Letter of Transmittal

Date: 19th May, 2022

To

Nasrin Khatun

Assistant professor

Jahangirnagar University · Department of Statistics

Subject: Submission of final research project on - **A predictive approach for allocating digital credit in the MSME sector using artificial neural network (ANN)**

Dear Madam,

With due respect, as a student of Jahangirnagar University, I have prepared my final report on Machine Learning in digital credit distribution. I have tried my level best to follow your guidelines in every aspect of planning of this report. I have also collected what I believe to be the most important information to make this report as specific and accurate as possible. I am honestly thankful for your guidance during the preparation of this draft of the report. I hope you will appreciate my effort. I have done the study in a complete form and I have tried my level best to conduct this in a professional manner. It is true that it could have been done in a better way if there were no limitations. I hope you will assess my report considering the limitations of the study.

Sincerely yours,

....
Md. Nazrul Islam
ID: 201900303076
PMASDS Program
3rd Batch
Jahangirnagar University

Declaration

I, hereby declare that the internship report entitled Merchandiser response about the **predictive approach for allocating digital credit in the MSME sector using ANN,** the result of my own effort after the completion of two months work under the supervisors of Nasrin Khatun, Assistant professor, Jahangirnagar University · Department of Statistics.

I further affirm that the work reported in this internship is original and is not part of any other students for the completion of PMADS (Masters in Applied Statistics and Data Science) or other degree have submitted the whole of the report.

Md. Nazrul Islam
ID: 201900303076
PMASDS Program
3rd Batch
Jahangirnagar University

Certificate of the Supervisor

This is to certify that Md. Nazrul Islam a student of PMASDS, ID No. 201900303076 successfully completed his project **predictive approach for allocating digital credit in the MSME sector using ANN** under my supervision as a partial fulfillment of the requirements for the award of PMASDS degree.

He has done his job according to my supervision and guidance. He has tried his best to do this assignment successfully. I think this program will help him to build up his future career.

I wish her success.

…………………………

Nasrin Khatun

Assistant professor

Jahangirnagar University · Department of Statistics

Abstract:

The study is conducted to solve the problem in distribution of Micro credit to the MSME sector in Bangladesh usings Artificial Intelligence.

As per the economic census 2013, We have around 8 million MSMEs in Bangladesh. As per the BTRC report, 2021 our mobile internet penetration is 65.56% of our total 166 million population. As per this ratio we can say that over 5 million of our MSMEs are under internet coverage now. Because of this digitalization nowadays there are few startup companies working on this sector to leverage this sector. One of them is "**TallyKhata App**". This is a free day by day digital bookkeeping app which is developed by Progoti Systems Limited. Currently 6 Million + download and 80% of this user base is MSMEs of Bangladesh. So now these apps generate huge data. Hence: we can facilitate them digitally by analyzing these data in different ways. One of them could be: Digital assessment and financing.

As per the **world bank report**: Financing Solutions For Micro, Small And Medium Enterprises In Bangladesh 2019 some of the key financing challenges of MSMEs are - (i) **Poor quality of collateral** (ii) **Inadequate formal documentation** (iii) **Ill-defined business plans**.

The MSME segment is also known as 'Missing Middle segment'. The micro small and medium-sized enterprises that have the most difficulty in accessing bank finance.

I have developed a Machine Learning Model using Artificial Neural Network (ANN) Algorithm to solve a credit eligibility Problem. Here, by analyzing the historical digital bookkeeping data of the MSMEs (Micro Small and Medium Enterprises) I tried to predict whether they are eligible for a bank loan or not. The solution of this binary classification problem may help the Banks, NBFIs and other NGOs who are currently distributing the credit in the MSME sector to assess the creditworthiness of the merchants.

Finally after apply the MLP classifier abled to predict the credit eligibility with accuracy of 99.8 %

Keywords:

