|  |
| --- |
|  |
| Data Mining |
| Case Study Assignment |

|  |
| --- |
|  |

Table of Contents

[Problem Statement: 2](#_Toc80612234)

[Exploratory Data Analysis 2](#_Toc80612235)

[Import Libraies 2](#_Toc80612236)

[Load the Dataset: 3](#_Toc80612237)

[Data Analsysis 3](#_Toc80612238)

[Shape, Collumn and Dtypes 3](#_Toc80612239)

[Mean, Count and Standard Deviation 4](#_Toc80612240)

[Skewness and Kurtosis of the Target Data 5](#_Toc80612241)

[Distribution Plot with respect to different columns 5](#_Toc80612242)

[Information about Data, Histogram, Corelation and HeatMap 6](#_Toc80612243)

[Count Plot 8](#_Toc80612244)

[Kde Plot 8](#_Toc80612245)

[RegPlot 9](#_Toc80612246)

[Value Count and Converting it into a Data Frame 10](#_Toc80612247)

[GroupBy Function 11](#_Toc80612248)

[Preprocess the Data 11](#_Toc80612249)

[Select the Training data & Test data 12](#_Toc80612250)

[Train and Test the Model 13](#_Toc80612251)

[Model Performance 14](#_Toc80612252)

[Ways of improving the model 15](#_Toc80612253)

**I performed the following steps to implement this case-study assignment**

* Problem Statement
* Exploratory Data Analysis
* Preprocess the data
* Select Training data, test data
* Train the model
* Test the model (Predictions and reporting)
* Evaluate the model performance
* Suggest ways of improving the model

**To implement credit card fraud detection algorithm I performed the following steps**

* Import Libraries
* Load the data set
* Analyze the data
* Data Wrangling
* Separating the independent and dependent variables (only “fraud” is considered as dependant variable(target variable) while all others are considered as independent variables)
* Train & Test Split (80% data is being for the training purpose while 20% data is being considered for test purpose)
* Training our data using different Machine Learning models i.e. KNN Classifier, Decision Tree Classifier, Random Forest Classifier
* Testing the model using the test data set and checking the accuracy, sensitivity, confusion matrix, area under the curve to check out which models performs better.

# Problem Statement:

According to the Data Breach Index, more than 5 million records are being stolen on a daily basis. In today’s digital world where trillions of Card transaction happens per day, detection of fraud is challenging. In this case study assignment, we construct a machine learning model from the dataset i.e."card\_transdata" given by considering different Supervised Machine Learning Algorithms for calssification purpose i.e. K-Nearest Neighbors, Decision Tree Classifier, Random Forest Classifier. We then analyzed the performance of our model in terms of accuracy, confusion matrix

# Exploratory Data Analysis

## Import Libraies

* import pandas as pd

Pandas library is used for data manipulation and analysis

* import numpy as np

Numpy provides a large set of numeric datatypes that you can use to construct arrays.

* import seaborn as sns

Seaborn is a library for making statistical graphics in Python

* import matplotlib.pyplot as plt

Matplotlib. pyplot is a collection of command style functions that make matplotlib work like MATLAB.

* from sklearn.model\_selection import train\_test\_split

train\_test\_split is a function in Sklearn model selection for splitting data arrays into two subsets: for training data and for testing data.

* from sklearn.tree import DecisionTreeClassifier

Machine Learning Model we are going to use

* from sklearn.ensemble import RandomForestClassifier

Machine Learning Model we are going to use

* from sklearn.metrics import precision\_recall\_curve

Compute precision-recall pairs for different probability thresholds.

* from sklearn.metrics import plot\_precision\_recall\_curve

To plot the percision recall curve

* plt.ticklabel\_format(useOffset=False)
* import warnings

Warning messages are typically issued in situations where it is useful to alert the user of some condition in a program, where that condition (normally) doesn't warrant raising an exception and terminating the program.

* warnings.filterwarnings('ignore')
* %matplotlib inline

%matplotlib inline sets the backend of matplotlib to the 'inline' backend

## Load the Dataset:

data = pd.read\_csv("card\_transdata.csv")



## Data Analsysis

### Shape, Collumn and Dtypes

In data analysis, we check out the shape of the dataset using data.shape and find out the index/name of the columns using data.columnsand to covert this into an array we write data.columns.values. Then we check out the data type of each of the column using data.dtypes and list bascially print out the list of data types we have.



