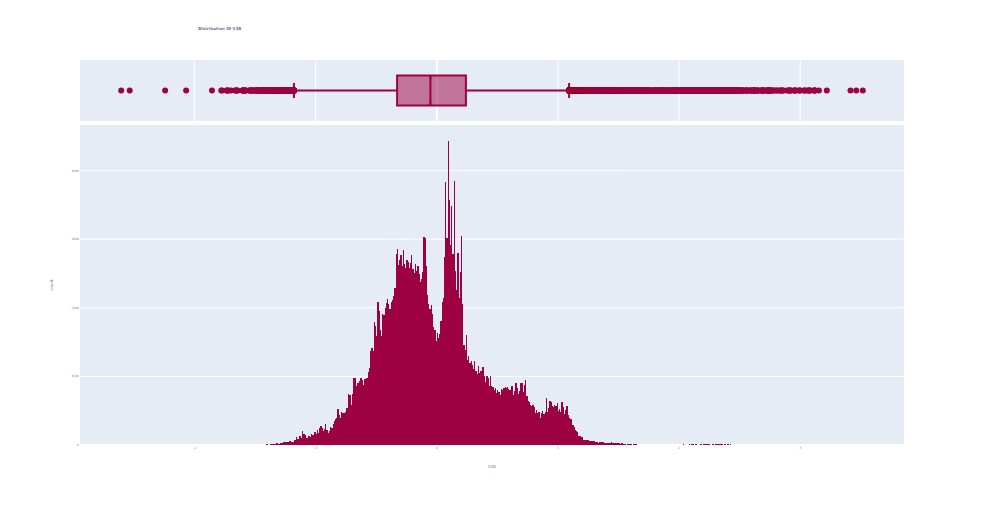
Distribution of V24:



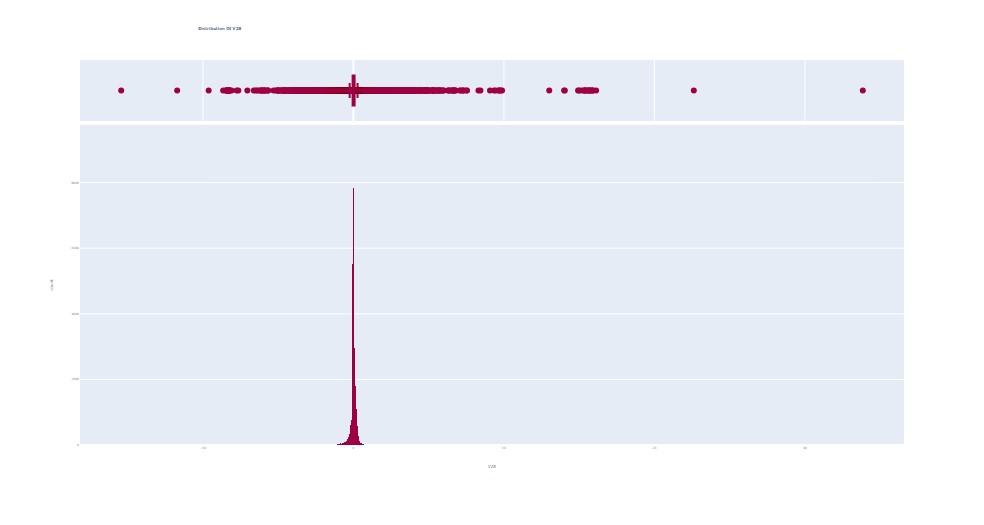
Distribution of V25:



Distribution of V26:



Distribution of V28:



## 5.4 Outlier Analysis:

Every variable in the dataset consists of few outliers which have been observed by performing the univariate analysis in the above section. The table 7 shown below represents the range after which the outliers are present in each variable. The values above these ranges are considered as outliers and they have been removed successfully. After removing the outliers, the size of the dataset has been reduced from (284807, 31) to (284602, 31).

Table 7

*Range of Outliers in each variable*

|  |  |
| --- | --- |
| **Variables** | **Outliers value** |
| V1 | <-40 |
| V2 | <-50 |
| V4 | >14 |
| V6 | >25 |
| V8 | <-42 |
| V10 | >10 |
| V12 | >5 |
| V13 | >5 |
| V16 | >10 |
| V17 | >10 |
| V18 | >5 |
| V20 | <-30 |
| V22 | >9 |
| V26 | >13 |
| V28 | >20 |

Table 8

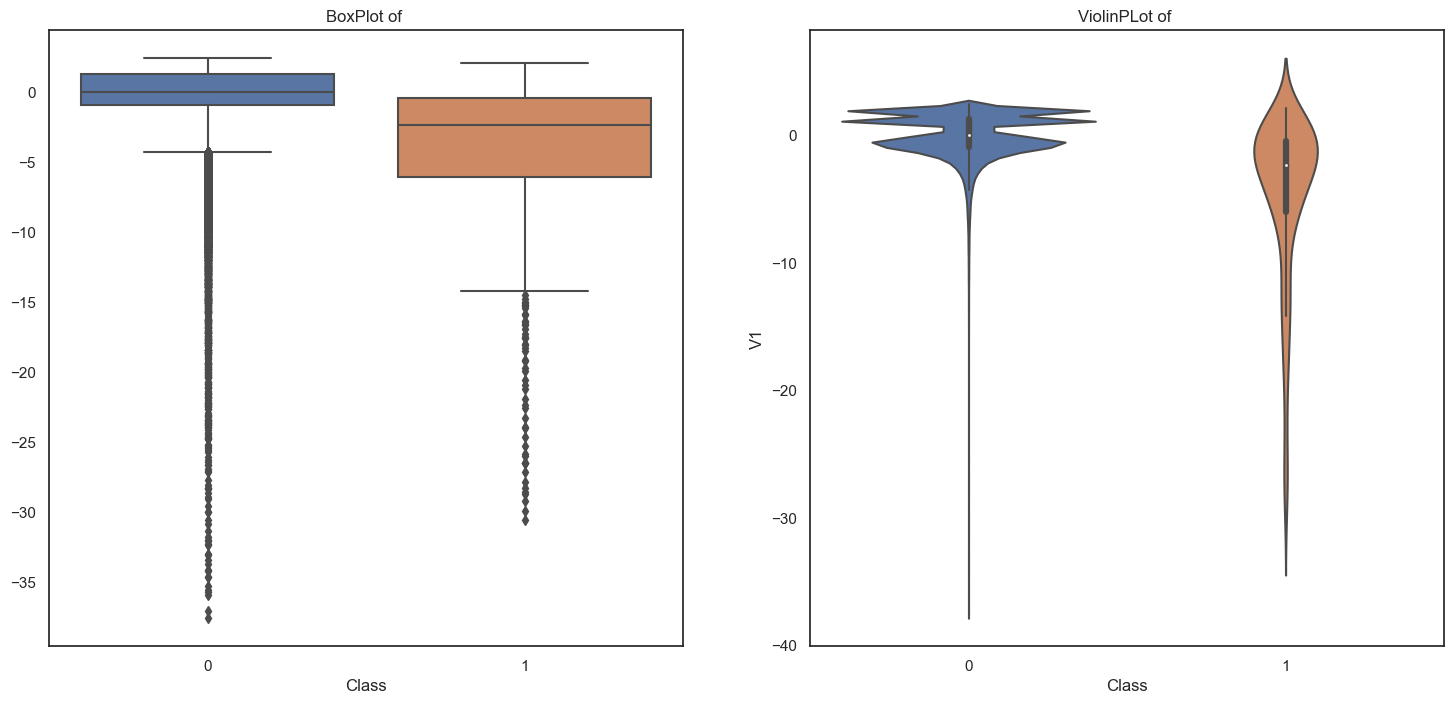
*Size of data before & after outlier treatment*

|  |  |
| --- | --- |
| **Outlier analysis** | **Size of the dataset** |
| Before Outlier removal | (284807,31) |
| After Outlier removal | (284602,31) |

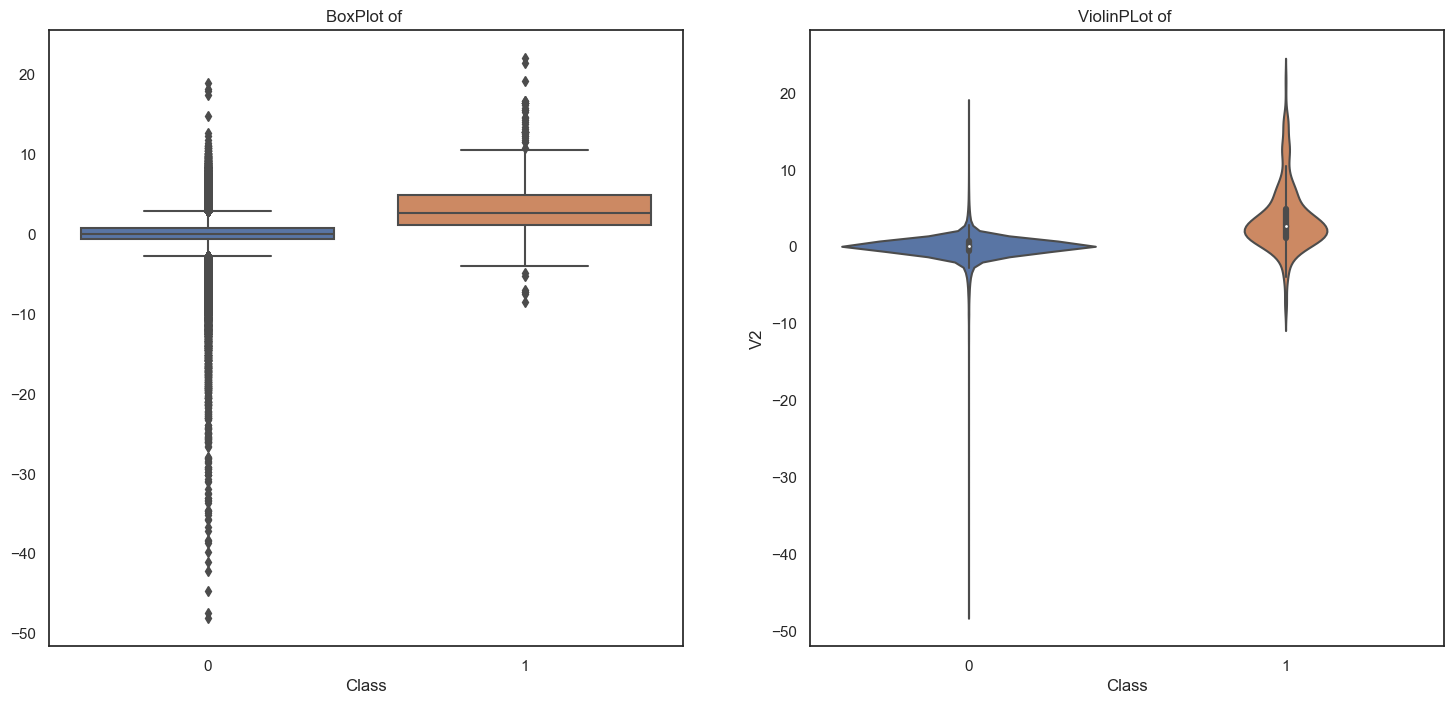
## 5.5 Bi-variate analysis:

Bi-variate analysis is the analysis of each variable’s behaviour with respect to the target variable. In this project, bi-variate analysis has been performed by visualising the data using box plots and violin plots. Below mentioned figures are the outputs obtained by importing seaborn library in python.