MSME - Micro and Small Medium Enterprise

BTRC -Bangladesh Telecommunication Regulatory Commission

NBFIs - Non Banking Financial Institutions

FMCG - Fast Moving Consumer Goods

GDP - Gross Domestic Production

IFC - International Financial Corporation

MFIs - Mobile Financial Institutions

NN -Neural Network

ANN - Artificial Neural Network

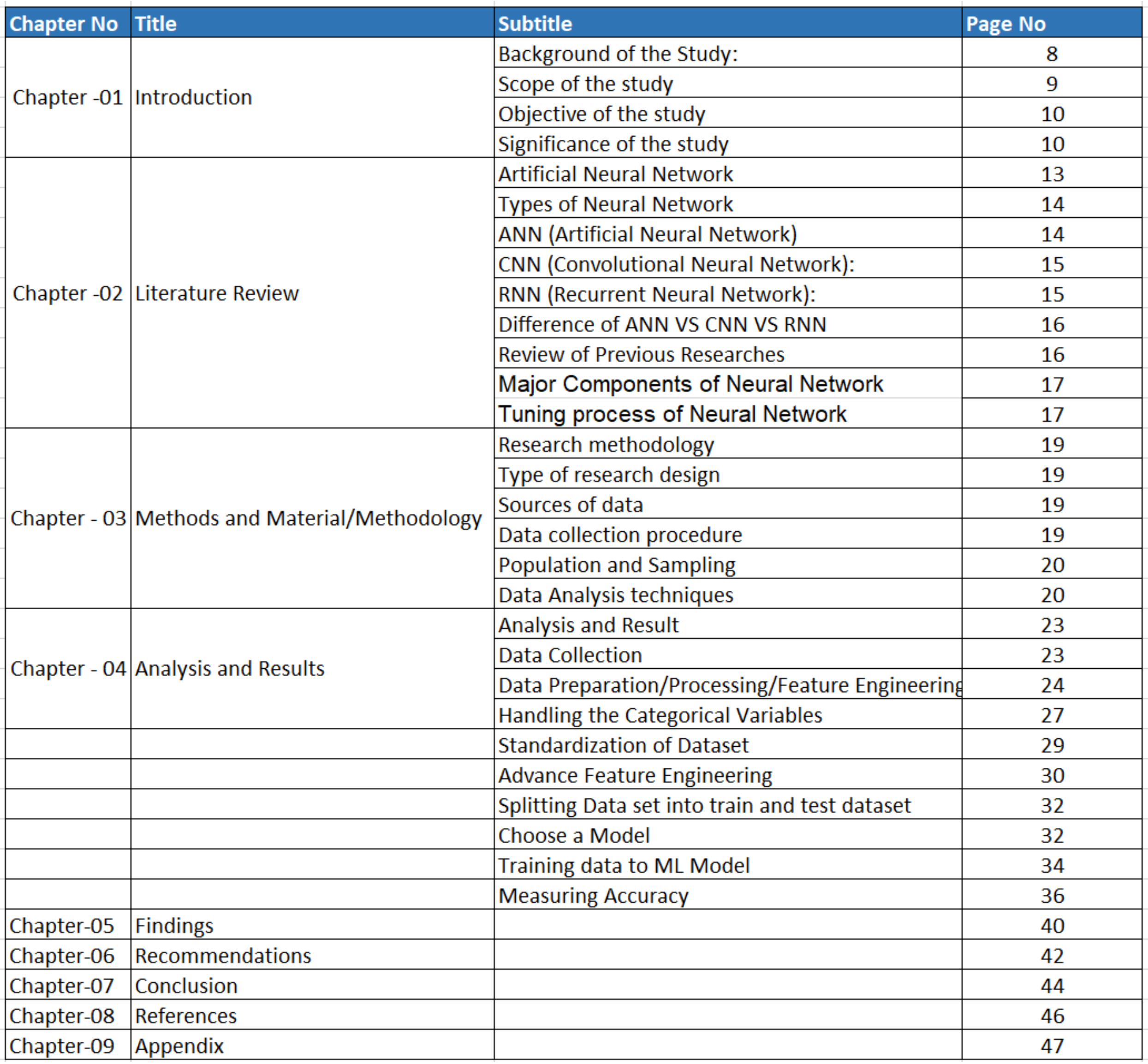
RNN - Recurrent Neural Network

CNN - Convolutional Neural Network

LSTM - Long Short Term Memory

MLP - Multilayer Perceptron

Table of Contents



Chapter - 01

Introduction

Introduction:

1. Background of the Study:

As per the economic census 2013, We have around 8 million MSMEs in Bangladesh. As per the BTRC report, 2021 our mobile internet penetration is 65.56% of our total 166 million population. As per this ratio we can say that over 5 million of our MSMEs are under internet coverage now. Because of this digitalization nowadays there are few startup companies working on this sector to leverage this sector. One of them is “**TallyKhata App ''**. This is a free day by day digital bookkeeping app which is developed by Progoti Systems Limited. Currently 6 Million + download and 80% of this user base is MSMEs of Bangladesh. So now these apps generate huge data. Hence: we can facilitate them digitally by analyzing these data in different ways. One of them could be: Digital assessment and financing.

According to the Economic census 2013, there are about 13.7 million people involved with micro, 6.6 million with small, 0.7 million people with medium and 3.4 million people with large enterprises in Bangladesh. That means the MSMEs directly created about over 2 billion jobs.

As per the economic census 2013, we have around 8 million MSMEs in Bangladesh, According to the BTRC Internet penetration report December 2021, we have 65.56% mobile internet penetration as per this out of 8 million MSMEs, we get more than 5 million+ of them are under mobile internet connectivity and according to annual digital growth report 2021, annually internet adoption growth rate is 19.2%.

2. Scope of the study :

Though such high mobile internet penetration in the country our MSME sector is still not that much digitized as a result day by day they are facing challenges such as -

Most important challenge MSMEs face is that they have **limited and complicated access to finance**, a **high-Interest rate in microfinance** imposed by Banks and NBFIs. On the other hand, there is an absence of harmonizing tax and tariff policies for the MSMEs in our country.

Except this some of the few other challenge they face such as -

1. Umpteen accounting khata in the shop. Very often after credit sales, MSMEs face credit recovery hassle. Moreover, they do not have digital evidence against credit sales like digital sales sleep.

2. A large group of MSMEs still do not have digital identity as a result they are not able to sell the goods and services remotely. Most of them maintain ⅘ employees attendance and salary khata in the shop, maintenance of multiple khata is an ache.

3. MSMEs do not have a digital inventory management system which causes a waste of time in B2B ordering.

4. The absence of digital guidance and training causes unawareness about market forces and insufficient digital adoption among remote area MSMEs. Supplier power is high on FMCG/Grocery merchants as they are not price-sensitive and uneducated regarding the product.

From this above discussion we can see that due to the availability of technology like Android phones, Internet , Free Mobile Apps like '**TallyKhata**' out MSMEs now generate a lot of data on - Day by day business, shop images, Geo location etc. On the other hand they are facing limited financing issues due to lack of collateral, documents. Hence Digital Lending could be an area where data science can help both parties merchants and Lenders to come into a common and trustable negotiation.

