**Data Collection and Preprocessing Phase**

**Data Collection:**

ML depends heavily on data. It is the most crucial aspect that makes algorithm training possible. So this section allows you to download the required dataset.

**Data Set :**

In this project we have used PS\_20174392719\_1491204439457\_logs.csv data. This data is downloaded from kaggle.com. Please refer to the link given below to download dataset

**Visualizing And Analyzing Data:**

As the dataset is downloaded. Let us read and understand the data properly with the help of some visualisation techniques and some analysing techniques.

Note: There are a number of techniques for understanding the data. But here we have used some of it. In an additional way, you can use multiple techniques.

**Importing The Libraries:**

Import the necessary libraries as shown in the image. (optional) Here we have used visualisation style

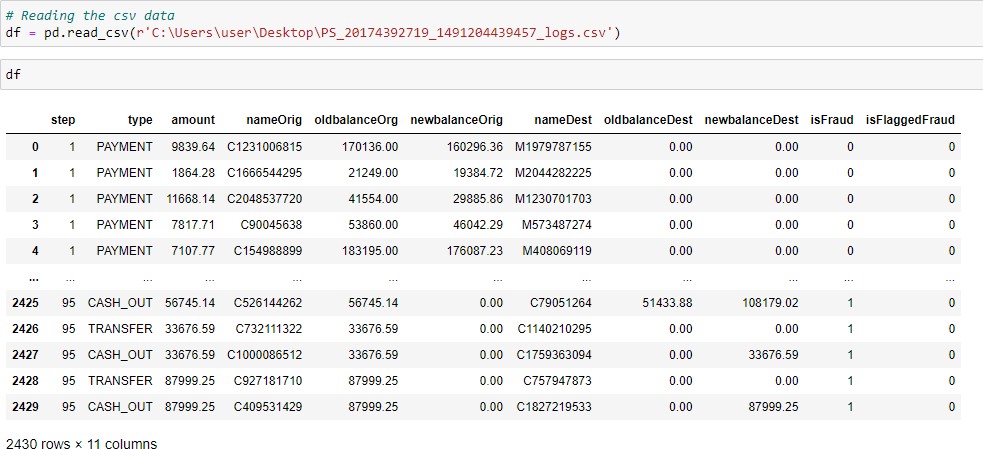
as five hundred thirty eight.



**Read The Dataset:**

Our dataset format might be in .csv, excel files, .txt, .json, etc. We can read the dataset with the help of pandas.

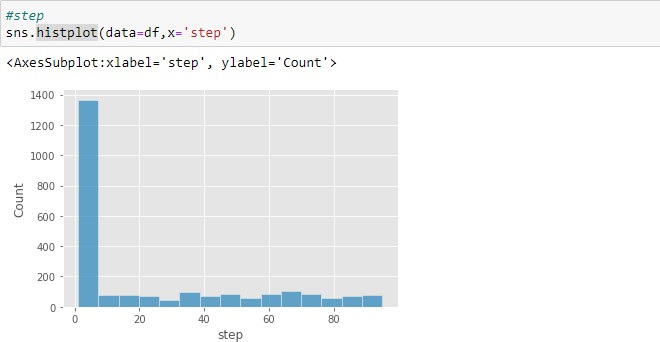
In pandas we have a function called read\_csv() to read the dataset. As a parameter we have to give the directory of the csv file.



**Univariate Analysis :**

In simple words, univariate analysis is understanding the data with a single feature. Here I have

displayed the graph such as histplot .