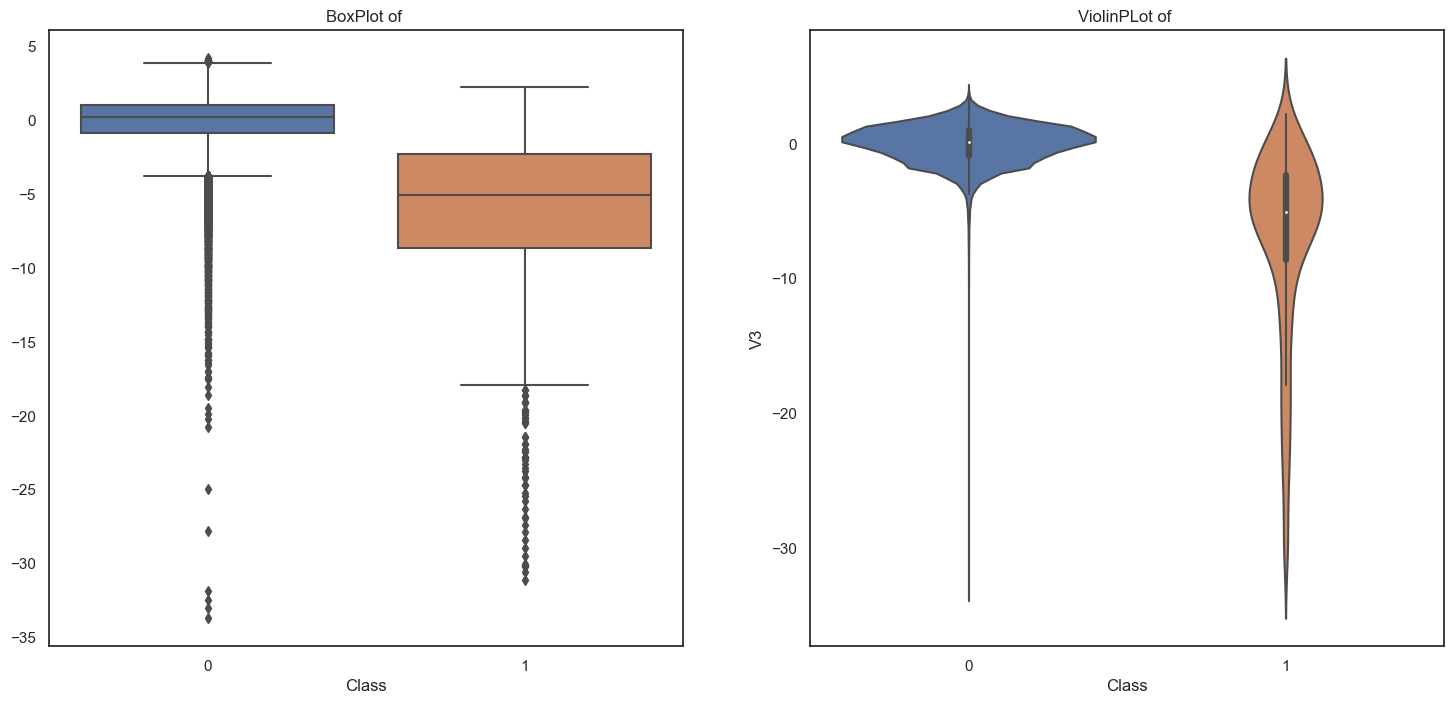
Plot between V1 vs Class



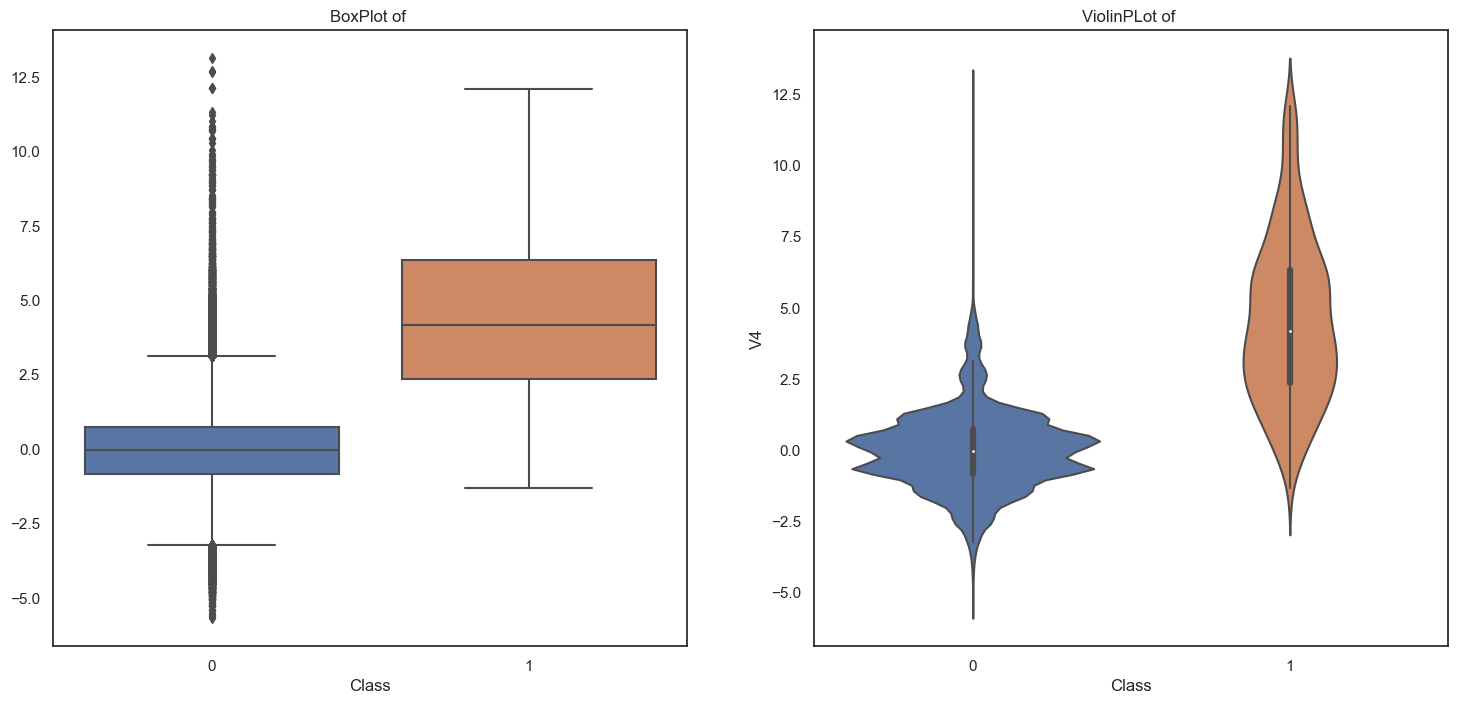
Plot between V2 vs Class



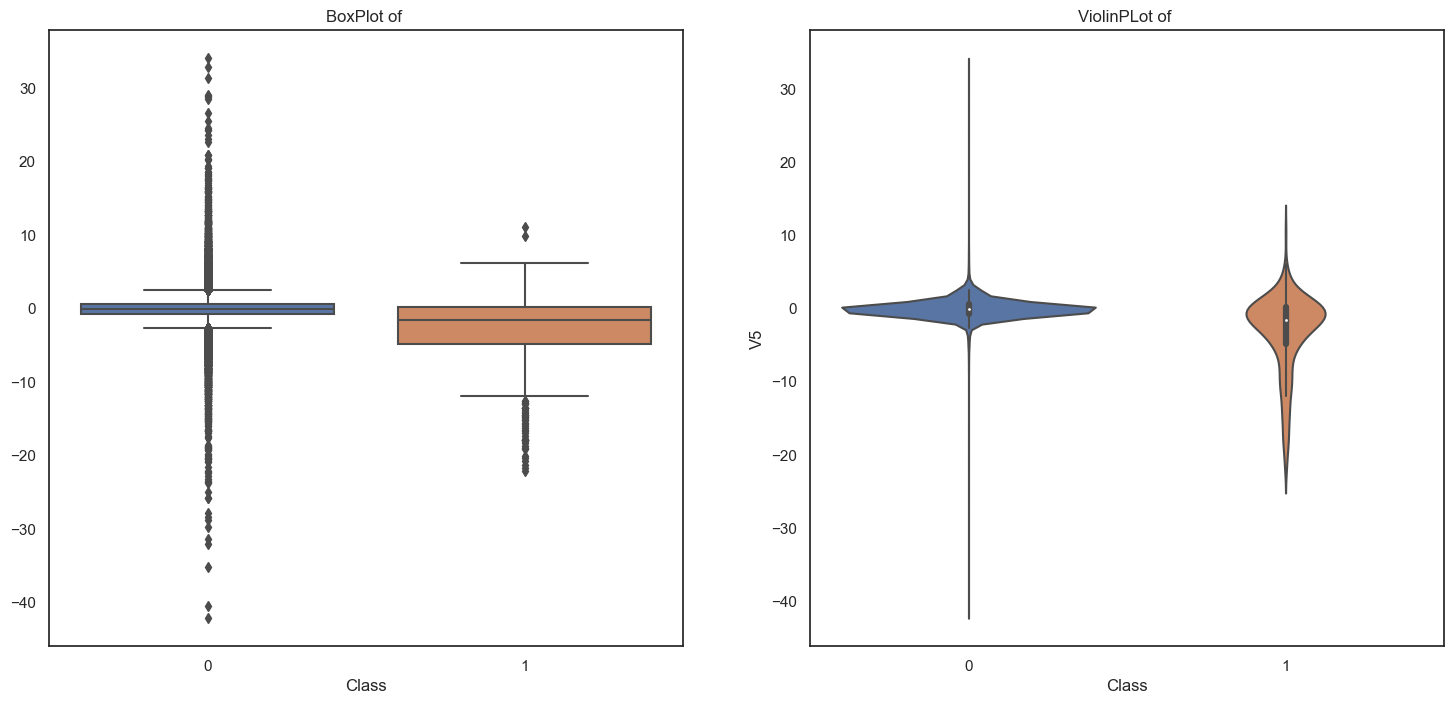
Plot between V3 vs Class



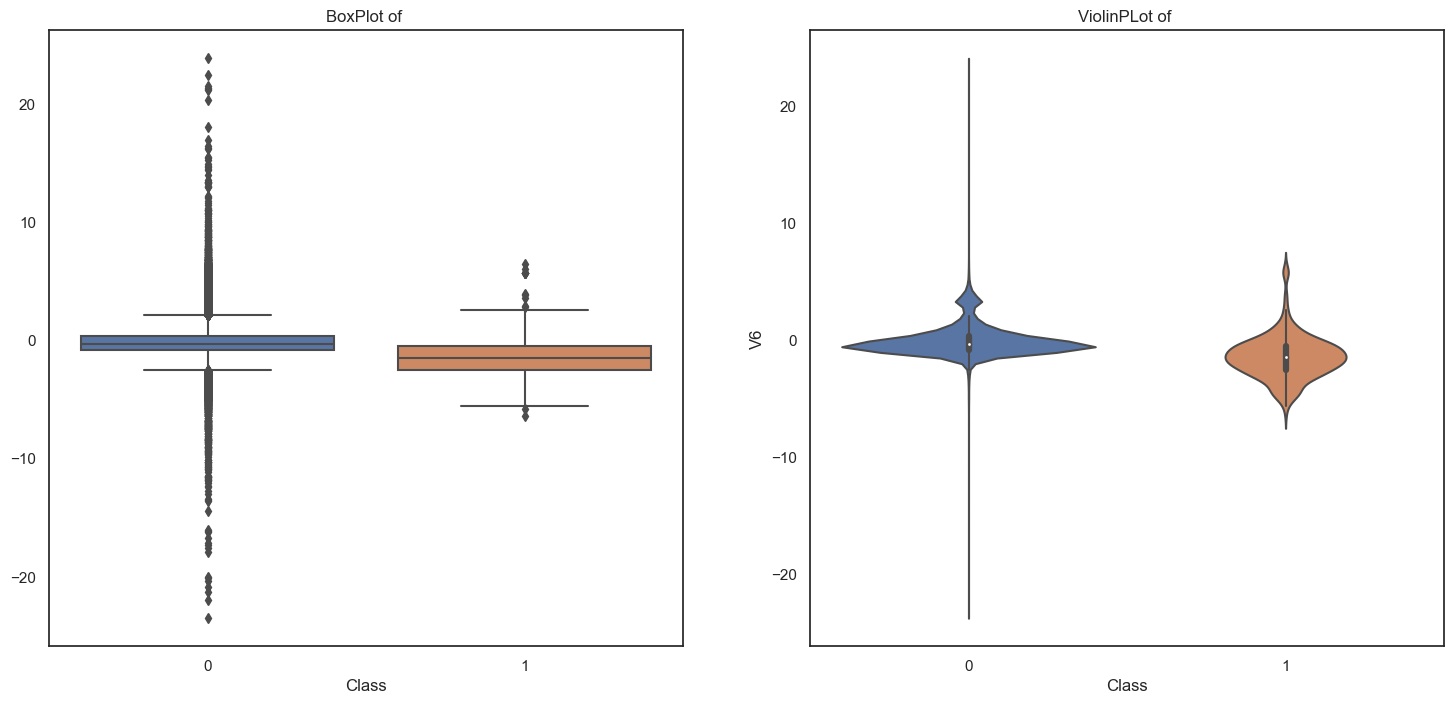
Plot between V4 vs Class



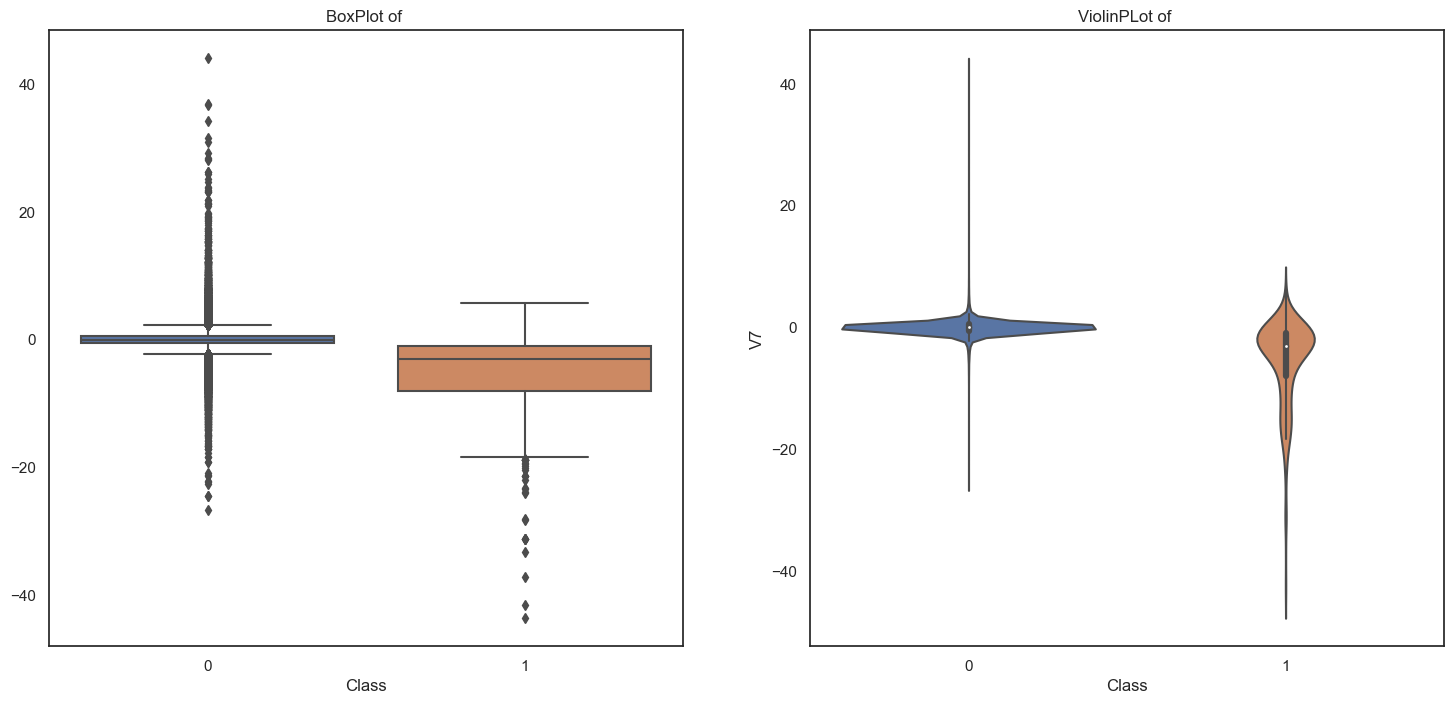
Plot between V5 vs Class



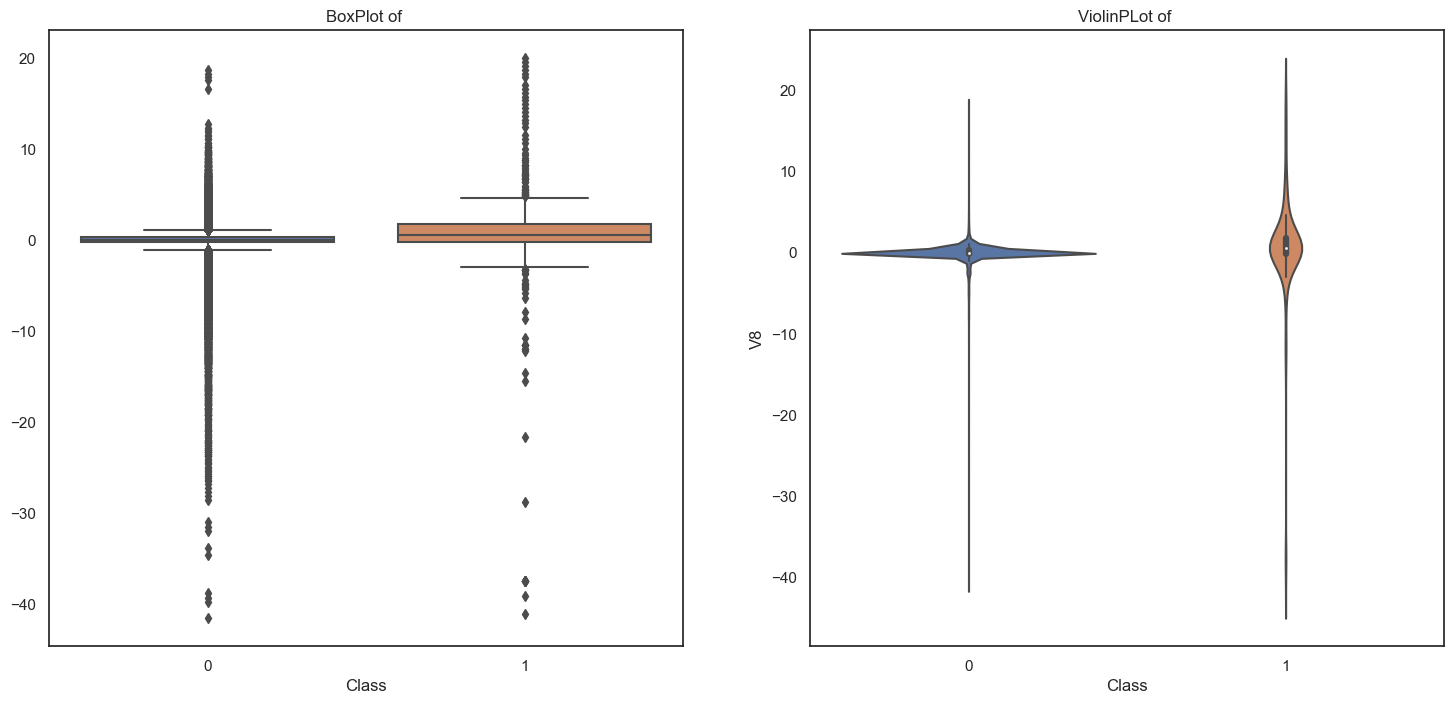
Plot between V6 vs Class



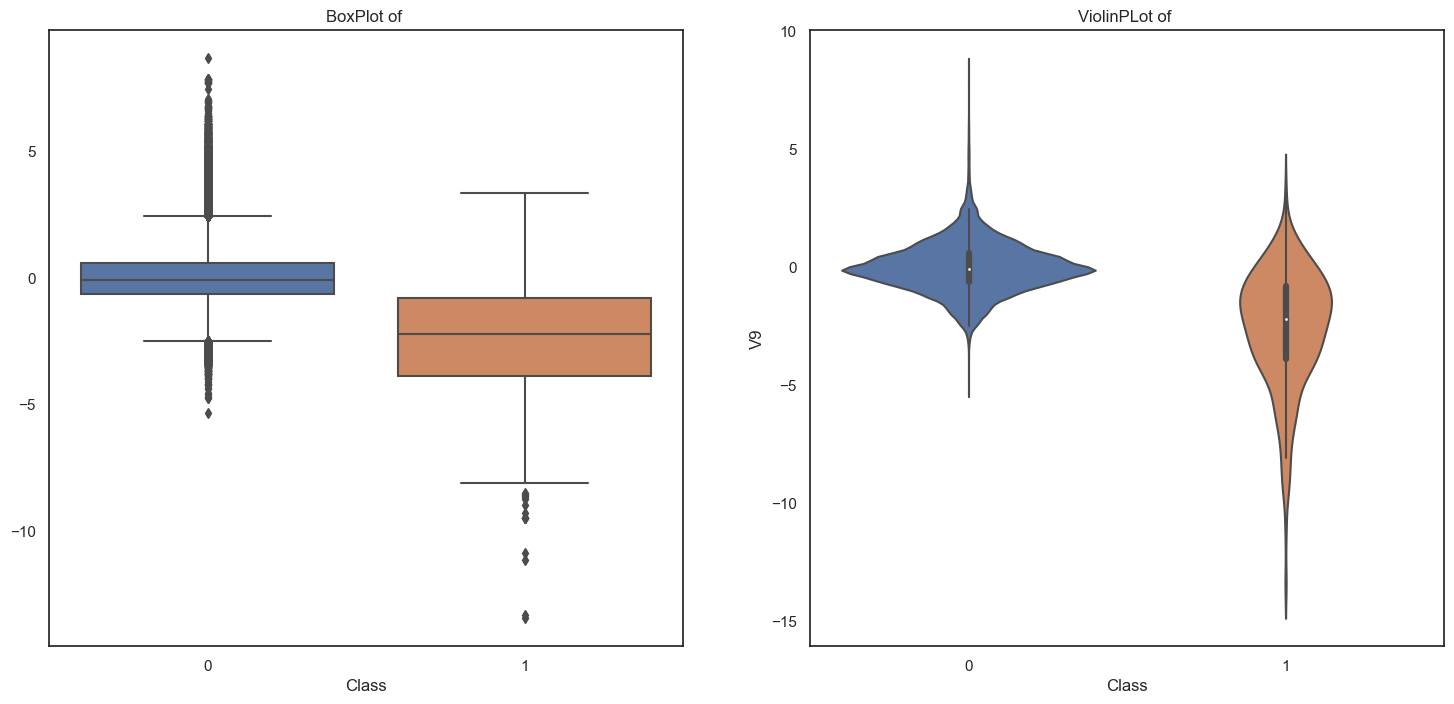
Plot between V7 vs Class



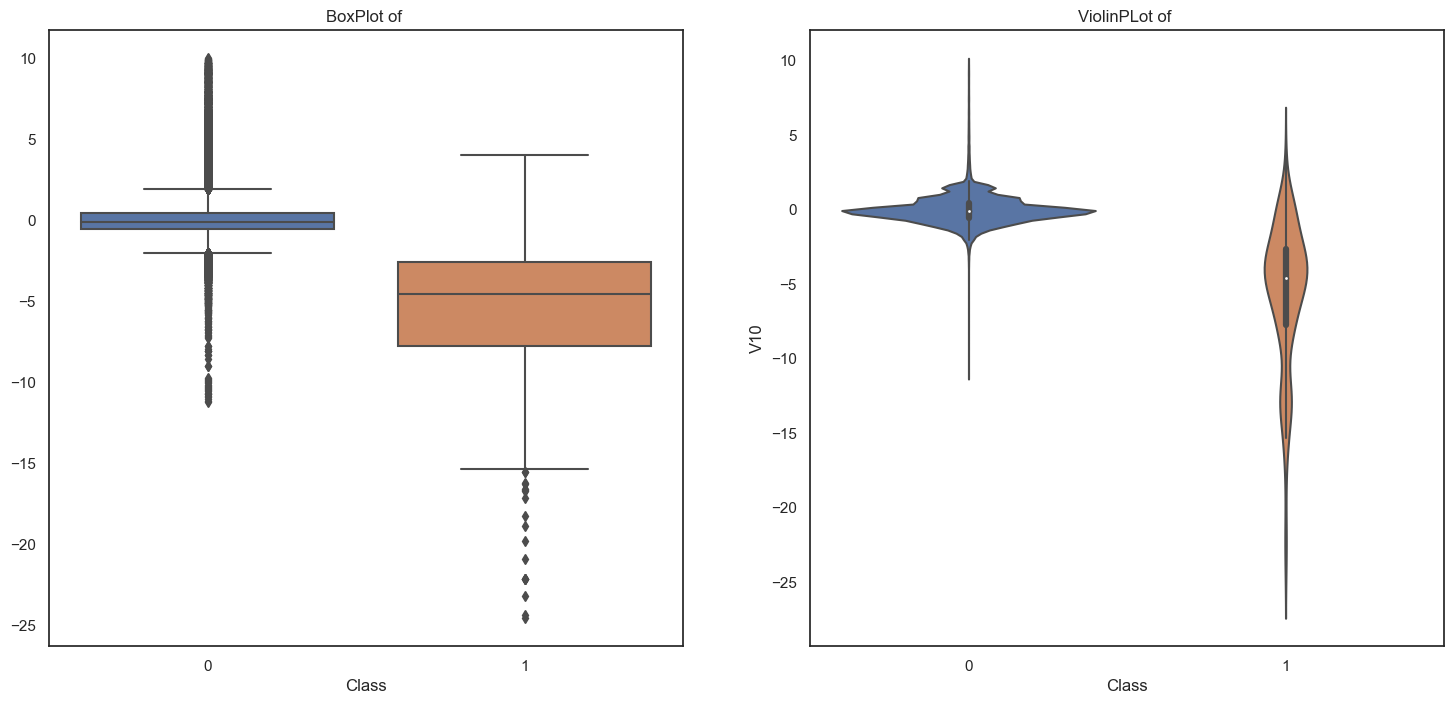
Plot between V8 vs Class



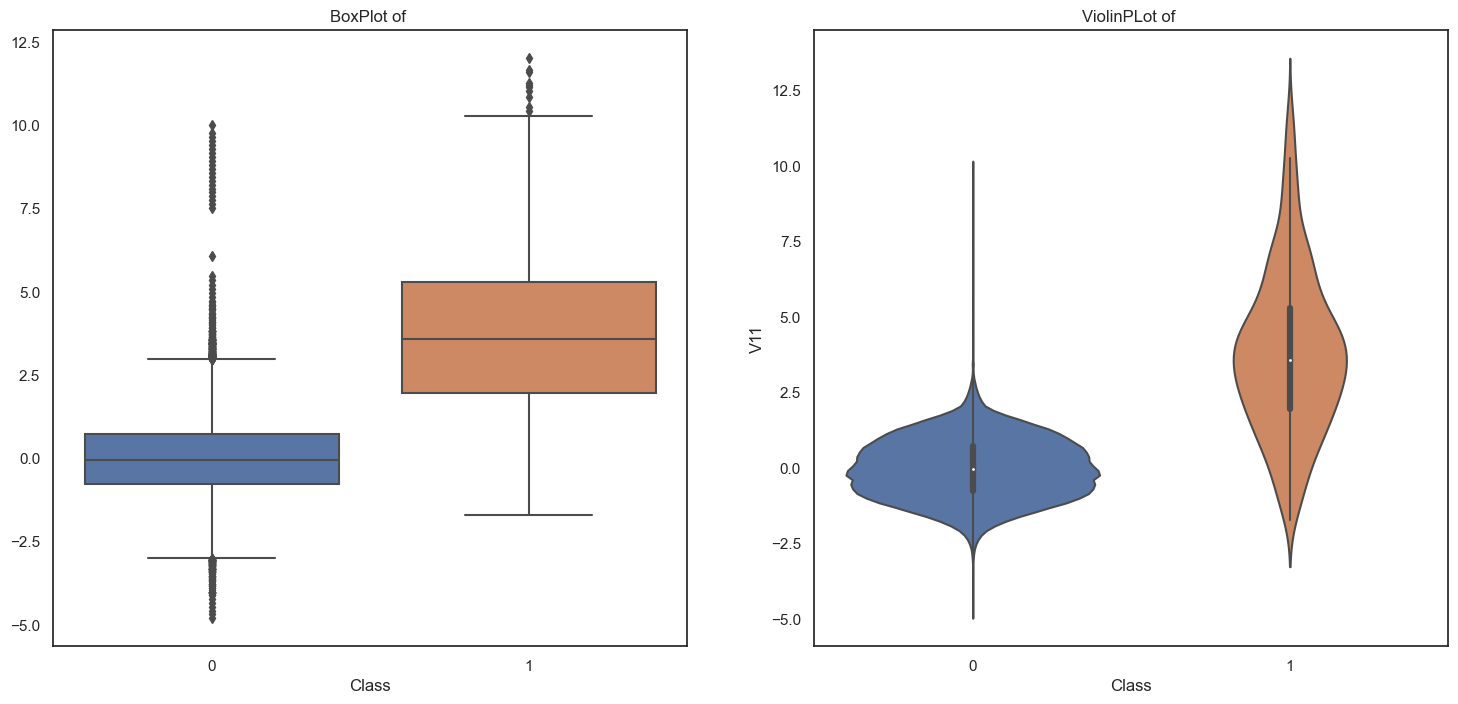
Plot between V9 vs Class



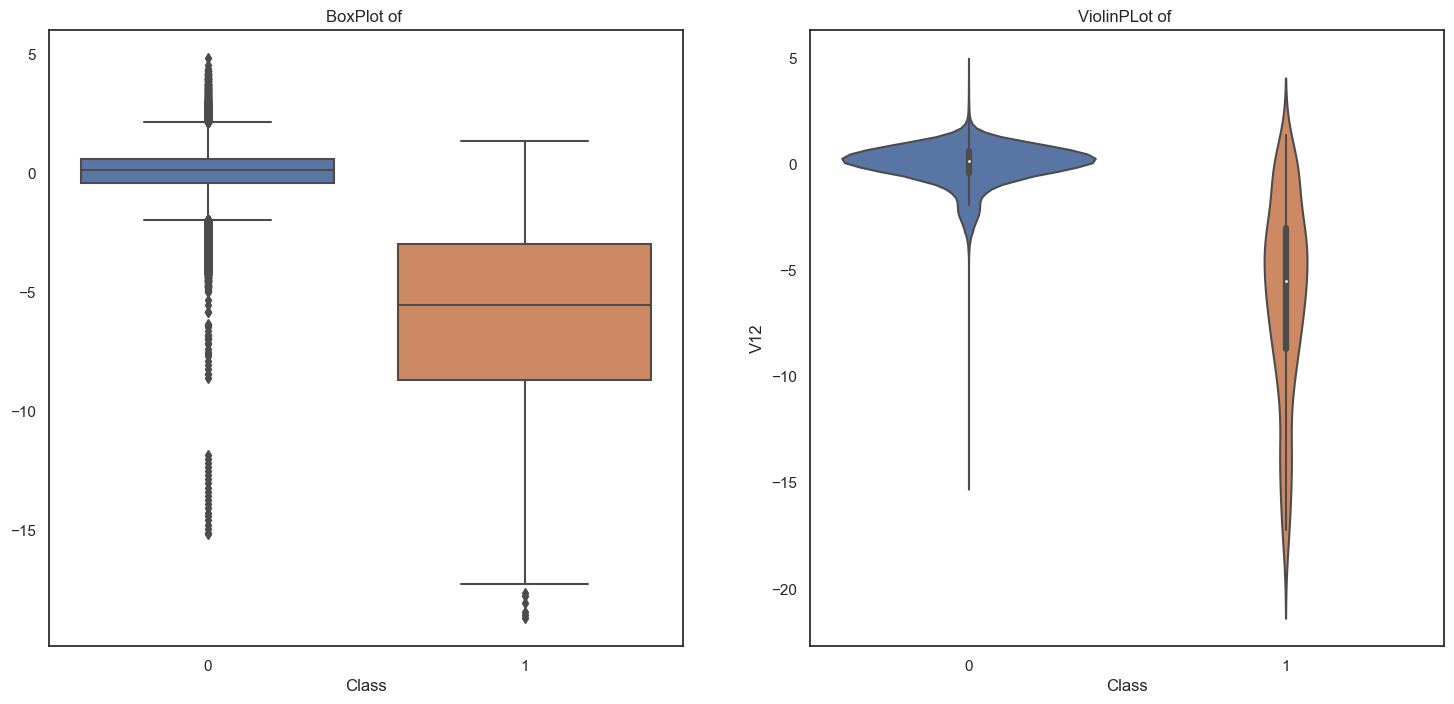
Plot between V10 vs Class



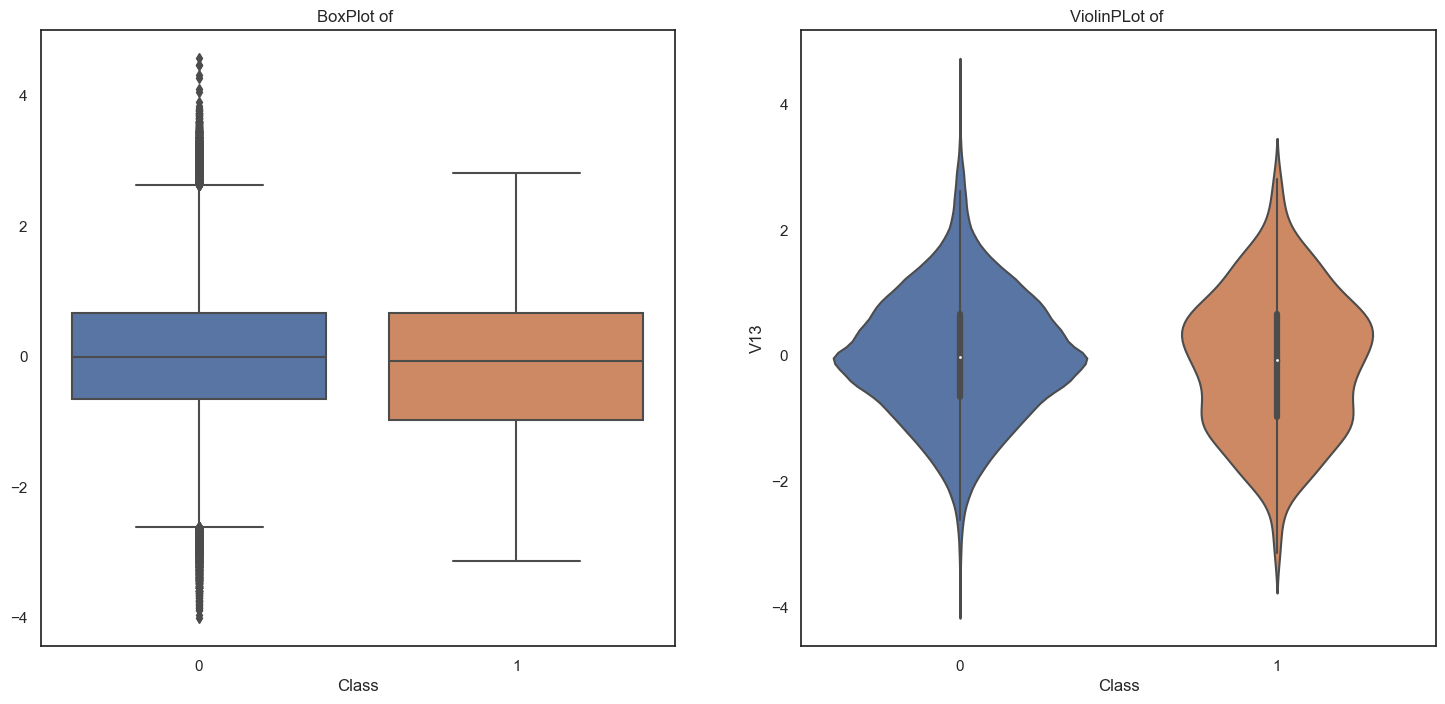
Plot between V11 vs Class



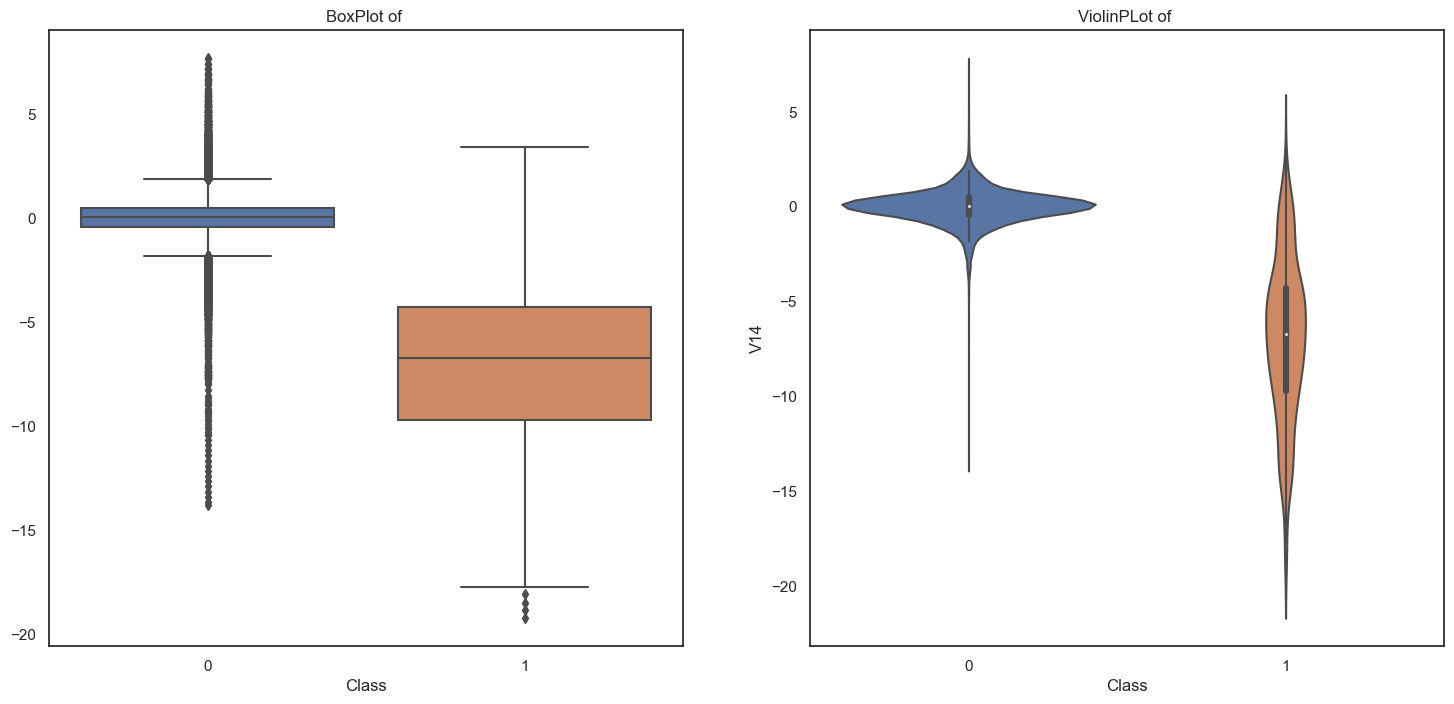
Plot between V12 vs Class



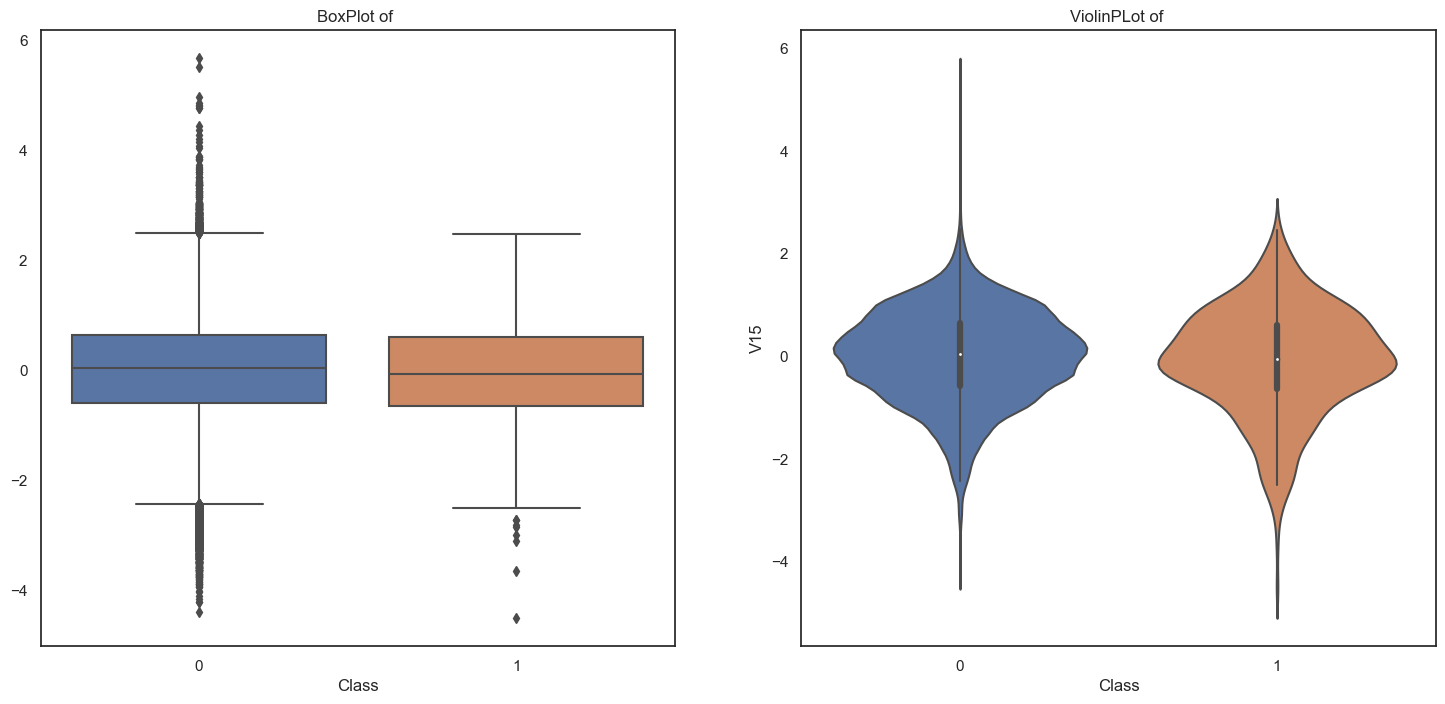
Plot between V13 vs Class



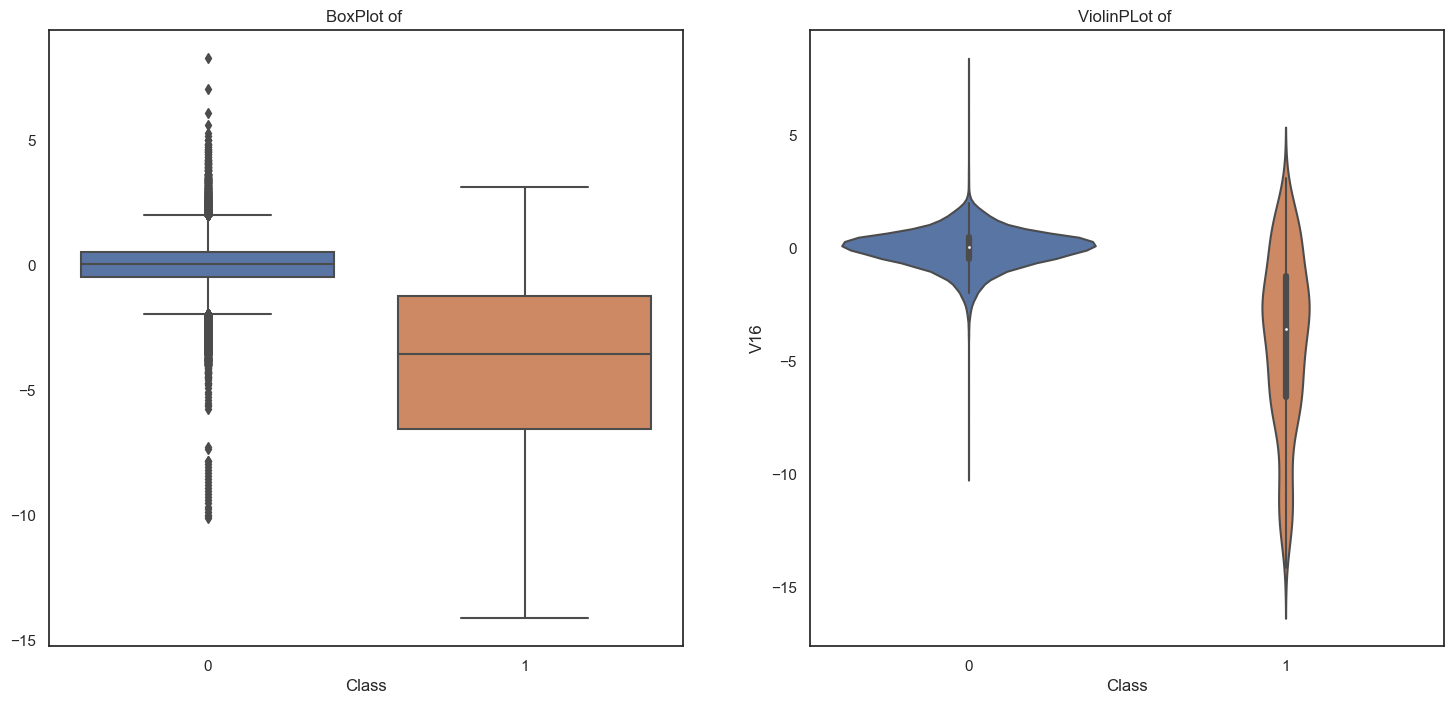
Plot between V14 vs Class



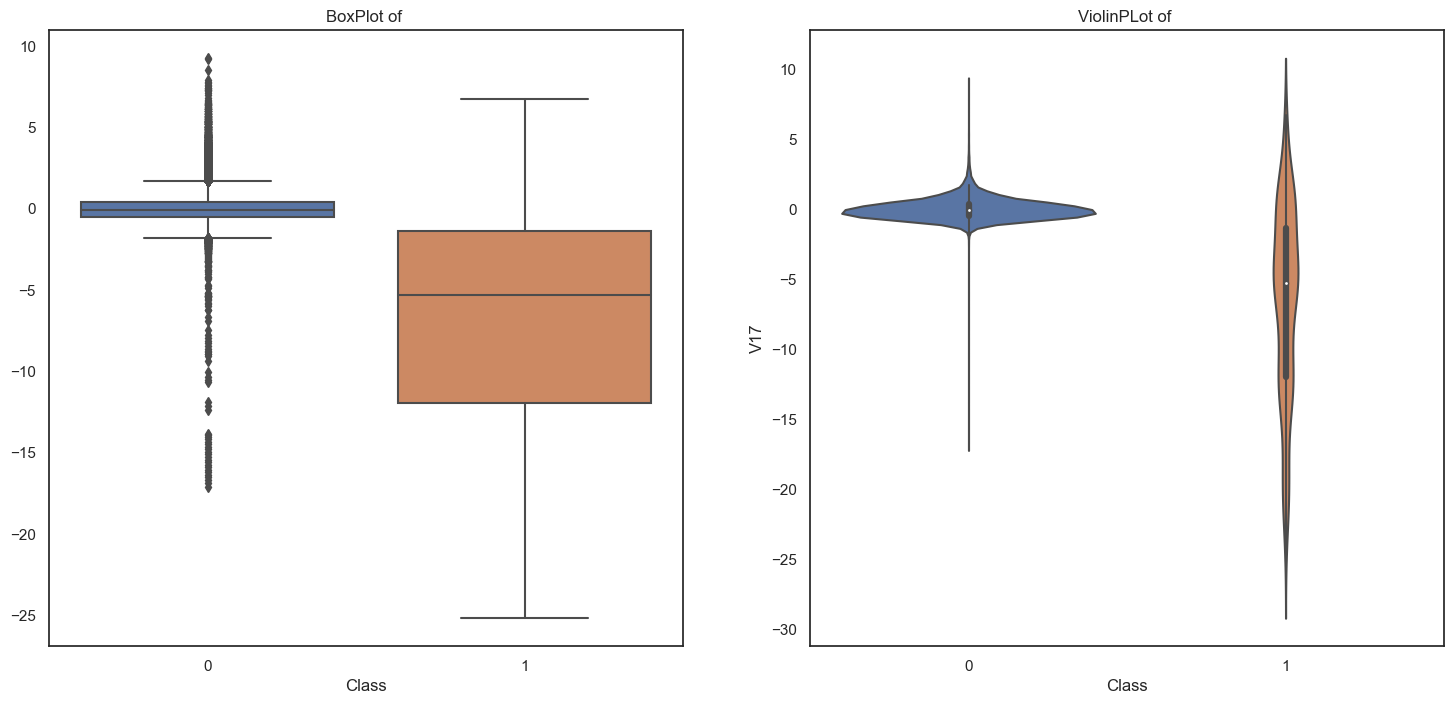
Plot between V15 vs Class



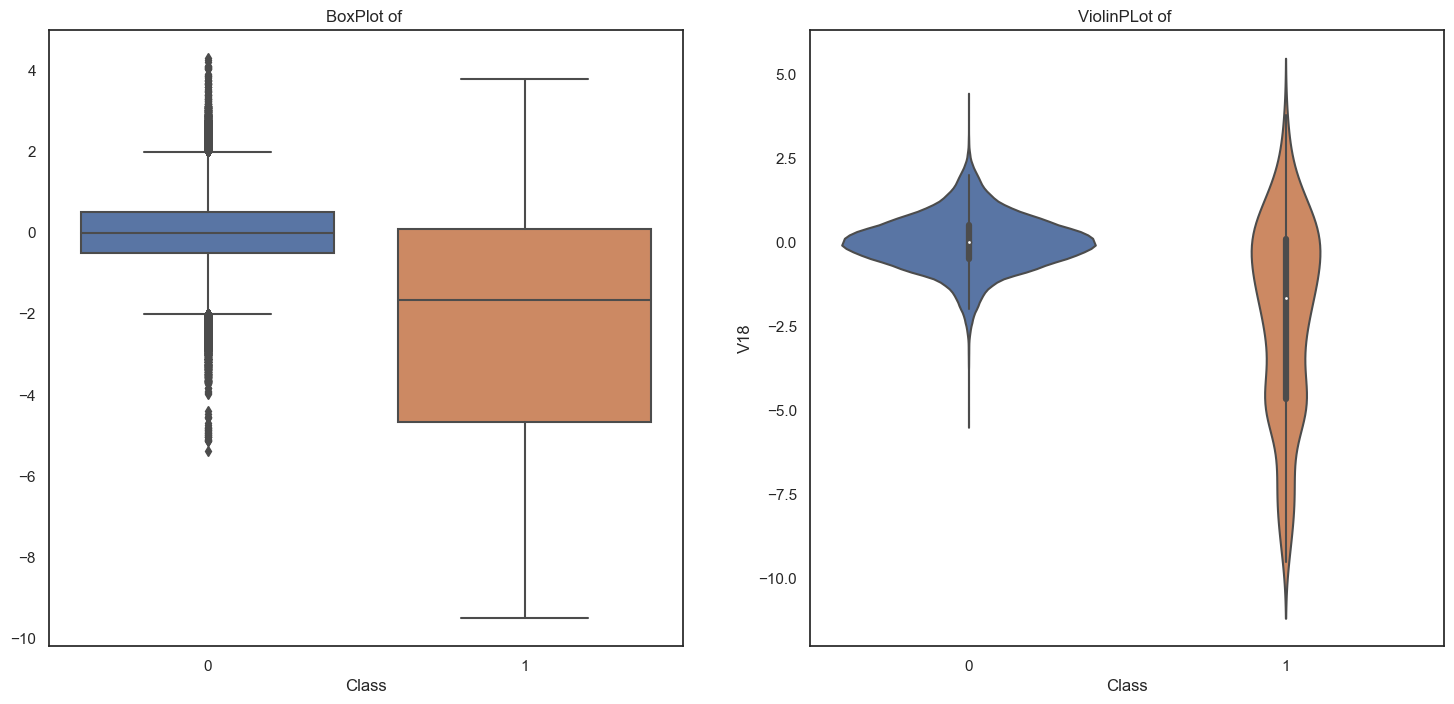