Here we are talking about the scope of data science in lending MSMEs. As we have most of these MSMEs digital data available in our database, which may help the Banks/NBFIs to assess them digitally using ML/AI Model.

3. Objective of the study :

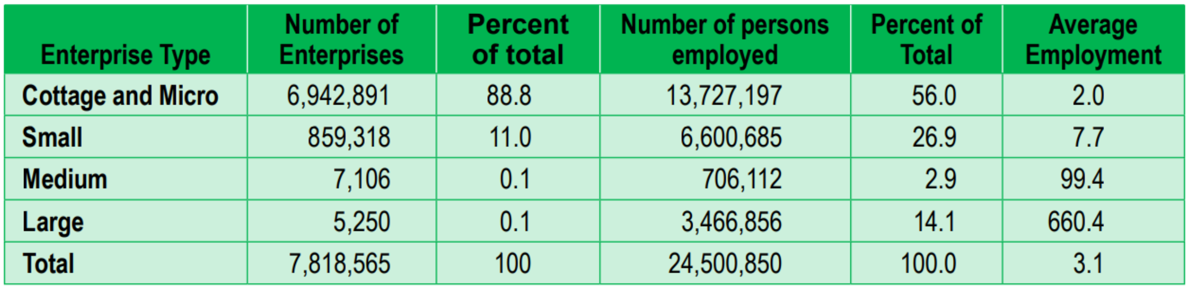
The main objective of the report is to find a way around distributing the Credit digitally properly to the MSME sector. However the details of the objectives of this study are as under:

1. Developing a Machine Learning model for digital assessment of creditworthiness.
2. Understanding the challenges of financing in the MSME sector.
3. Recommend how Machine Learning can mitigate the credit gap in the MSME sector.

4. Significance of the study:

MSMEs play a vital role in our national economies. Currently, according to the Asian development banks report 2015, This sector contributes approximately 25% of our Gross Domestic Production (GDP) which is still comparatively low compared to our neighbors India, China, Japan and Indonesia, in all of these countries contribution of this sector is greater than 60% of total GDP.

Table 15: Size and Employment of Enterprises in Bangladesh, 2013

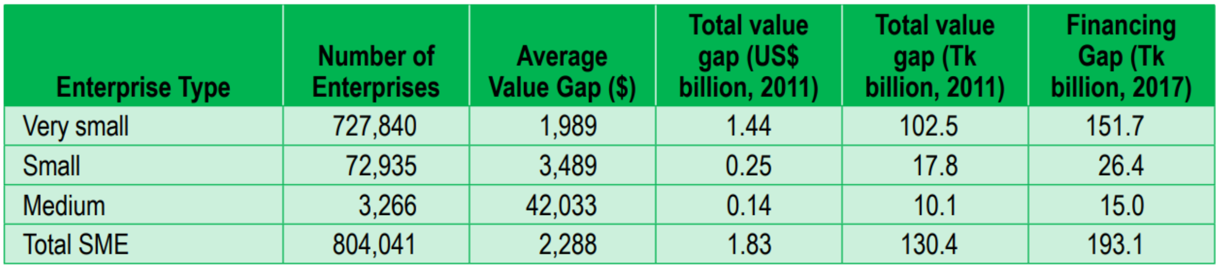


Enterprise Type	Number of Enterprises	Percent of total	Number of persons employed	Percent of Total	Average Employment
Cottage and Micro	6,942,891	88.8	13,727,197	56.0	2.0
Small	859,318	11.0	6,600,685	26.9	7.7
Medium	7,106	0.1	706,112	2.9	99.4
Large	5,250	0.1	3,466,856	14.1	660.4
Total	7,818,565	100	24,500,850	100.0	3.1

Source: BBS, Economic Census 2013.

The Ministry of Planning (Planning Division) reveals that MSMEs employ 87% of the industrial employment. 25% of the country's entire employment.

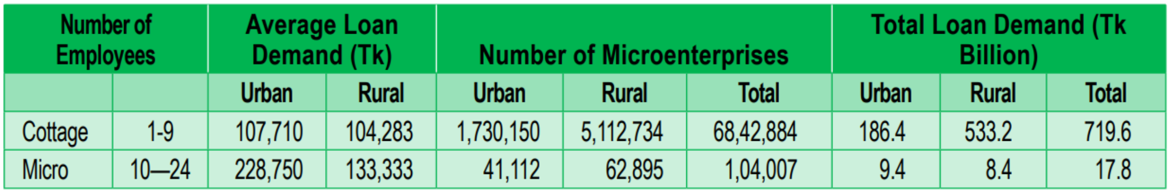
Table 6: IFC Estimates of Financing Gap, 2011 versus 2017



Enterprise Type	Number of Enterprises	Average Value Gap ($)	Total value gap (US$ billion, 2011)	Total value gap (Tk billion, 2011)	Financing Gap (Tk billion, 2017)
Very small	727,840	1,989	1.44	102.5	151.7
Small	72,935	3,489	0.25	17.8	26.4
Medium	3,266	42,033	0.14	10.1	15.0
Total SME	804,041	2,288	1.83	130.4	193.1

Source: IFC, 2013.