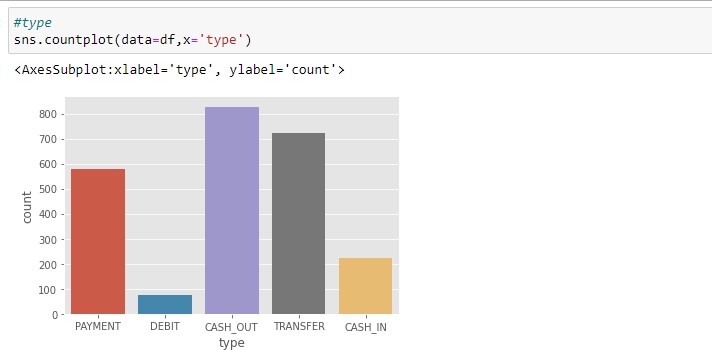


The distribution of one or more variables is represented by a histogram, a traditional visualisation tool,

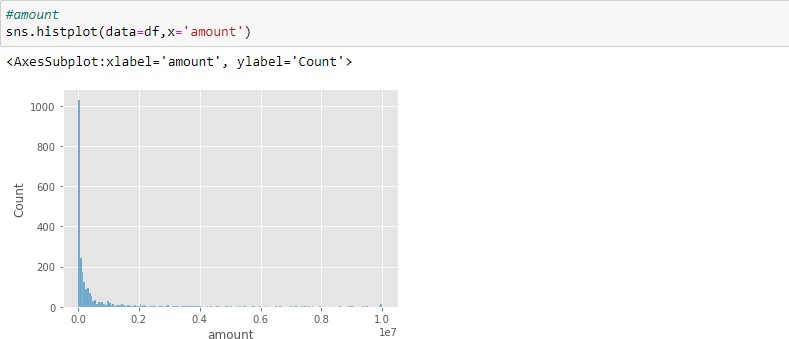
by counting the number of observations that fall within.



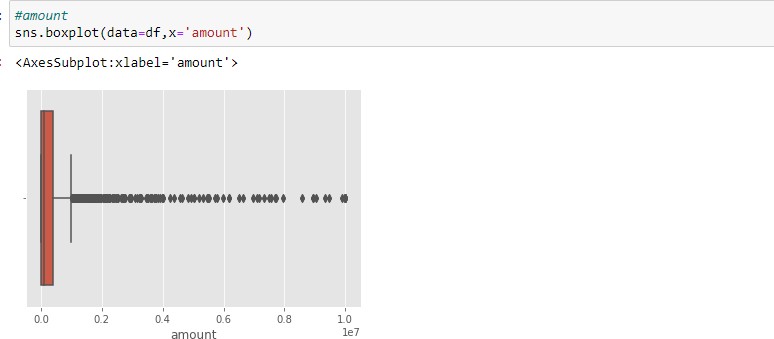
Here, the relationship between the step attribute and the boxplot is visualised.



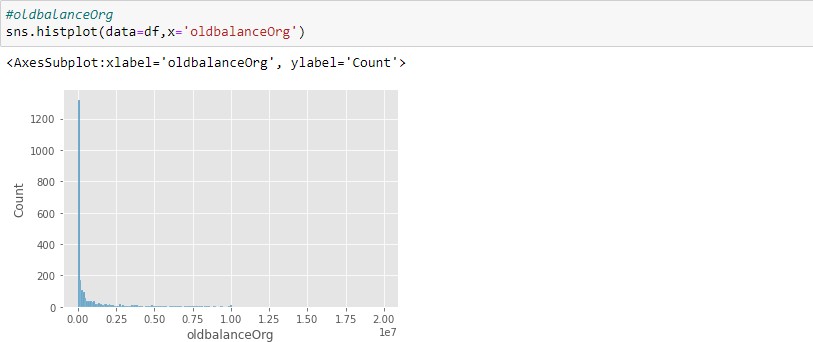
Here, the counts of observations in the type attribute of the dataset will be displayed using a countplot.



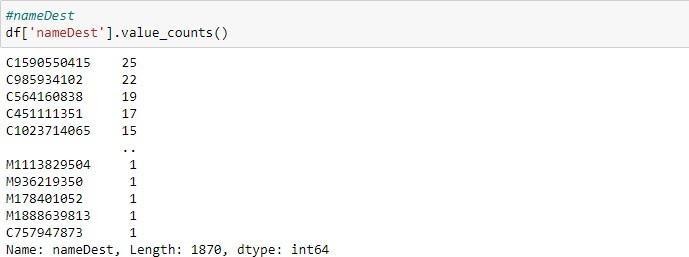
By creating bins along the data's range and then drawing bars to reflect the number of observations that fall within the amount attribute in the dataset.