Plot between V16 vs Class



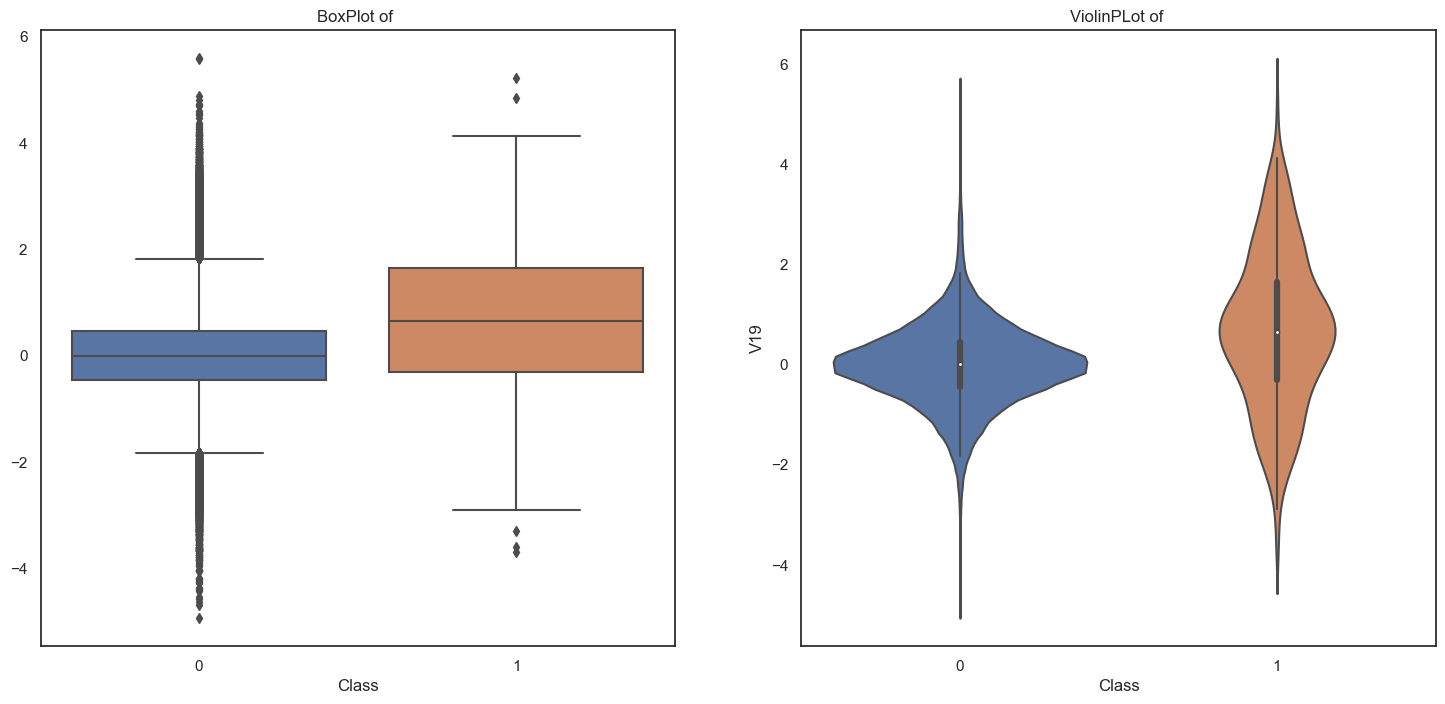
Plot between V17 vs Class



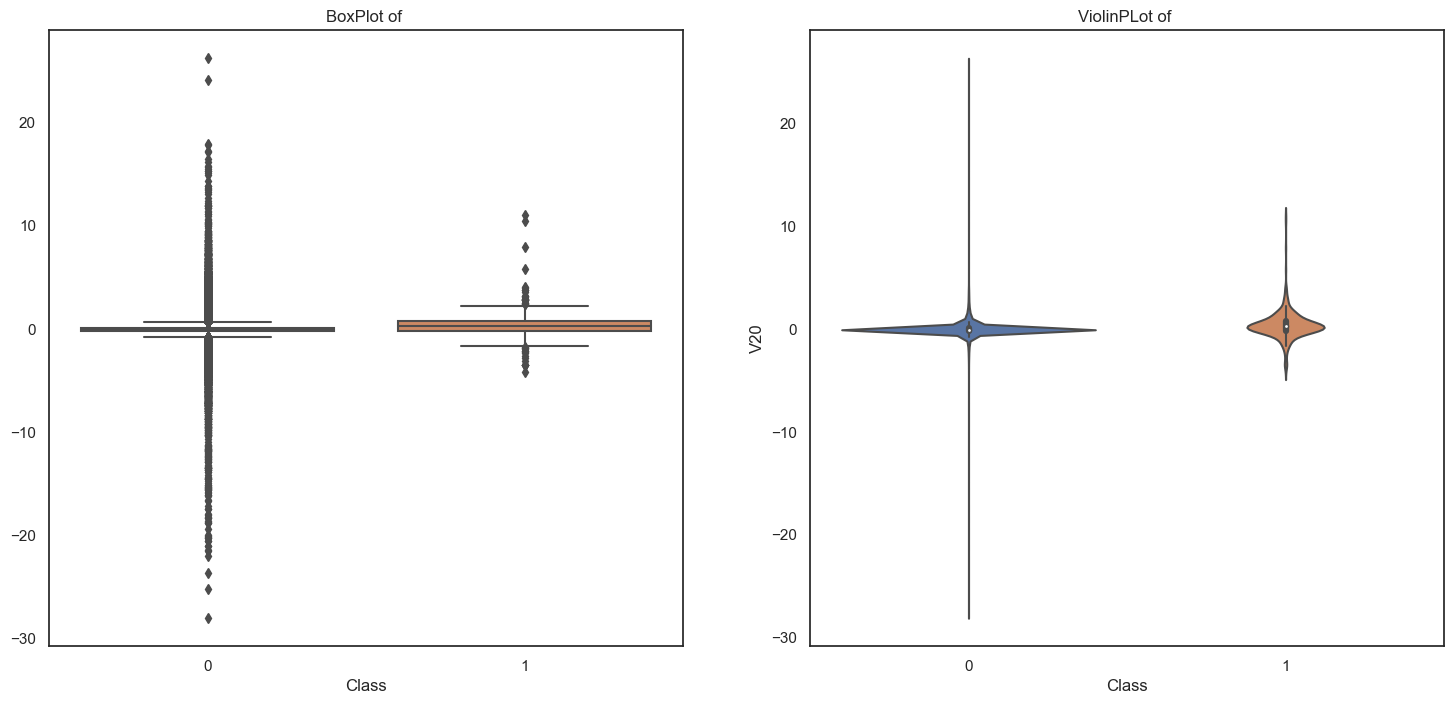
Plot between V18 vs Class



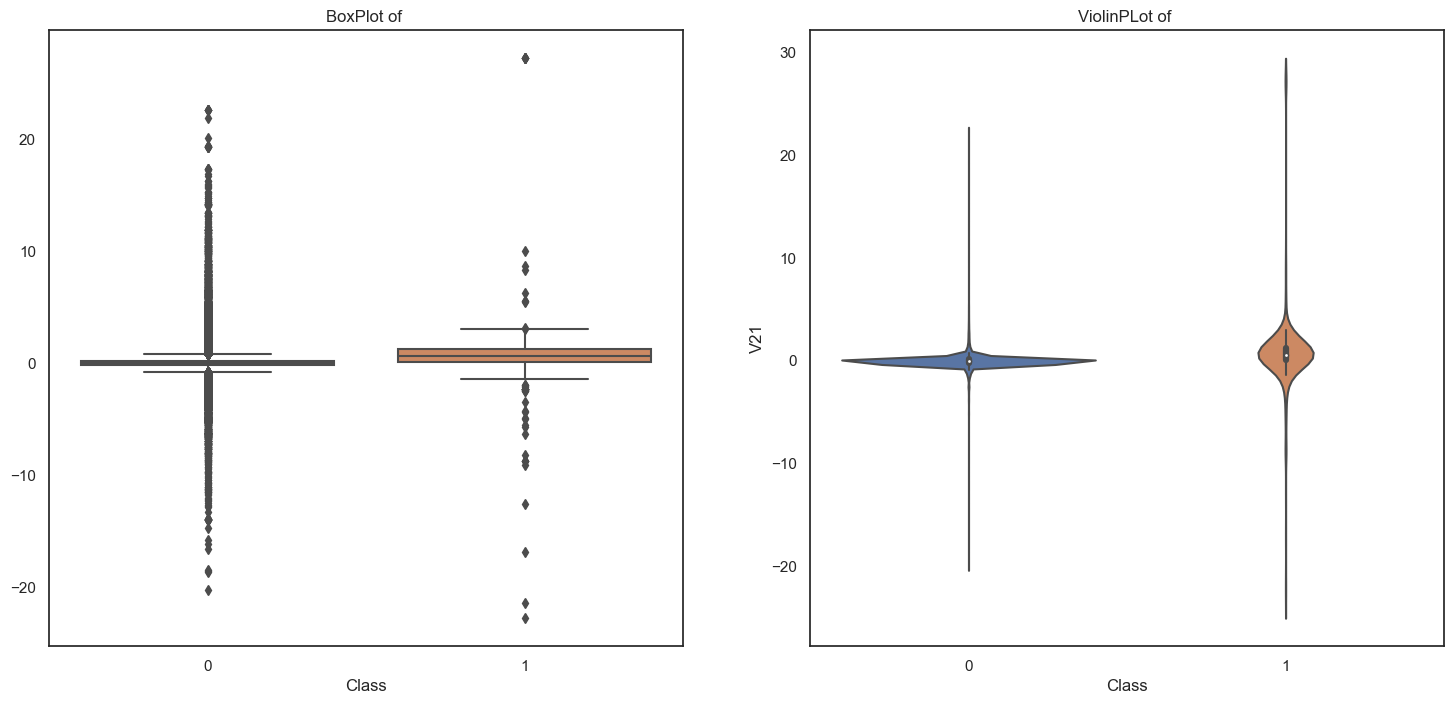
Plot between V19 vs Class



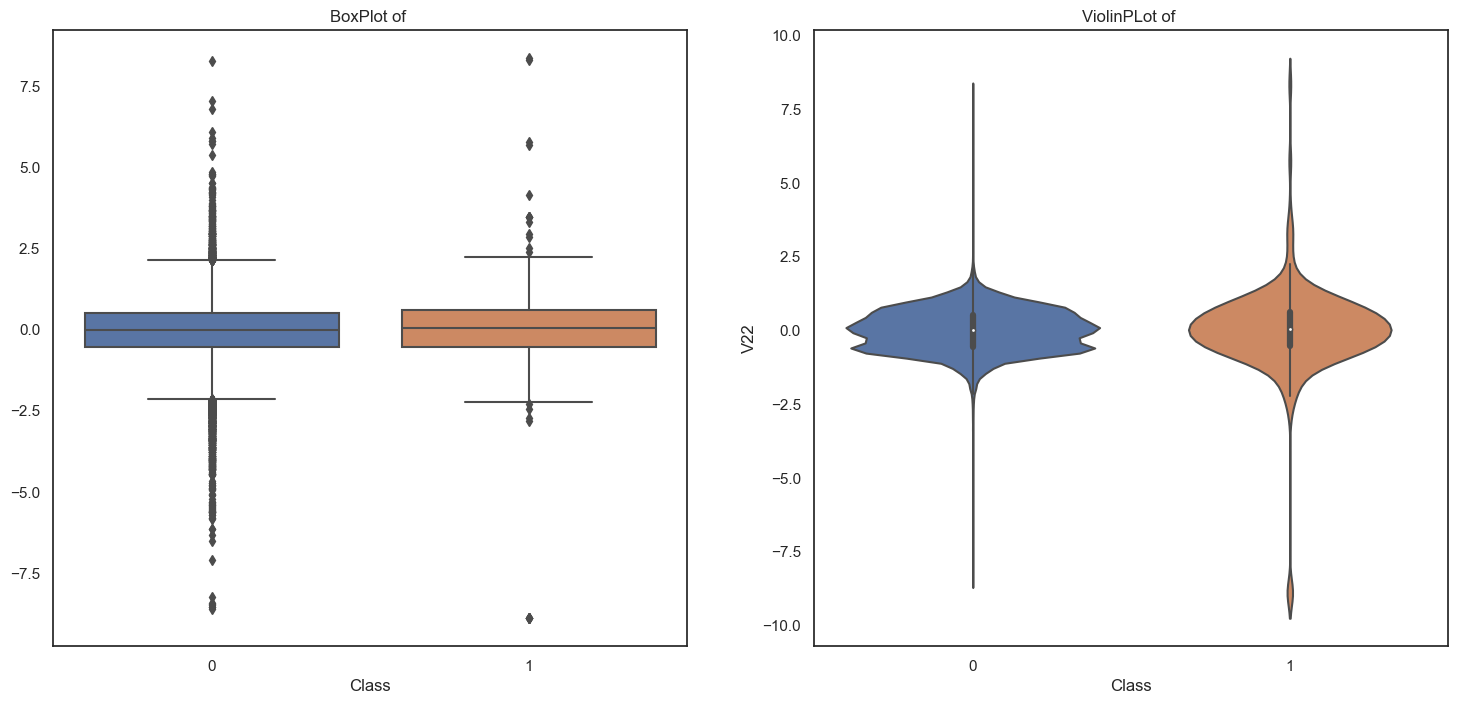
Plot between V20 vs Class



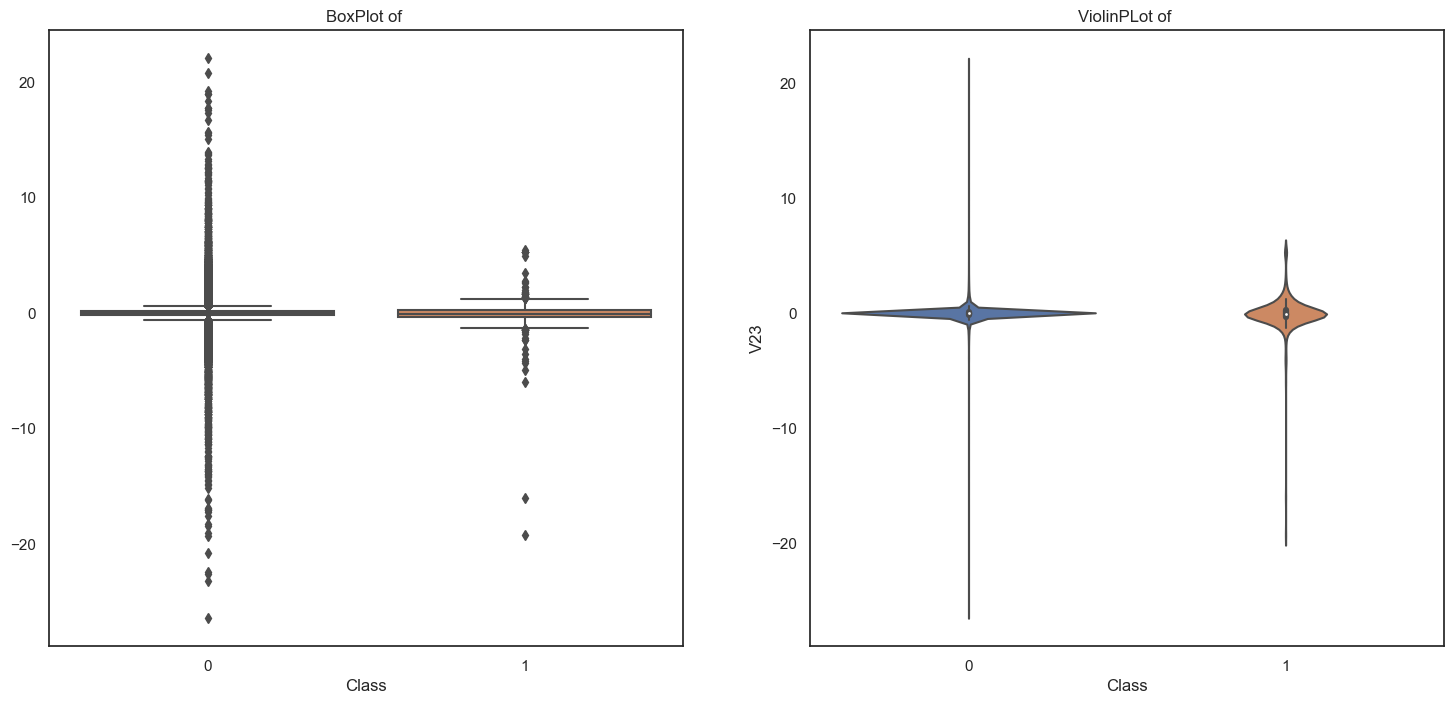
Plot between V21 vs Class



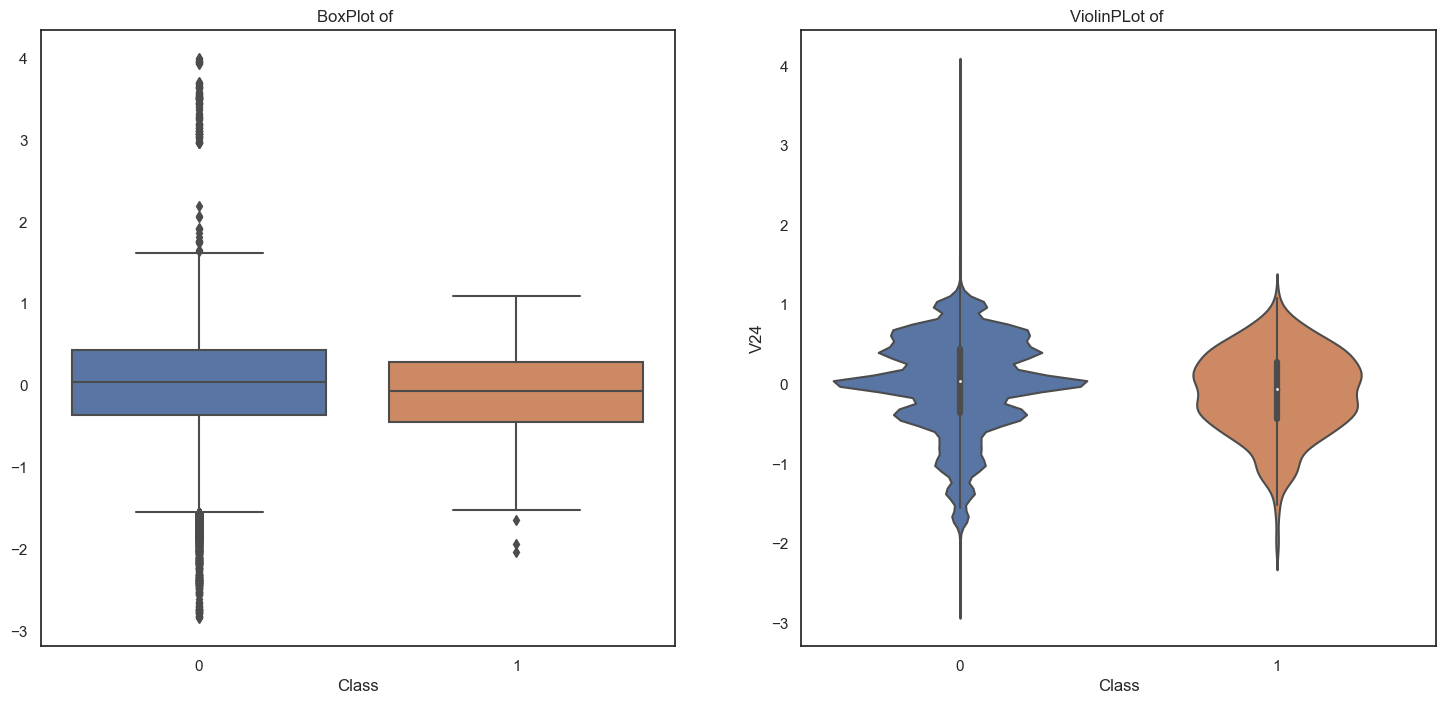
Plot between V22 vs Class



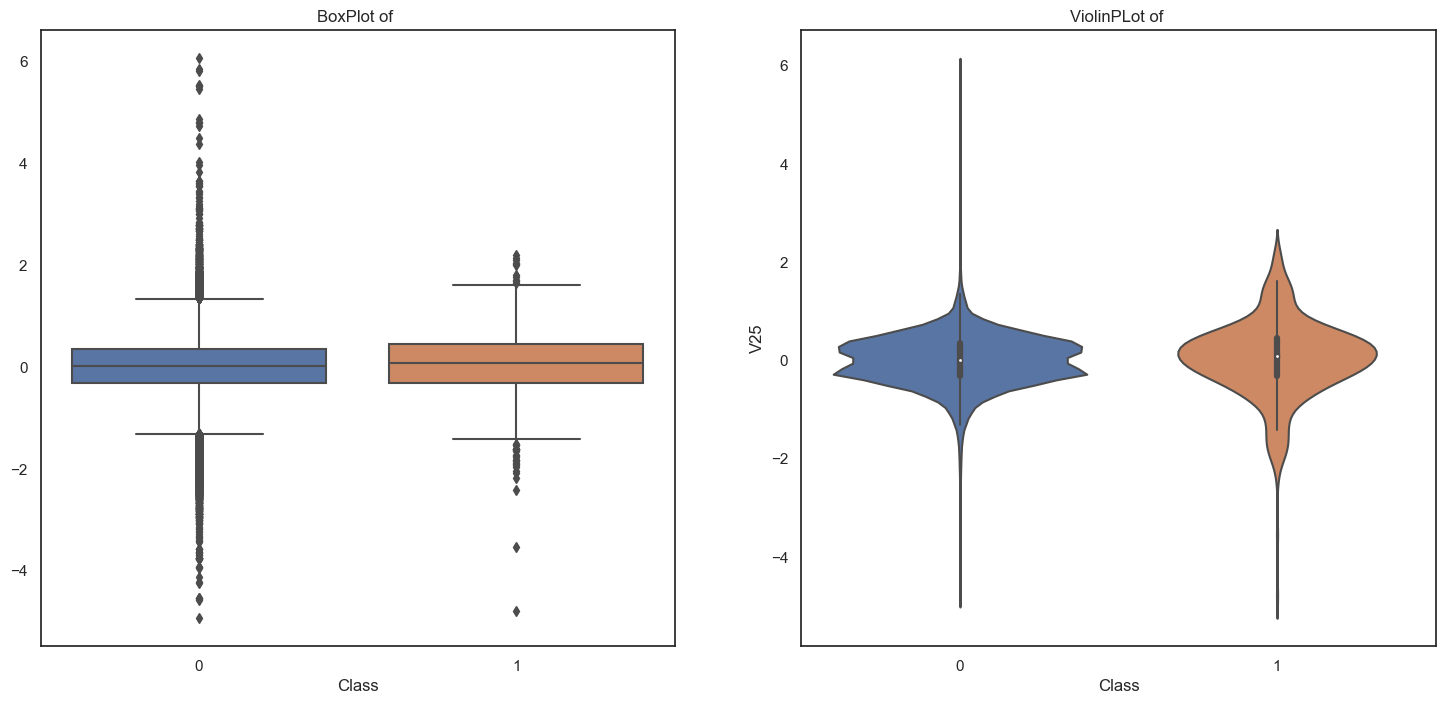
Plot between V23 vs Class



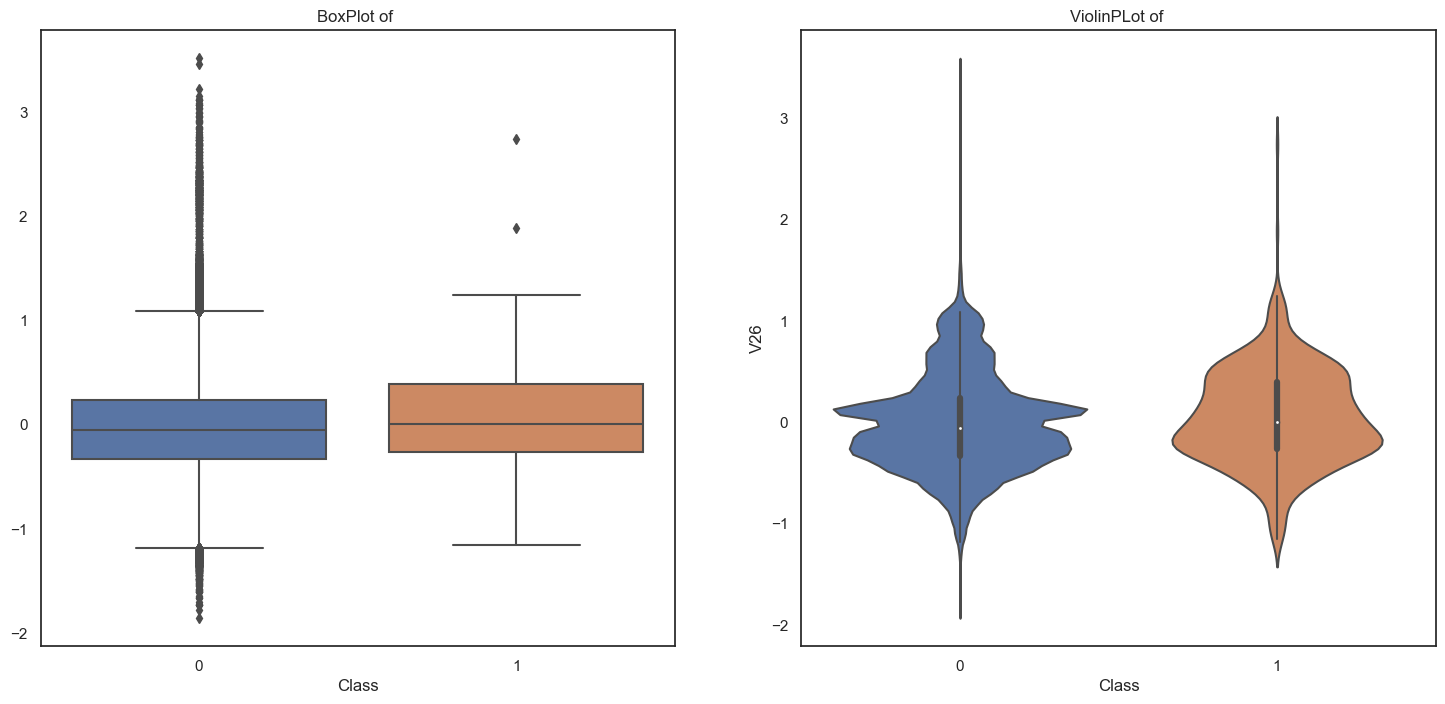
Plot between V24 vs Class



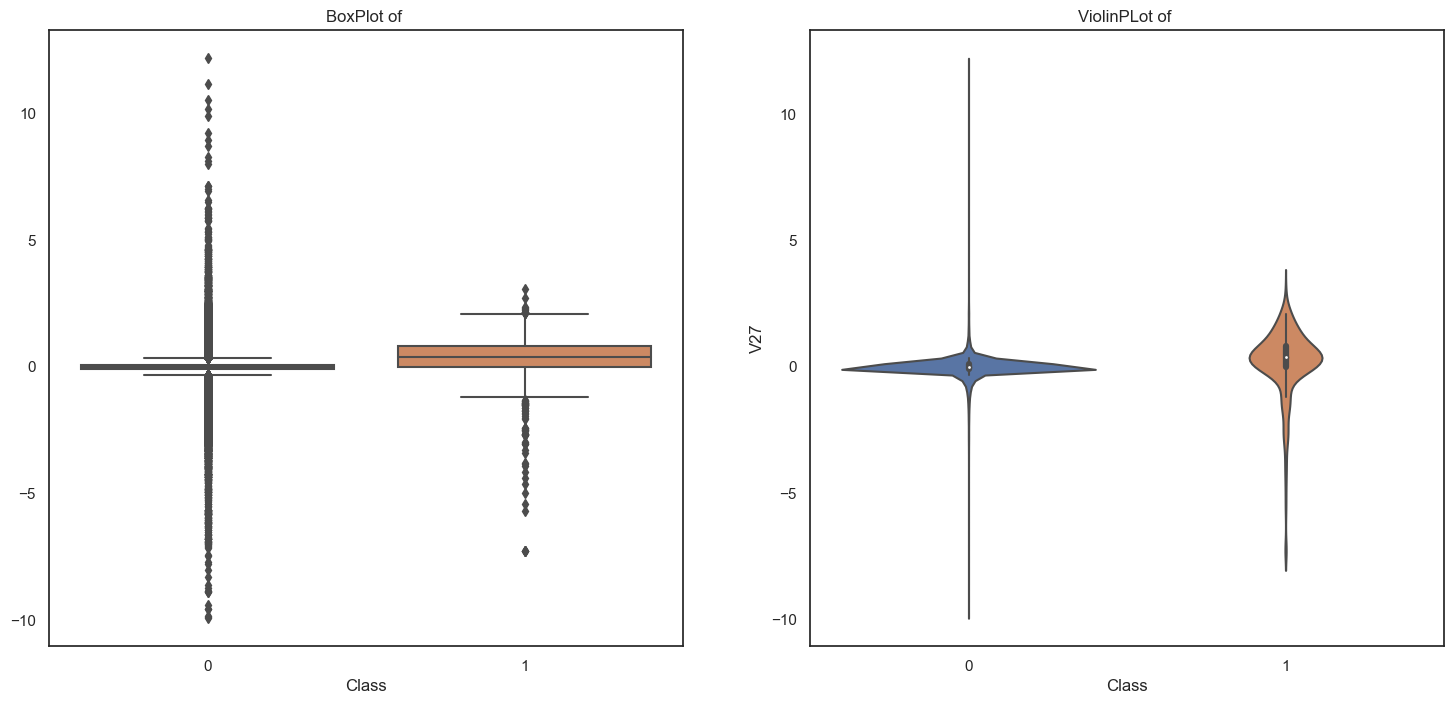
Plot between V25 vs Class



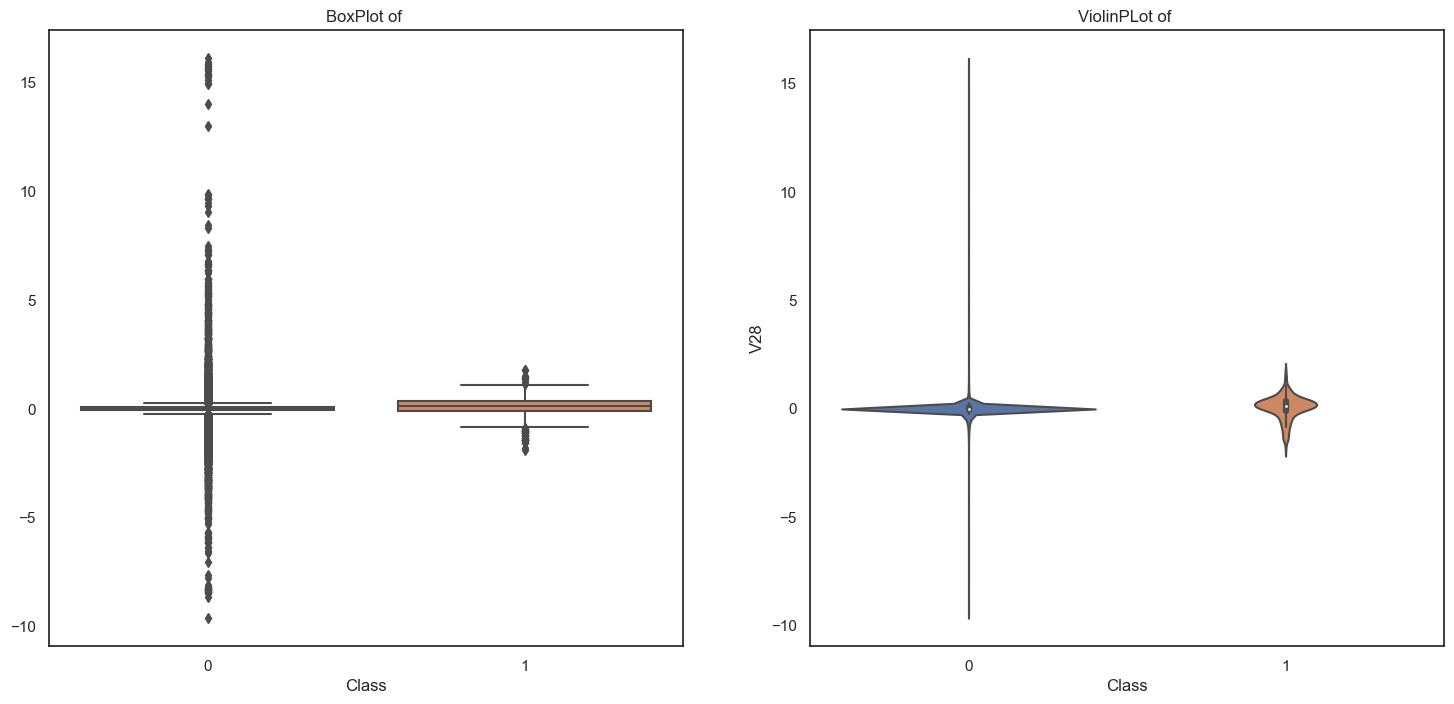
Plot between V26 vs Class



Plot between V27 vs Class



Plot between V28 vs Class

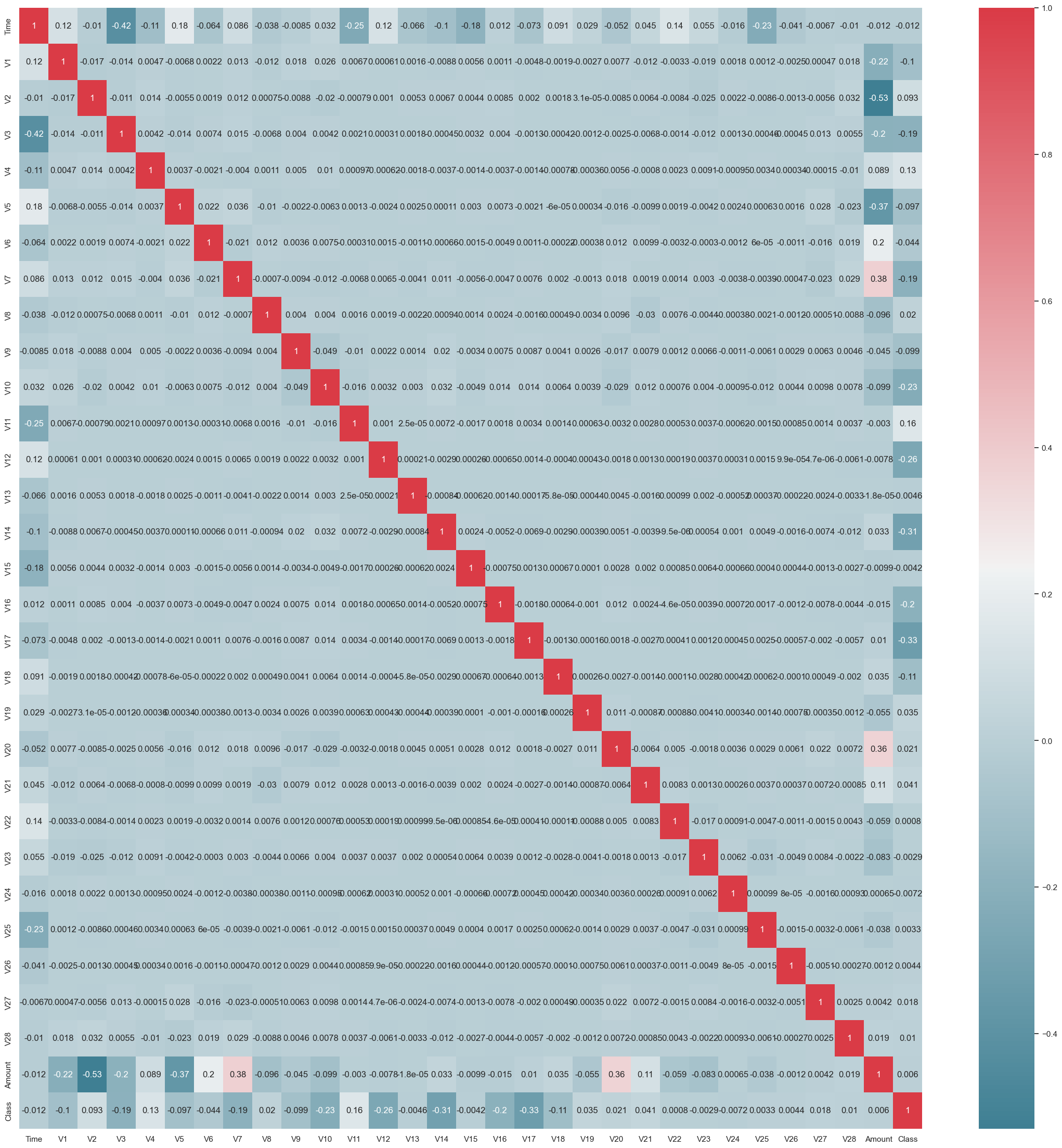


## Correlation analysis:

To understand more about how the variables are related to each other, correlation analysis has been done. This is achieved by visualising a heat map which represents all the variables on both x-axis and y-axis with their correlation coefficient values in each box of the heatmap. The below shown fig is the correlation heatmap obtained.