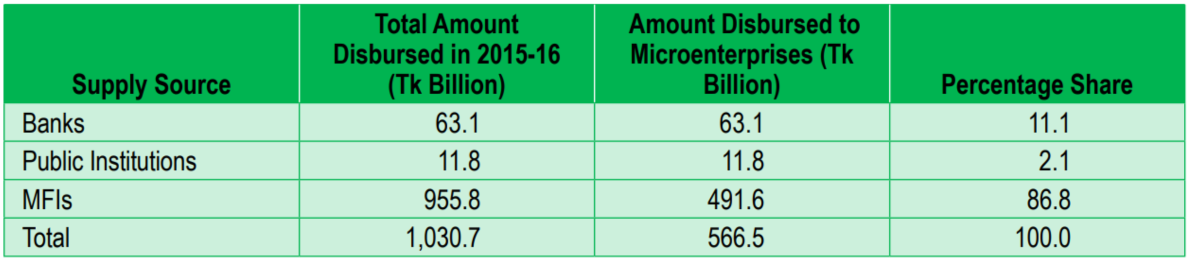
Table 4: Average Demand for Loans Among Microenterprises



Number of Employees		Average Loan Demand (Tk)		Number of Microenterprises			Total Loan Demand (Tk Billion)		
		Urban	Rural	Urban	Rural	Total	Urban	Rural	Total
Cottage	1-9	107,710	104,283	1,730,150	5,112,734	68,42,884	186.4	533.2	719.6
Micro	10—24	228,750	133,333	41,112	62,895	1,04,007	9.4	8.4	17.8

Source: InM Microenterprise Survey, 2016, and Economic Census, 2013.

Table 5: Loan Supply for Microenterprises, June 2016



Supply Source	Total Amount Disbursed in 2015-16 (Tk Billion)	Amount Disbursed to Microenterprises (Tk Billion)	Percentage Share
Banks	63.1	63.1	11.1
Public Institutions	11.8	11.8	2.1
MFIs	955.8	491.6	86.8
Total	1,030.7	566.5	100.0

Source: Credit and Development Forum, 2016.
Note: MFIs= Microfinance Institutions.

As per the microenterprise survey there is a 170 Million financing gap in this sector.

Chapter - 02

Literature Review

Literature Review:

1. Artificial Neural Network:

At the heart of the neural network is the unit - Called node or neuron. A unit takes one or more inputs, multiplies each input by a parameter called weight then sums the weighted input’s values along with some basis value then feeds the value into an activation function. This output is then sent forward to the other neurals deeper in the neural networks if it exists.

Feedforward neural networks, also called multilayer perceptrons, are the simplest artificial neural networks used in any real-world setting. Neural network can be visualized as a series of connected layers that form a network connecting and observing feature values at one end, and the target value at the other end.

Neural networks with many hidden layers are called ‘deep’ networks and their application is called deep learning.

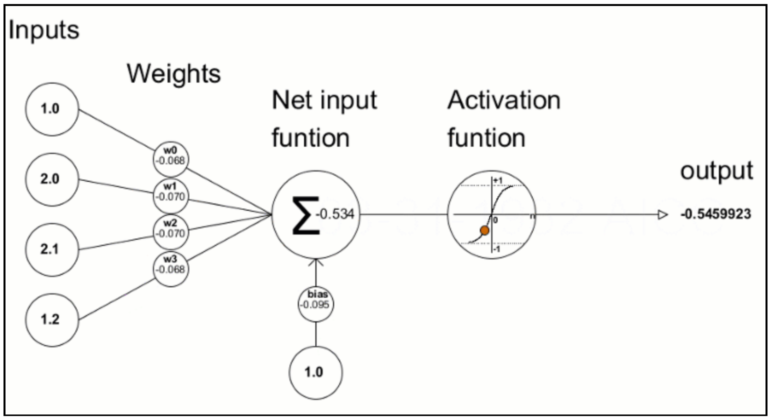
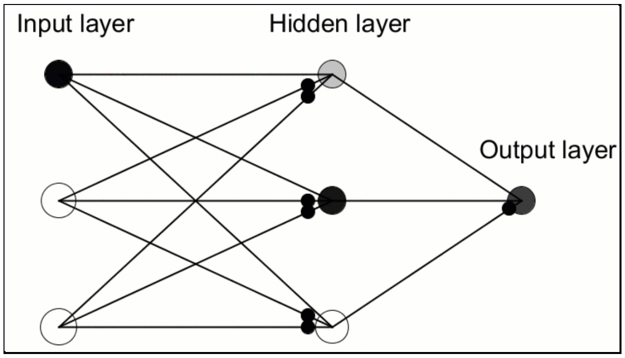


Image 01: Single neuron structure



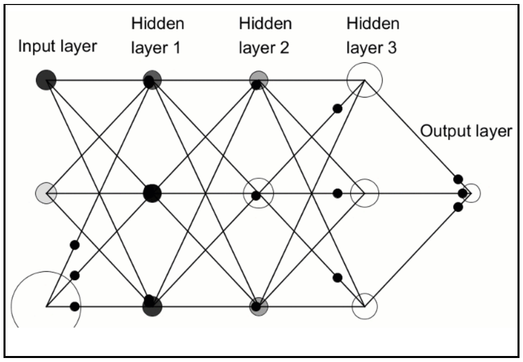


Image 02: Neural Network *Image 03: Deep* Neural Network

<u>Types of Neural Network:</u>

1. **<u>ANN (Artificial Neural Network):</u>**
    - Multiple perceptron in each layer
    - Feed Forward Neural Network
    - Pass information in one direction
    - May or May not have hidden layers

    Advantage:
    - They store information on the entire network
    - They have the ability to work with incomplete knowledge
    - They offer fault tolerance and have distributed memory
    - They offer us the ability to work with incomplete knowledge

    Disadvantage:
    - They have huge hardware dependency
    - They sometime have unexplained behavior which can leave us tormented with result
    - There is no specific rule for determining the structure of an artificial neural network and appropriate neural network structure is achieved through experience and trial and error.