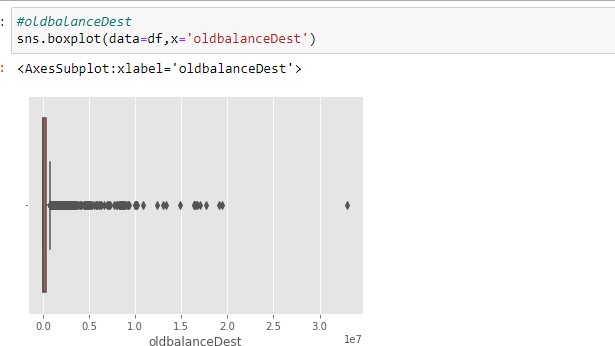
Here, the relationship between the amount attribute and the boxplot is visualised.



By creating bins along the data's range and then drawing bars to reflect the number of observations that fall within the oldbalanceOrg attribute in the dataset.



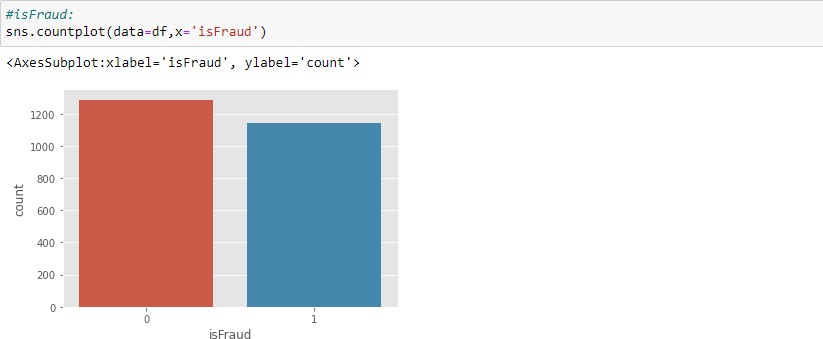
utilising the value counts() function here to determine how many times the nameDest column appears.



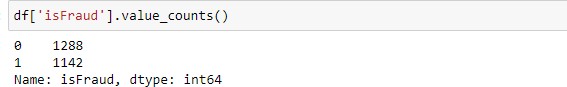
Here, the relationship between the oldbalanceDest attribute and the boxplot is visualised.



Here, the relationship between the newbalanceDest attribute and the boxplot is visualised.

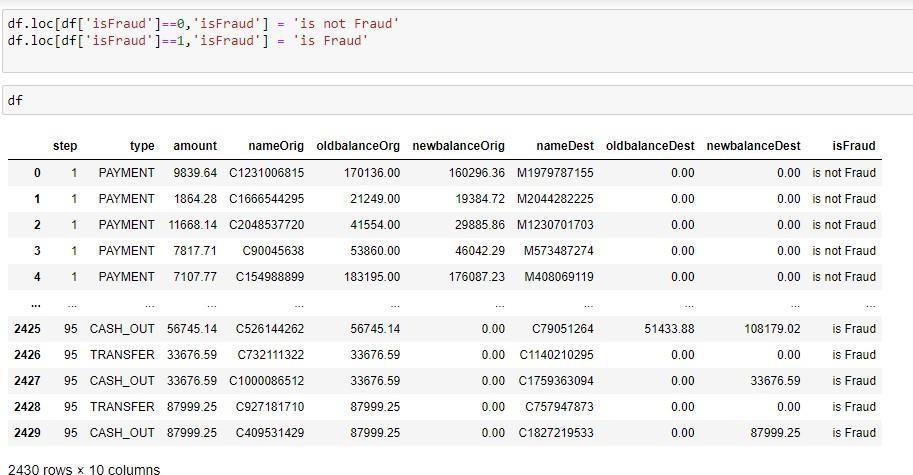


using the countplot approach here to count the number of instances in the dataset's target isFraud column.