Figure

*Correlation plot of the data*



## 5.7 Finding Missing values:

As discussed in the above section, it is advantageous to clean the data before implementing any machine learning algorithms. In this project, the data has No missing values, if in case missing values are present it can be been cleaned by filling the missing values with mean and median values of respective variables. The below shown table represents the number of missing values present in each variable. Missing values can be checked with isnull().sum() function.

Time 0

V1 0

V2 0

V3 0

V4 0

V5 0

V6 0

V7 0

V8 0

V9 0

V10 0

V11 0

V12 0

V13 0

V14 0

V15 0

V16 0

V17 0

V18 0

V19 0

V20 0

V21 0

V22 0

V23 0

V24 0

V25 0

V26 0

V27 0

V28 0

Amount 0

Class 0

# Module 6: Implementation

In this section, the process of implementing the proposed methods is discussed. The platform and programming language used to build and execute the models are explained in this section. Certain factors considered for building machine learning and deep learning models are mentioned and the reasons to choose those parameters are discussed.

## 6.1 Train-Test Split:

To make the data ready to be fit on the models, it is divided into training and testing in a ratio of 70:30. This means that 70% of the data is considered as training set which is used to train the models and 30% of the data is considered as testing set which is used to test the model’s performance. This is achieved by importing a function from scikit-learn library called “train\_test\_split”. Generally, when solving machine learning problems, the data used is split in the ratio of 70:30 because it produces optimal results. This ratio of dividing the data can be varied according to the application. The independent variables are grouped together and are referred to as X. While the dependent variables grouped together are represented by Y. Next, this X and Y sets are divided into training and testing sets of data. In the below shown table 10 the shape of training and testing data after splitting the data is represented.

Table 10