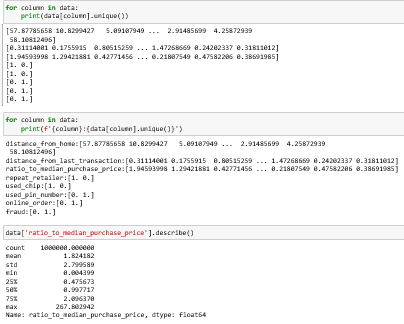
Now, to find the unique values ineach of the columns we just simple do a for loop which prints out all the unique values in each of the collumn. Now if want to print the unique values in each column along with their column name, i will simply write

for column in data:

print(f'{column}:{data[column].unique()}')

## Mean, Count and Standard Deviation

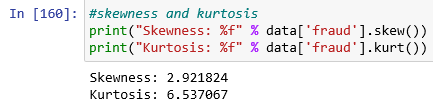
To check mean, count, standard deviation in any collumn we simply use .describe() function.



### Skewness and Kurtosis of the Target Data

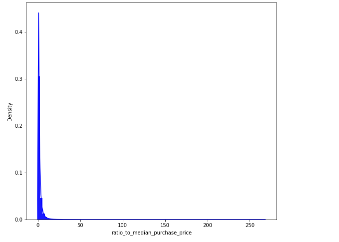
The skewness is a parameter to measure the symmetry of a data set and the kurtosis to measure how heavy its tails are compared to a normal distribution

kurtosis(array, axis=0, fisher=True, bias=True) function calculates the kurtosis (Fisher or Pearson) of a data set. It is the the fourth central moment divided by the square of the variance.



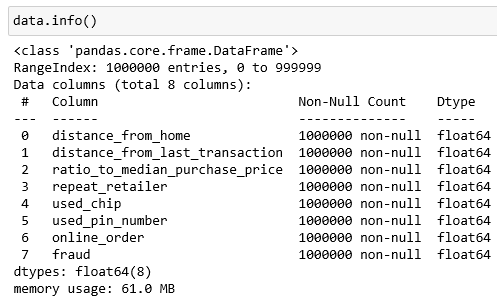
### Distribution Plot with respect to different columns

The distplot represents the univariate distribution of data i.e. data distribution of a variable against the density distribution.



### Information about Data, Histogram, Corelation and HeatMap

In the next step we write, data. info(), to check if there are any null values in any collumn.



Now, we plot a histogram of all the columns we have using,

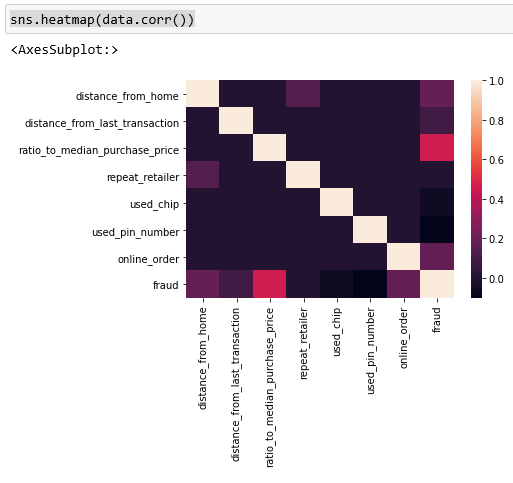


We find the corr() among each collumn using data. corr()



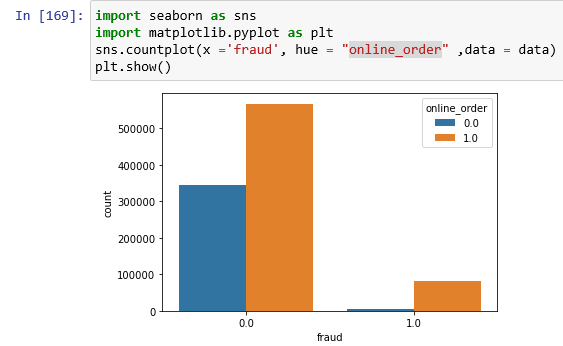
In the next step, we create a heatmap with respect to each collumn using

sns.heatmap(data.corr())