Here, we're using the value counts method to figure out how many classes there are

in the dataset's target isFraud column.

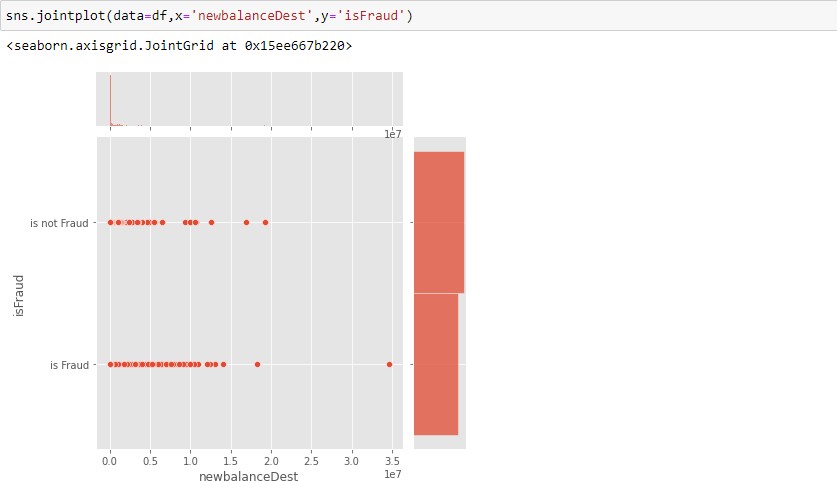


converting 0-means: is not fraud and 1-means: is fraud using the loc technique here

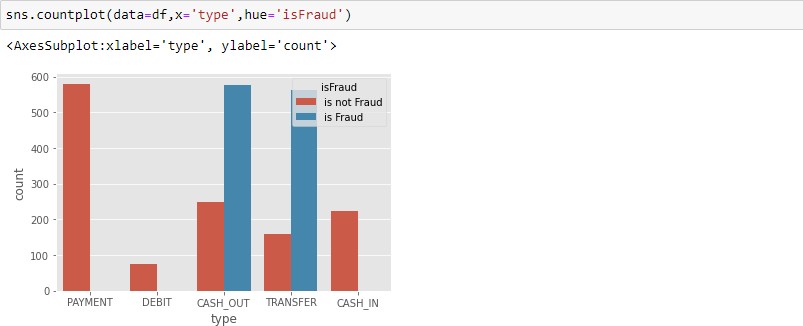
**Bivariate Analysis**

To find the relation between two features we use bivariate analysis. Here we are visualising the relationship between newbalanceDest and isFraud.

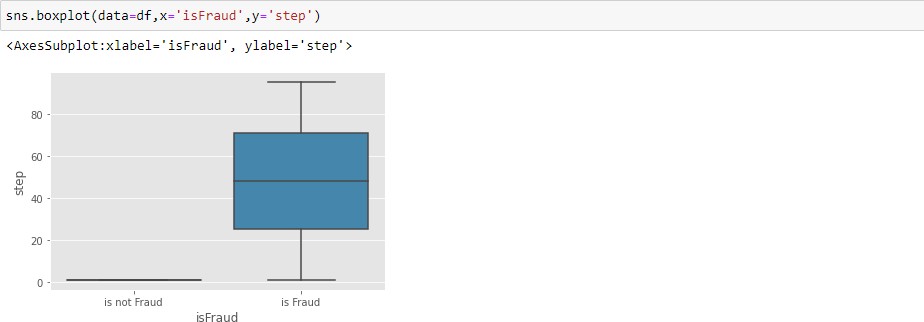
jointplot is used here. As a 1st parameter we are passing x value and as a 2nd parameter we are passing hue value.