*Shape of Training & Testing data*

|  |  |
| --- | --- |
| Shape of X\_train | (199221, 30) |
| Shape of X\_test | (85381, 30) |
| Shape of y\_train | (199221, 1) |
| Shape of y\_test | (85381, 1) |

## 

## 6.2 Machine learning algorithms Model Building :

Machine learning models used in this project are implemented by importing the scikit-learn library. The decision tree algorithm used is fitted on the training and testing sets of data. The random forest algorithm is fitted on the data by setting certain parameters which decide the performance of it. The estimator’s value is set as 100 and the max\_features is set as sqrt. Similarly, logistic regression and support vector machine are fitted on the data without setting any specific parameters. The k nearest neighbour algorithm is supplied by the number of neighbours values being 5 which means it is supposed to calculate the distance between its five nearest neighbours.

### 

# Module7: Experimental Results

In this section of the report, the results obtained by implementing the proposed methodology are discussed. These results include various types of metrics applied to evaluate the performance of machine learning algorithms. This section also includes comparison of accuracy scores of all the models applied, which explains the reason why a specific model has been chosen as the best one.

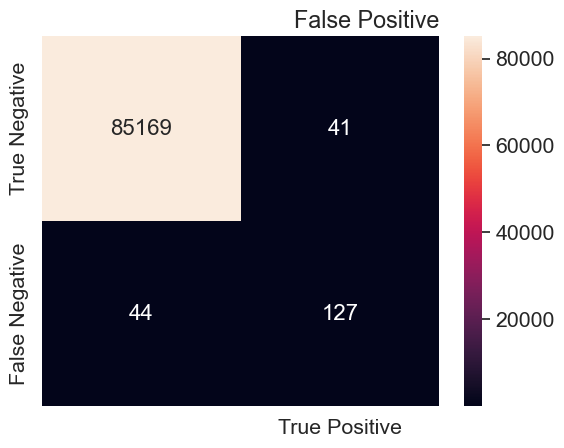
## 7.1 Confusion Matrix

1. **Decision Tree Algorithm:**

The figure shown below is a 2\*2 confusion matrix of Decision Tree algorithm.

Figure

*Confusion matrix of Decision tree classifier*

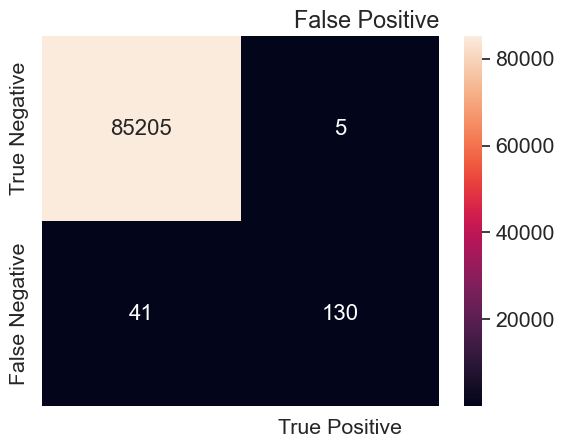


1. **Random Forest Algorithm:**

The figure shown below is a 2\*2 confusion matrix of Random Forest Algorithm.

Figure

*Confusion matrix of Random Forest classifier*

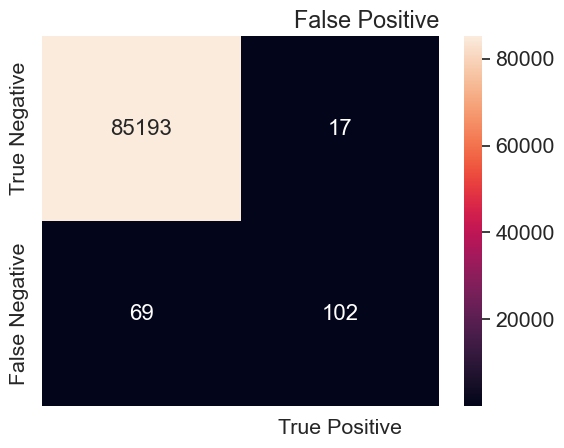


1. **Logistic Regression:**

The below shown figure is a 2\*2 confusion matrix of Logistic Regression algorithm.

Figure

*Confusion matrix of Logistic Regression*

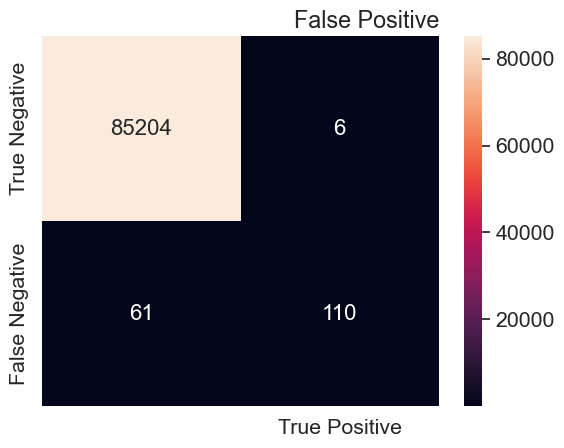


1. **Support Vector Classifier:**

The below shown figure is a 2\*2 confusion matrix of Support Vector Classifier algorithm.