1. **CNN (Convolutional Neural Network):**
    - Variation of multi layer perceptrons
    - One or more convolutional layers
    - Convolutional layers create feature maps for image region detection
    - Non linear processing

Advantage:

- Very High Accuracy in image recognition problem
- They are capable of automatically detecting important features without any human supervision.
- Wight sharing

Disadvantage:

- Do not encode the position and orientation of the object
- Lack of ability to be spatially variation to the input data
- Lot of input data required to work efficiently

2. **RNN (Recurrent Neural Network):**
    - Do not pass the information in one direction only
    - Each note works like a memory cell and continuously predict and learn
    - Backpropagation
    -

Advantage:

- Remember each and every information through time
- Useful in time series prediction
- This is also called as LSTM (Long Short Term Memory)
- Effective pixel neighborhood

Disadvantage:

- Gradient vanishing and exploding problems
- Training in RNN is very difficult task
- Can not process long sequences using Relu activation function

Difference of ANN VS CNN VS RNN:

ANN	CNN	RNN
Works with tabular and text data	Works with image data	Works with sequence data
Parameter sharing is not possible	Possible	Possible
Operate on Fixed Length Input	Operate on Fixed Length Input	Do not
Recurrent connection not possible	Recurrent not possible	Possible
Spatial relationship not possible	Possible	Not possible
Less power full compared to CNN and RNN	More powerful then ANN and RNN	Less powerful than CNN
Having fault tolerance and ability to work with incomplete knowledge	High accuracy in image recognition problems and it offers input sharing	Remembers each and every information and offers time series prediction

As we are working one tabular dataset in this paper will be working on Artificial Neural Network Model.

Review of Previous Researches:

Before doing this analysis I have studied 3 different papers. Here are the following limitations I found that is why I think this kind of project could add value.

- No body used ANN (MLP) Classifier where i think MLP can produce better result
- Lack of real dataset most of the paper written on the dummy and incomplete dataset. I received permission on the real company dataset which could be very helpful for the Banks and NBFIs for future research also.
- Seems most authors did very few RnD on this sector before doing analysis.

Major Components of Neural Network:

1. Layers
2. Weights
3. Bais
4. Activation

Tuning process of Neural Network:

1. Number of Hidden layer
2. Activation Functions
3. Batch and Epoch
4. Optimization Algorithm
5. Learning Rate

Chapter - 03

Methods and Material/Methodology

Research methodology

Type of research design

This report is a Quantitative type of research. The study is performed based on the information extracted from the database by using SQL and Python.

Sources of data

To make the Report more meaningful and presentable, two sources of data and information have been used widely. They are primary Data and Secondary Data.

1. **Primary sources:** Primary sources of information are the database of TallyKhata (Progoti Systems Limited). Initially I communicated with the CEO of the company for the permission of the data.

2. **Secondary sources:** Secondary sources of information are the internet and books.

Data collection procedure

In order to prepare the report both primary and secondary data are needed.

1. **Primary Data:** Primary data is collected from the documents and records of the company called TallyKhata (Progoti Systems Limited). I receive the data set in the email after doing formal and official communication.

2. **Secondary Data:** Secondary data is collected from the internet and books.

Population and Sampling

Total 6000 TallyKhata MSME users were selected for this analysis. TallyKhata has close to 2k users who have disbursed loans and a total of 4k users whom they can give loans to as they are eligible for the loan as per the data analysis. And another 2K users selected randomly for validation and testing the accuracy.

Data Details : I have segmented total required data points into 3 different top level categories for easy understanding and analysis

- Personal Information
- Business Information
- App Statistics + Transaction Information

Data Analysis techniques

Data Analysis will be done by usings different data mining techniques.

1. **Model Selection:** As this is a classification problem whether a MSME is eligible for loan or not, I choose the Multi-Layer perceptron (MLP) which is an ANN classifier for solving this classification problem.
2. MLP results with KNN, Decision Tree, Naive Bayes to find the best fit model.
3. More than 7- features initially selected and features will be optimized using statistical methods like - correlation among the dependent and independent variables.
4. The Artificial Neural Network Model will be used for solving this classification problem. Number of Hidden layers, weight initialization technique details, Epoch, Optimization techniques, Accuracy measurements all of these analyses will be done by usings different statistical methods.

5. Total dataset will be splitted as follows for training and testing-

   Train Data set: 6K TallyKhata user dataset, Prefer users whom Banks, MFI Or NGO's already disbursed loan. 4k or 70% data used for training.

   Test Dataset: 2K or 30% of the data used for ML validation.

6. Accuracy of the Model measured using the Confusion Matrix.

Chapter - 04

Analysis and Results

Analysis and Result:

Basic workFlow of a Machine Learning Model -

1. Data Collection
2. Data preparation/processing/feature engineering
3. Split data into training and test data set
4. Choose a Model
5. Training data to ML Model
6. Model Validation
7. Deployment

1. **Data Collection:**

   As we already explained the data collection details in the **methodology part.** By keeping the problem statement in mind that - Developing ML model for the assessment of creditwortheness of the MSMEs the data set that we collected is the structure dataset of a startup company called **TallyKhata**.

1.1 Data Structure : Data appears in tabulated format (rows and columns style, like what you'd find in an Excel spreadsheet). It contains different types of data like - numerical, categorical, time series.