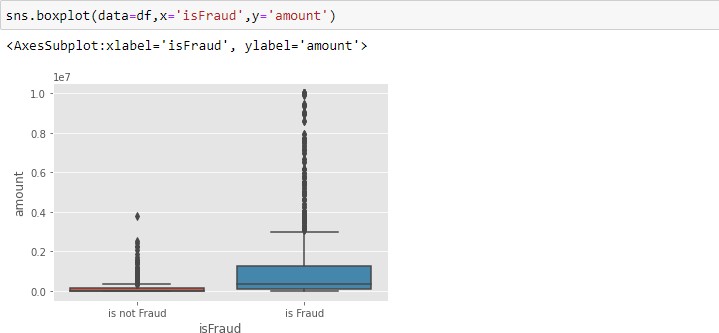
Here we are visualising the relationship between type and isFraud.countplot is used here. As a 1st parameter we are passing x value and as a 2nd parameter we are passing hue value.



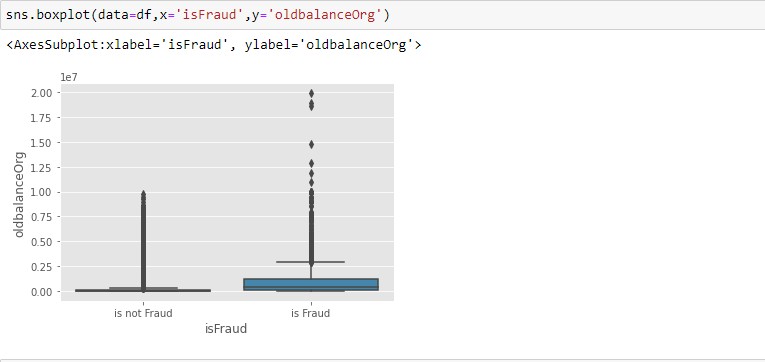
Here we are visualising the relationship between isFraud and step.boxtplot is used here. As a 1st parameter we are passing x value and as a 2nd parameter we are passing hue value.



Here we are visualising the relationship between isFraud and amount.boxtplot is used here. As a 1st parameter we are passing x value and as a 2nd parameter we are passing hue value.

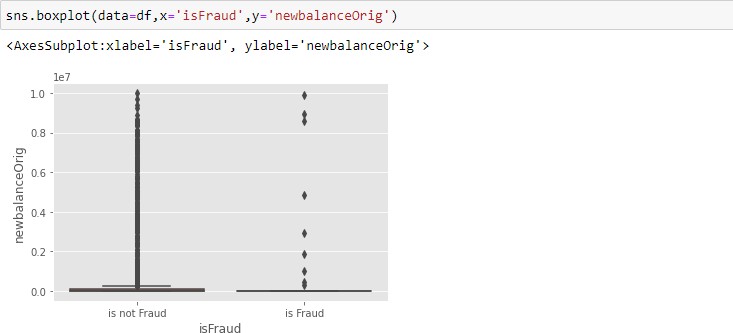


Here we are visualising the relationship between isFraud and oldbalanceOrg. boxtplot is used here. As a 1st parameter we are passing x value and as a 2nd parameter we are passing hue value.

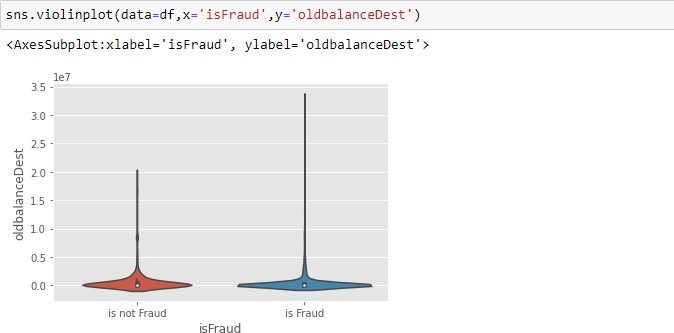


Here we are visualising the relationship between isFraud and newbalanceOrig. boxtplot is used here.