Figure

*Confusion matrix of Support Vector Classifier*

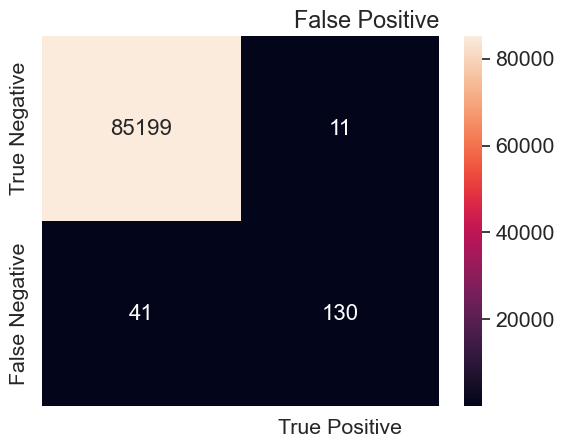


1. **K Nearest Neighbour Classifier:**

The below shown figure is a 2\*2 confusion matrix of K Nearest Neighbour Classifier algorithm.

Figure

*Confusion matrix of KNN*



## 7.2 Classification Report:

1. **Decision Tree Classifier:**

The below shown figure is the classification report obtained for Decision Tree algorithm.

*Classification report of Decision Tree*

Decision Tree Classifications Report:

precision recall f1-score support

0 1.00 1.00 1.00 85210

1 0.76 0.74 0.75 171

accuracy 1.00 85381

macro avg 0.88 0.87 0.87 85381

weighted avg 1.00 1.00 1.00 85381

1. **Random Forest Classifier:**

The below shown figure is the classification report obtained for Random Forest Classifier algorithm.

*Classification report of Random Forest*

Random Forest classification Report:

precision recall f1-score support

0 1.00 1.00 1.00 85210

1 0.96 0.76 0.85 171

accuracy 1.00 85381

macro avg 0.98 0.88 0.92 85381

weighted avg 1.00 1.00 1.00 85381

1. **Logistic Regression:**

The below shown figure is the classification report obtained for Logistic Regression algorithm.

*Classification report of Logistic Regression*

Logistic Regression clarification Report:

precision recall f1-score support

0 1.00 1.00 1.00 85210

1 0.86 0.60 0.70 171

accuracy 1.00 85381

macro avg 0.93 0.80 0.85 85381

weighted avg 1.00 1.00 1.00 85381

1. **Support Vector Classifier algorithm:**

The below shown figure is the classification report obtained for Support Vector Classifier algorithm.

*Classification report of SVM*

SVC Clasification Report:

precision recall f1-score support

0 1.00 1.00 1.00 85210

1 0.95 0.64 0.77 171

accuracy 1.00 85381

macro avg 0.97 0.82 0.88 85381

weighted avg 1.00 1.00 1.00 85381

1. **K Nearest Neighbour algorithm:**

The below shown figure is the classification report obtained for K Nearest Neighbour algorithm.

*Classification report of KNN*

KNN classifications Report:

precision recall f1-score support

0 1.00 1.00 1.00 85210

1 0.92 0.76 0.83 171

accuracy 1.00 85381

macro avg 0.96 0.88 0.92 85381

weighted avg 1.00 1.00 1.00 85381

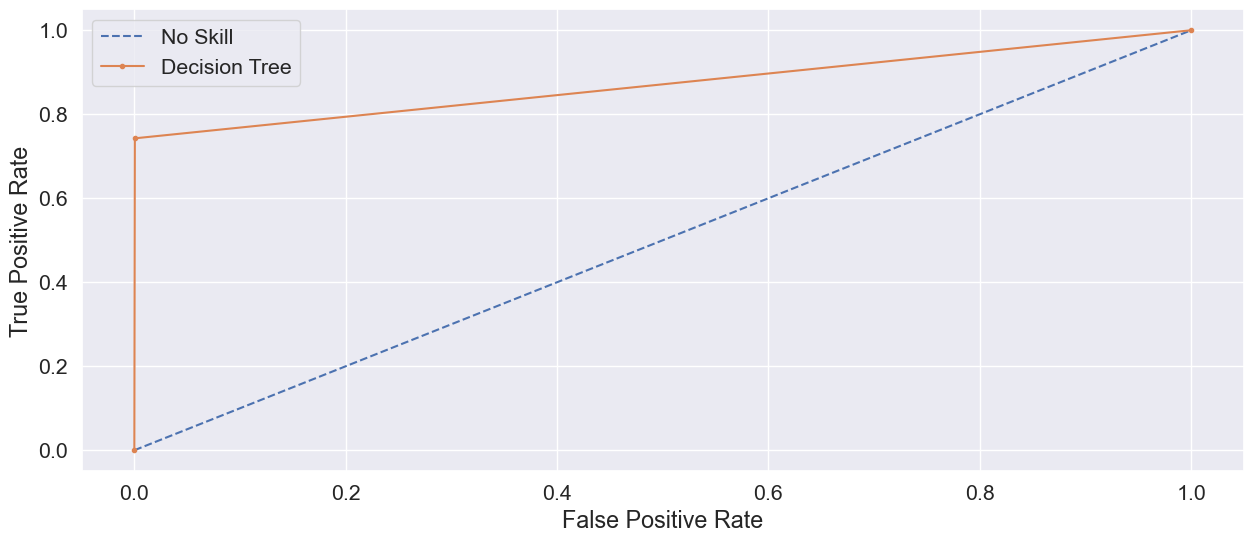
## 7.3 AUC-ROC Score & Curve

1. **Decision Tree Classifier:**

The below shown figure represents the AUC score and ROC curve of the Decision Tree Classifier algorithm.

Figure

*ROC Curve of Decision Tree*

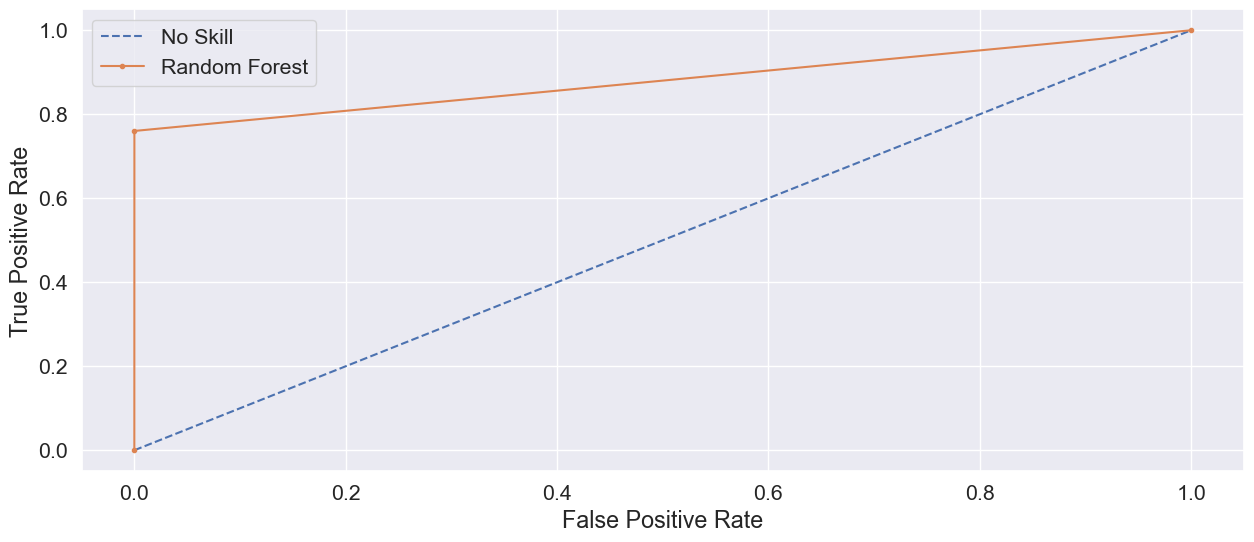


1. **Random Forest Classifier:**

The below shown figure represents the AUC score and ROC curve of the Random Forest Classifier algorithm.

Figure

*ROC Curve of Random Forest classifier*

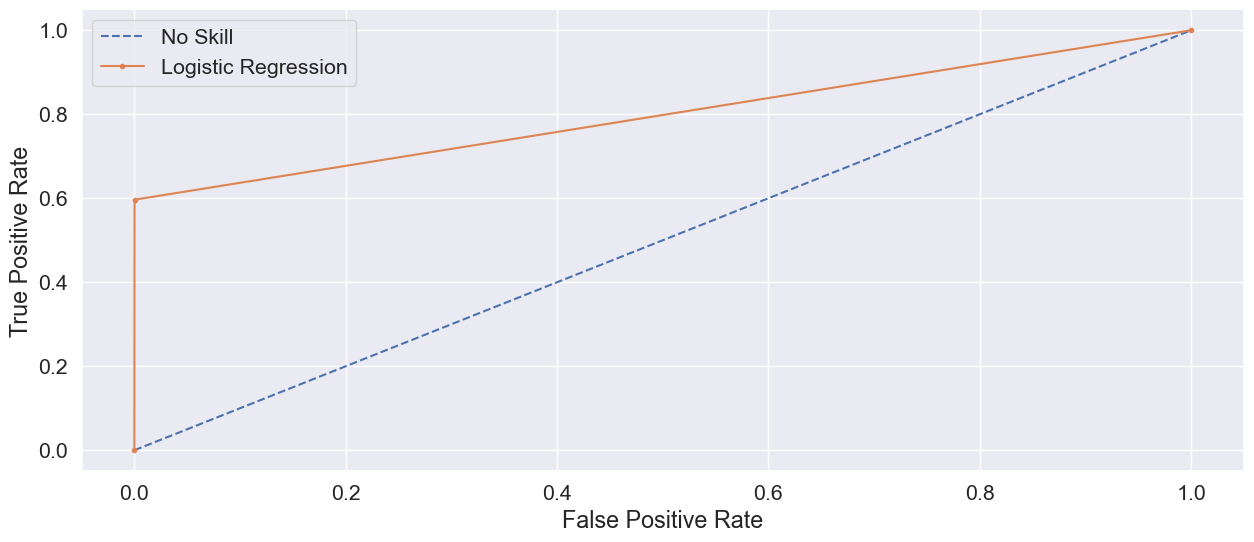


1. **Logistic Regression:**

The below shown figure represents the AUC score and ROC curve of the Logistic Regression algorithm.

Figure

*ROC Curve of Logistic Regression*

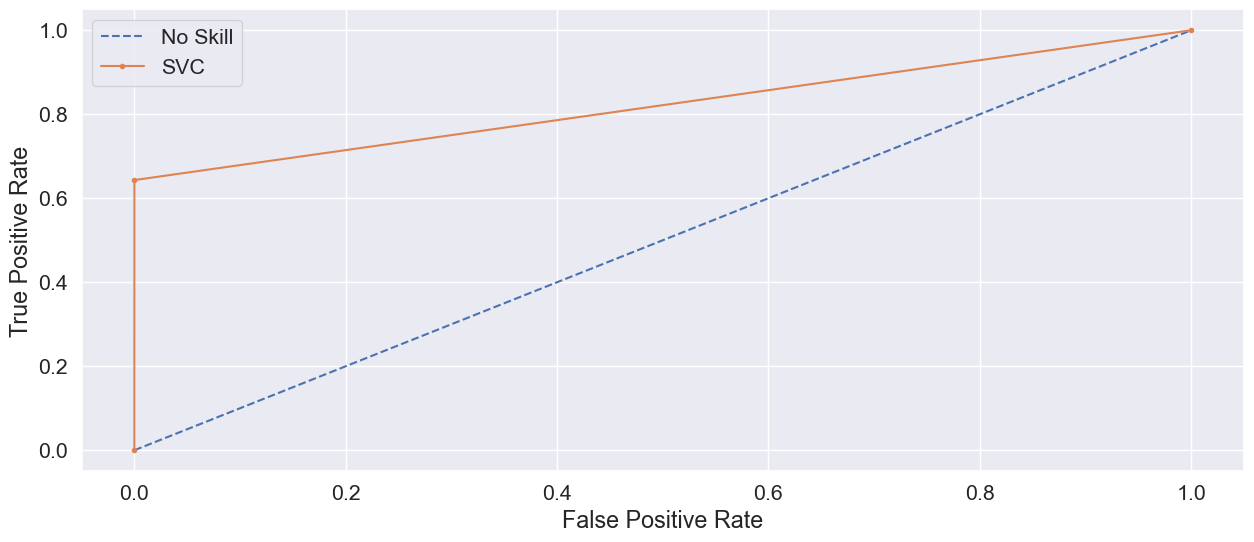


1. **Support Vector Machine:**

The below shown figure represents the AUC score and ROC curve of the Support Vector Machine algorithm.

Figure

*ROC Curve of SVM*

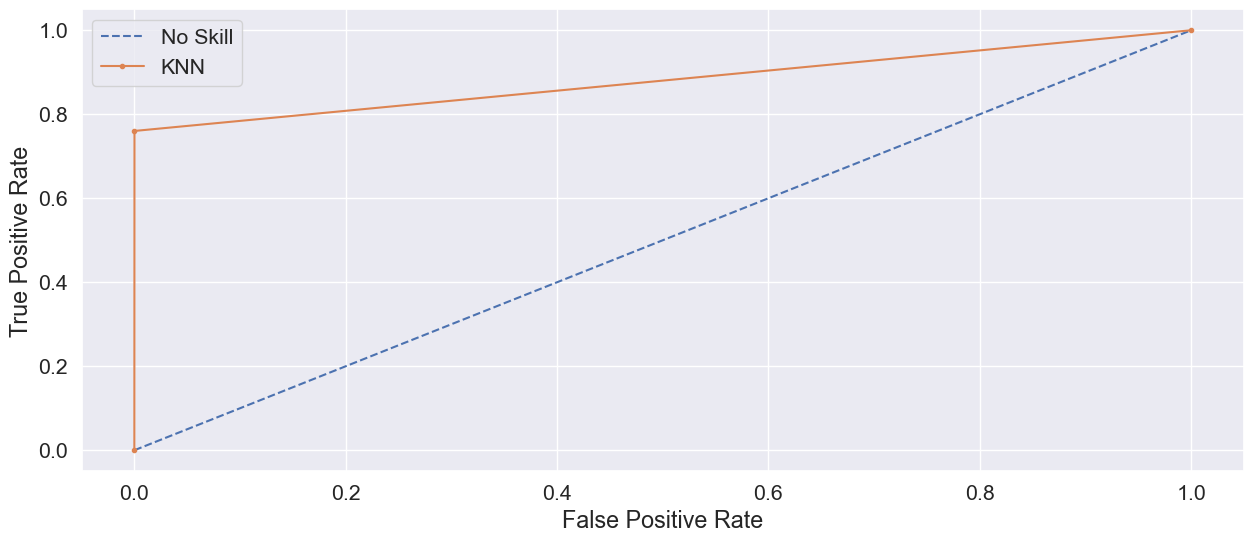


1. **K Nearest Neighbour algorithm:**

The below shown figure represents the AUC score and ROC curve of the K Nearest Neighbour algorithm.

Figure

*ROC Curve of KNN*



### Future Work:

There are diverse ways of tackling the credit card prediction problem. The extensions to the current work can be done by finding wide variety of datasets which contain more observations and certain other variables. There is a scope of doing the work differently at every stage of implementing this project, starting from data collection, to evaluating the models. Apart from the current machine learning algorithms used in this project, other supervised algorithms can be used on the dataset and their accuracies can be compared. The model’s performance can be analysed using different metrics such as cross-validation score. The machine learning algorithms’ behaviour can be examined by varying the data, i.e., by adding new data or splitting the dataset in a different ratio. Research can be done in this area to understand what factors can be weighed most in predicting credit card transactions. The machine learning algorithms’ performance can be improved by calculating the optimal parameters using hyperparameter optimisation techniques.