Numerical: Days with platform, Total Active days, Recorded Transaction, Total Recorded Value (Sales), Business Age etc.

Categorical: Business Type, Device Brand, division, district, upazila etc.

Time series data: Last 7 days sales, Last 30 days sales, Last 90 days sales etc.

2. Data Preparation/Processing/Feature Engineering:

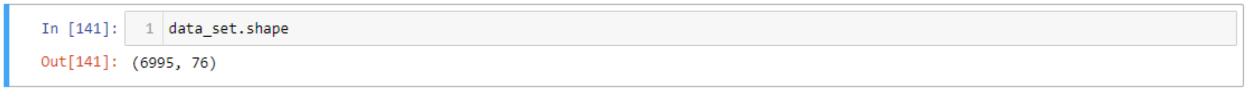
After receiving the dataset at first we do some exploratory data analysis (EDA). EDA is a technique to understand various aspects of data. This help us to understand flowing:

- Feature Variables and Target Variable
- Number of Null or Missing values in the dataset
- Handling the Null or Missing values using feature imputation
- Existence of outliers in the dataset or not
- Relationship between the variables
- Understand the faulty points in the data

So the objective of EDA is to clean the dataset before moving to more complex machine learning processes.

EDA:

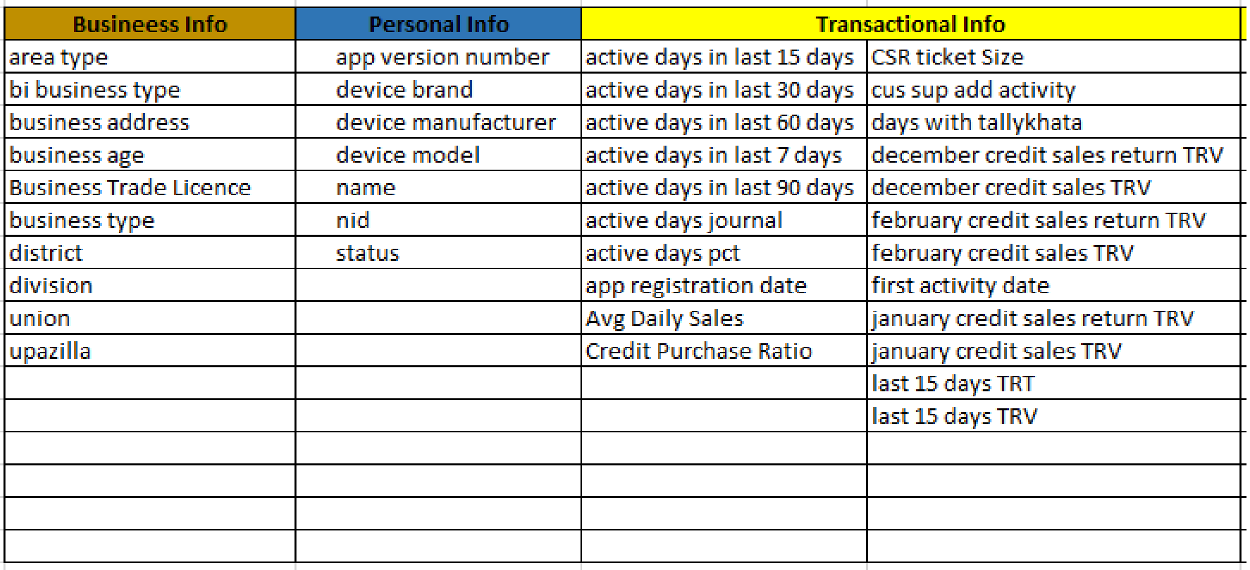
- **Identify Shape of the dataset**



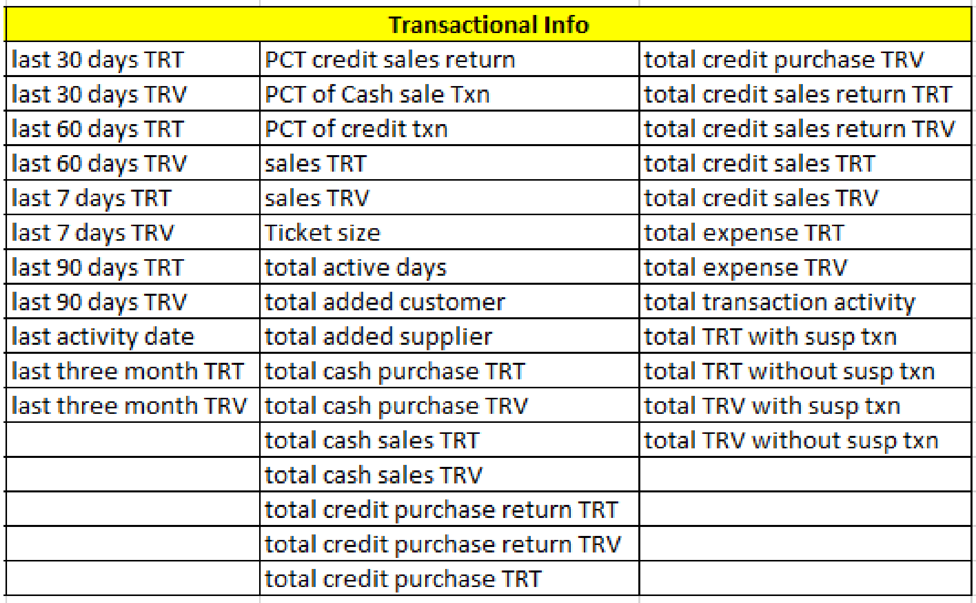
```
In [141]:  data_set.shape
Out[141]: (6995, 76)
```

Here from the output we see that there are 6995 records with 76features/attributes in my dataset.