As a 1st parameter we are passing x value and as a 2nd parameter we are passing hue value.

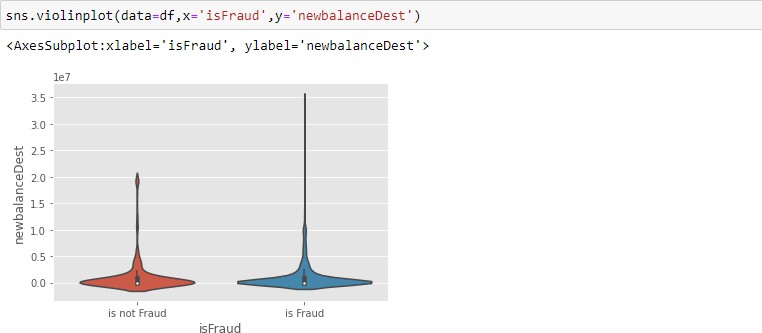


Here we are visualising the relationship between isFraud and oldbalanceDest. violinplot is used here. As a 1st parameter we are passing x value and as a 2nd parameter we are passing hue value.



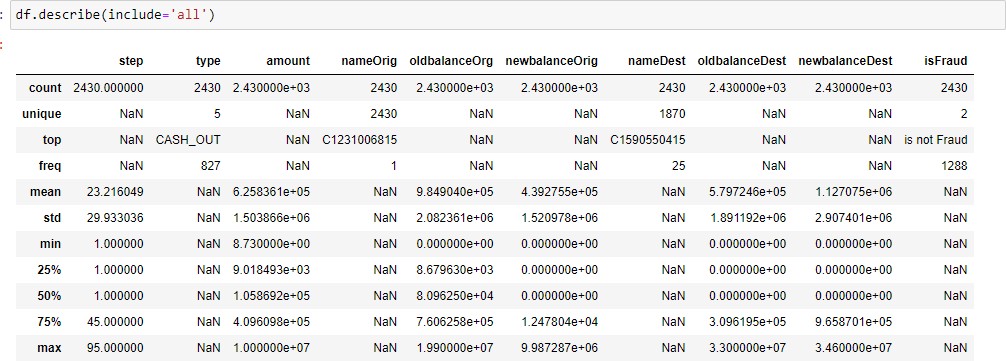
Here we are visualising the relationship between isFraud and newbalanceDest. violinplot is used here.

As a 1st parameter we are passing x value and as a 2nd parameter we are passing hue value.



**Descriptive Analysis**

Descriptive analysis is to study the basic features of data with the statistical process. Here pandas has a worthy function called describe. With this describe function we can understand the unique, top and frequent values of categorical features. And we can find mean, std, min, max and percentile values of continuous features.



# Data Pre-Processing

As we have understood how the data is, let's pre-process the collected data.

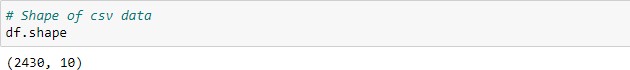
The download data set is not suitable for training the machine learning model as it might have so much randomness so we need to clean the dataset properly in order to fetch good results. This activity includes the following steps.

Handling missing values

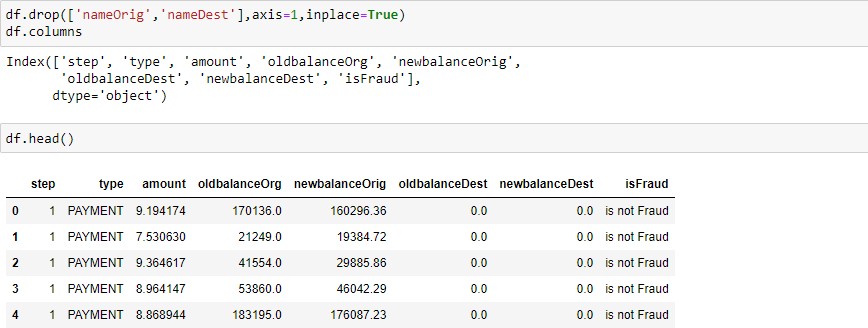
Handling Object data label encoding

Splitting dataset into training and test set

Note: These are the general steps of pre-processing the data before using it for machine learning. Depending on the condition of your dataset, you may or may not have to go through all these steps.