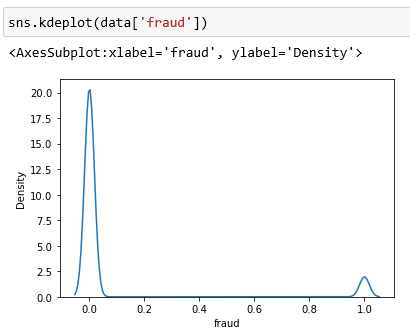
### Count Plot

Now, create count plot of each independant variable with respect to a dependant variable an example plot is shown here.



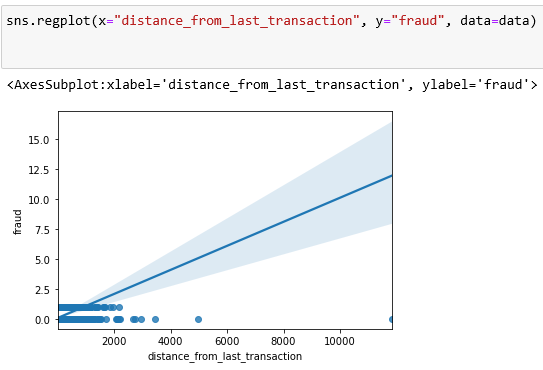
### Kde Plot

In the next step, we create kde plot to Plot univariate or bivariate distributions using kernel density estimation.



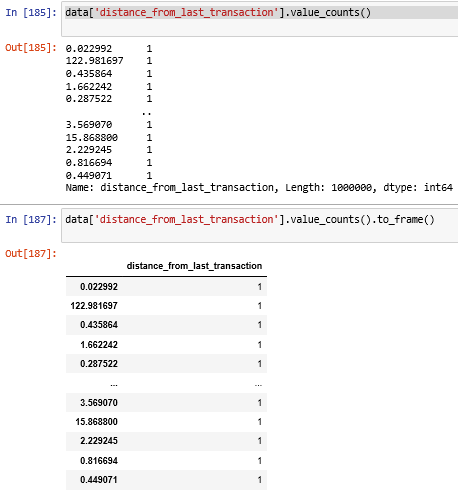
### RegPlot

We also create regplot, as shown in the image below



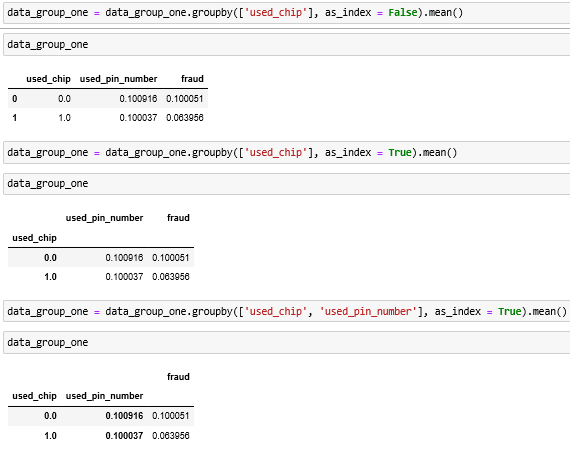
### Value Count and Converting it into a Data Frame

In the next step, we do value count with respect to each collumn and create a data frame using to\_frame()



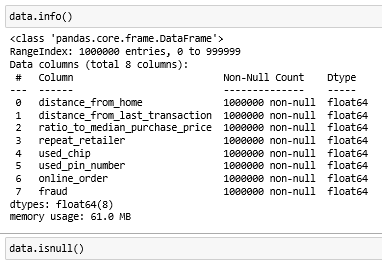
### GroupBy Function

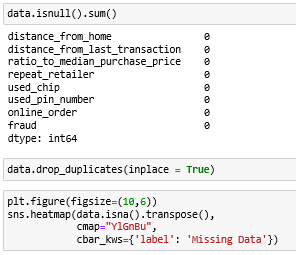
In the next step we group by different collumns and check the relation among them



# Preprocess the Data

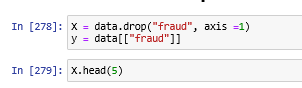
Now we will preprocess the data, will check if there are any null values/ missing or if there exist any duplicates, we will perform all these steps while doing preprocessing of the data.