- **Initial attributes and features**



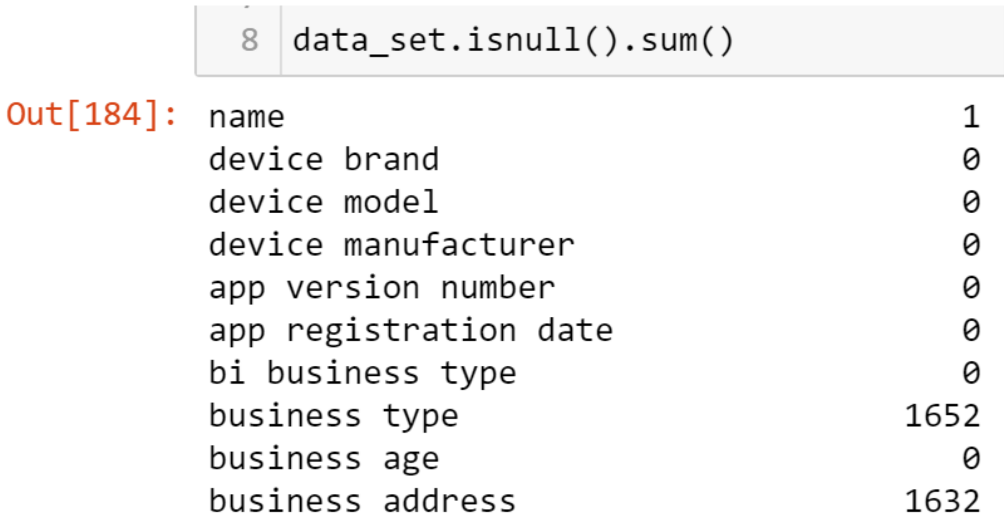
Busineess Info	Personal Info	Transactional Info	
area type	app version number	active days in last 15 days	CSR ticket Size
bi business type	device brand	active days in last 30 days	cus sup add activity
business address	device manufacturer	active days in last 60 days	days with tallykhata
business age	device model	active days in last 7 days	december credit sales return TRV
Business Trade Licence	name	active days in last 90 days	december credit sales TRV
business type	nid	active days journal	february credit sales return TRV
district	status	active days pct	february credit sales TRV
division		app registration date	first activity date
union		Avg Daily Sales	january credit sales return TRV
upazilla		Credit Purchase Ratio	january credit sales TRV
			last 15 days TRT
			last 15 days TRV



Transactional Info		
last 30 days TRT	PCT credit sales return	total credit purchase TRV
last 30 days TRV	PCT of Cash sale Txn	total credit sales return TRT
last 60 days TRT	PCT of credit txn	total credit sales return TRV
last 60 days TRV	sales TRT	total credit sales TRT
last 7 days TRT	sales TRV	total credit sales TRV
last 7 days TRV	Ticket size	total expense TRT
last 90 days TRT	total active days	total expense TRV
last 90 days TRV	total added customer	total transaction activity
last activity date	total added supplier	total TRT with susp txn
last three month TRT	total cash purchase TRT	total TRT without susp txn
last three month TRV	total cash purchase TRV	total TRV with susp txn
	total cash sales TRT	total TRV without susp txn
	total cash sales TRV	
	total credit purchase return TRT	
	total credit purchase return TRV	
	total credit purchase TRT	

Hence, now we have 3 datetime variables, 59 integer variables, 13 categorical or text variables and 1 float variable.

- **Identification of Null or Missing Values**



```
data_set.isnull().sum()
```

```
Out[184]: name                          1
          device brand                  0
          device model                  0
          device manufacturer           0
          app version number            0
          app registration date         0
          bi business type              0
          business type              1652
          business age                  0
          business address           1632
```

- **Handling the Missing Values and cleaning redundant and unnecessary columns :**

```
data_set.drop('app registration date', inplace=True, axis=1)

data_set.drop('first activity date', inplace=True, axis=1)

data_set.drop('last activity date', inplace=True, axis=1)

data_set.drop('app version number', inplace=True, axis=1)

data_set.drop('business type', inplace=True, axis=1)

data_set.drop('business address', inplace=True, axis=1)

data_set.drop('device brand', inplace=True, axis=1)

data_set.drop('device model', inplace=True, axis=1)

data_set.drop('division', inplace=True, axis=1)

data_set.drop('district', inplace=True, axis=1)

data_set.drop('upazilla', inplace=True, axis=1)

data_set.drop('union', inplace=True, axis=1)

data_set.drop('name', inplace=True, axis=1)
```

- **Deleting Records with ‘Null’ Values**

```
data_set_wn = data_set.dropna()

data_set_wn.isnull().sum()
```

Note:

- Here we dropped the both missing value columns. Because we have separate business location attributes and BI Business Type attributes that fulfill these requirements. 1 NULL value record removed from the name column.