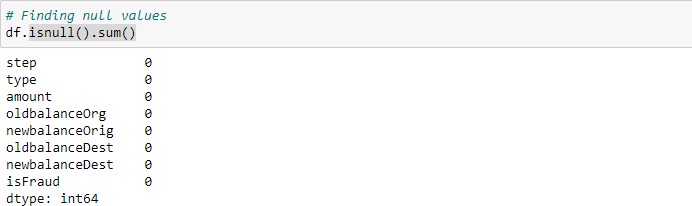
Here, I'm using the shape approach to figure out how big my dataset is



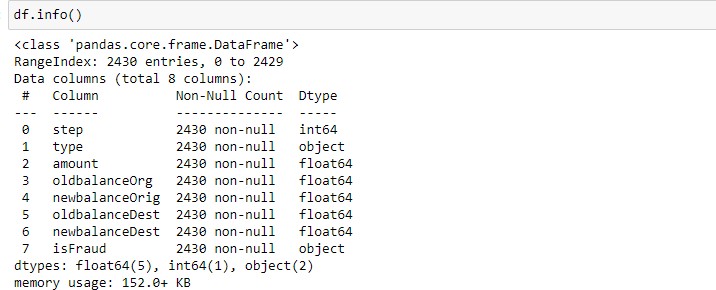
here, the dataset's superfluous columns (nameOrig,nameDest) are being removed using the drop method.

# Checking For Null Values

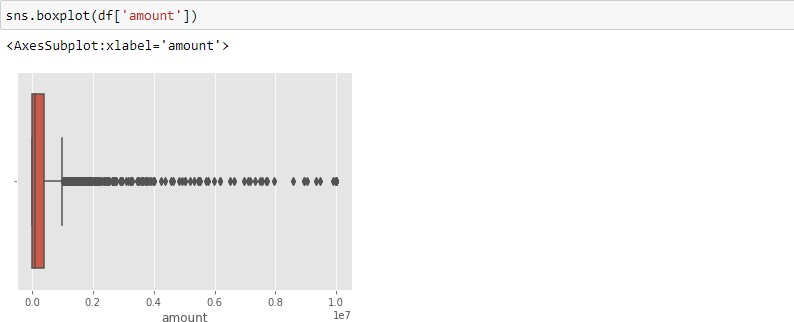
Isnull is used (). sum() to check your database for null values. Using the df.info() function, the data type can be determined.



For checking the null values, data.isnull() function is used. To sum those null values we use the .sum() function to it. From the above image we found that there are no null values present in our dataset.So we can skip handling of missing values step.

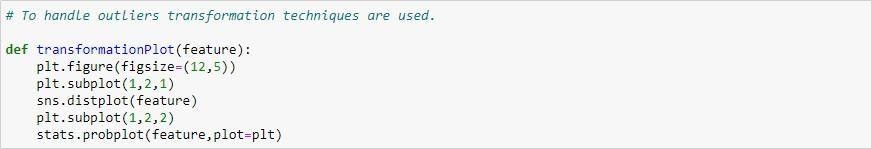
 determining the types of each attribute in the dataset using the info() function