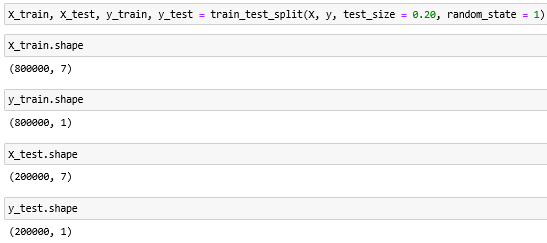


# Select the Training data & Test data

Now we separate the training and the target data as shown in the figure below

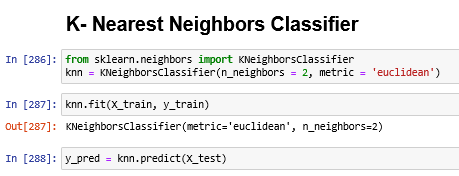


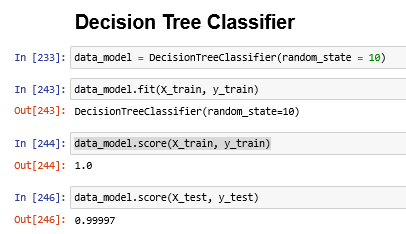
Now we do the train and test split

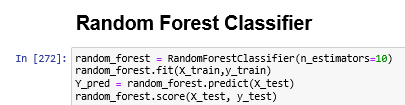


# Train and Test the Model

Now we train our model using different supervised machine learning algorithms i.e. K- Nearest Neighbors Classifier, Decision Tree Classifier, Random Forest Classifier and then we test our model using the unseen data

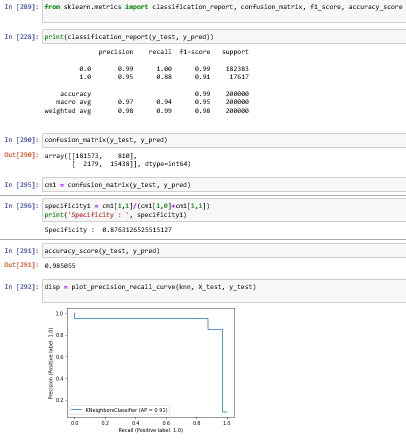






# Model Performance

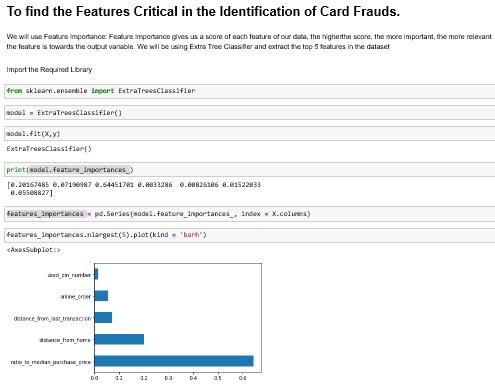
We evaluate model performance considering different parameters accuracy, specificity, the area under the precision-recall curve, confusion matric



# Ways of improving the model

The following are the two possible ways which can help us to improve the accuracy of the model

1. We can further improve our model by doing feature selection which means considering only those collumns which have an impact on output and removing the other collumns which donot have impact on output orvery least impact. I have shown as an example as well how we can do feature selection by using one of the techniques of feature selection which is ExtraTreeClassifier



1. We can also improve our model by doing hyperparameter tunning means optimizing our model using Grid Search CV or Random Search CV which enables us to select the parameters which gives us the best accuracy.

# Conclusion:

In this case study assignment I designed a credit card fraud detection model. The results i.e. accuracy, sensitivity, confusion matrix show the effectiveness of our design and it can be seen that the required performance can be achieved through the design. However, Machine Learning techniques are not limited only to credit card fraud detection or any other specific application. Its true potential lies in that how we can take maximum advantage of this. As of today we are working on different demand forecasting technology and trying to understand with what added parameters it can be helpful for the modern businesses by solving the tasks such as credit card fraud detection, forecasting the customer engagement, future trends, helping in the marketing campaigns, effective resources utilization, minimizing financial risks and helping in the brand development.