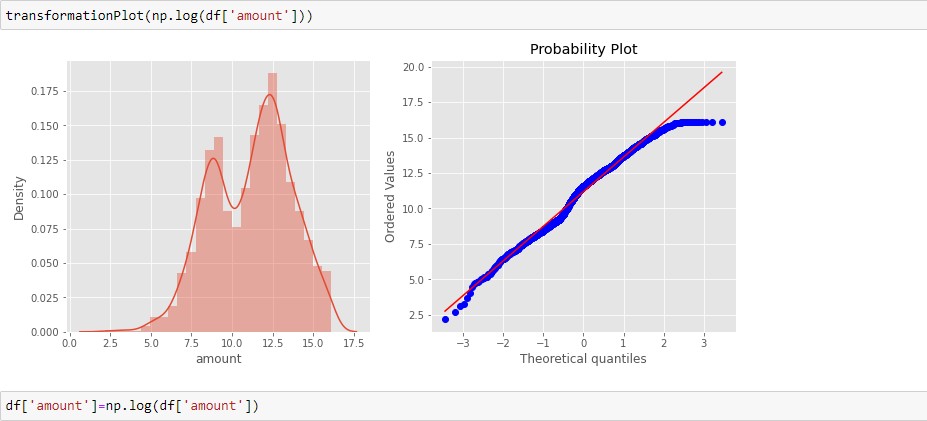
**Handling Outliers**



Here, a boxplot is used to identify outliers in the dataset's amount attribute.

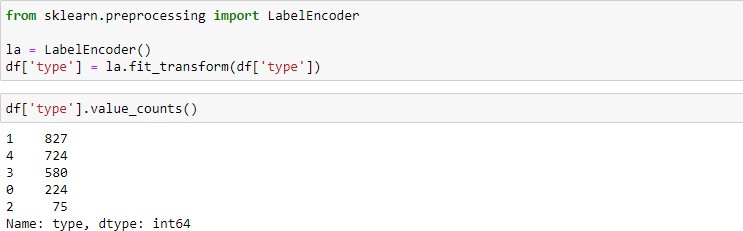






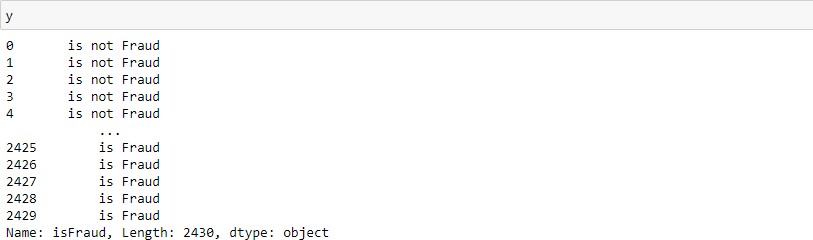
Here, transformationPlot is used to plot the dataset's outliers for the amount property.

# Object Data Labelencoding



using label encoder to encode the dataset's object type

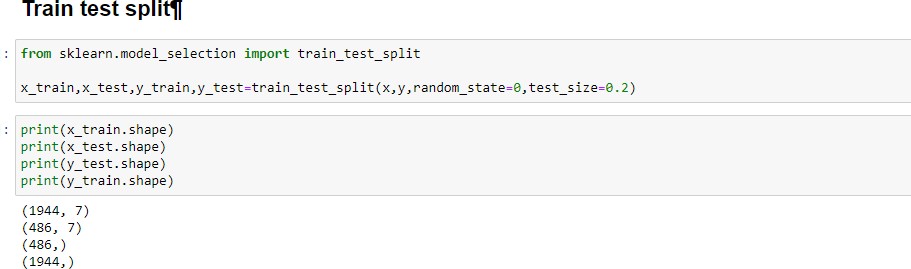




**Splitting Data Into Train And Test**

Now let’s split the Dataset into train and test setsChanges: first split the dataset into x and y and then split the data set.

Here x and y variables are created. On x variable, df is passed with dropping the target variable. And my target variable is passed. For splitting training and testing data we are using the train\_test\_split() function from sklearn. As parameters, we are passing x, y, test\_size, random\_state.



**Evaluating Performance Of The Model And Saving The Model**

From sklearn, accuracy\_score is used to evaluate the score of the model. On the parameters, we have given svc (model name), x, y, cv (as 5 folds). Our model is performing well. So, we are saving the model is svc by pickle.dump().