Now after cleaning the redundant and unnecessary attributes and handling the null values the shape of my data is stands as -

```
dtypes: float64(1), int64(59), object(4)

memory usage: 3.4+ MB
```

- 1 Float attribute
- 59 Integer attribute
- 4 categorical attributes/variables

Now total feature reduces from 71 to 64

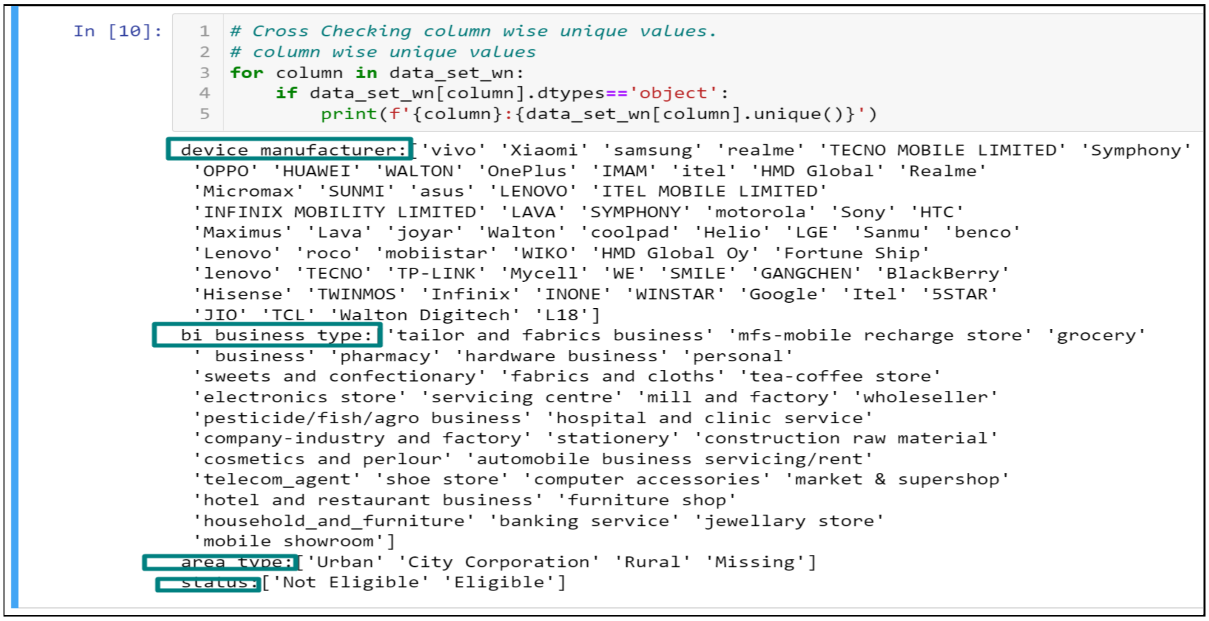
- **Handling the Categorical Variables**

We need to convert these categorical variables to numbers such that the model is able to understand and extract valuable information.So following ways we handled the categorical features -

List of Categorical Variables and type of encoding used:

- Device manufacturer : Ordinal Encoding Used

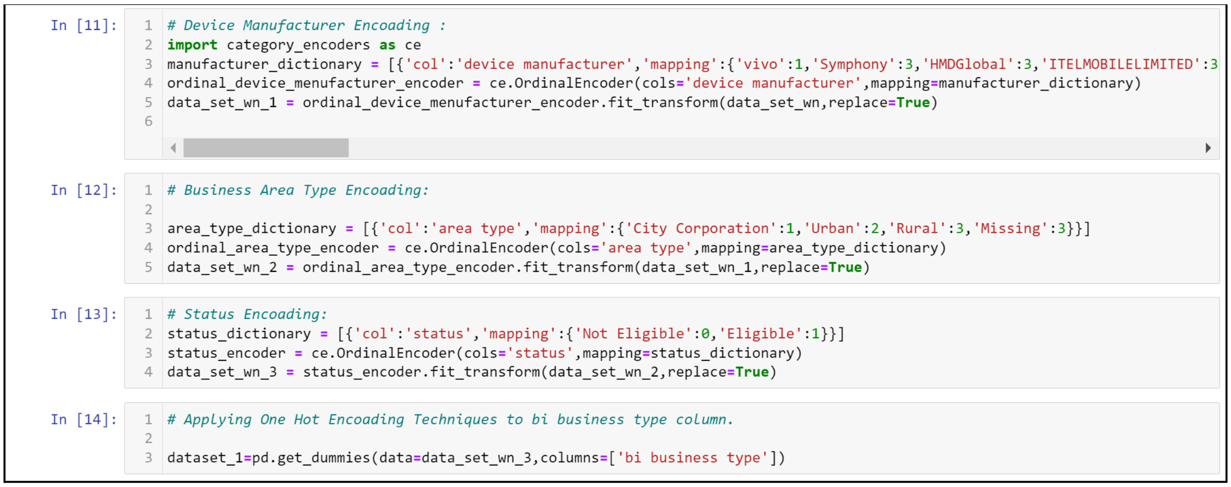
- Bi Business Type : One Hot Encoded Encoding Used
- Area Type : Ordinal Encoding Used
- Status : Normally replaced : **Eligible** - 1 and **Not Eligible** - 0



In [10]:
```
# Cross Checking column wise unique values.
# column wise unique values
for column in data_set_wn:
    if data_set_wn[column].dtypes=='object':
        print(f'{column}:{data_set_wn[column].unique()}')
```

```
device manufacturer:['vivo' 'Xiaomi' 'samsung' 'realme' 'TECNO MOBILE LIMITED' 'Symphony'
 'OPPO' 'HUAWEI' 'WALTON' 'OnePlus' 'IMAM' 'itel' 'HMD Global' 'Realme'
 'Micromax' 'SUNMI' 'asus' 'LENOVO' 'ITEL MOBILE LIMITED'
 'INFINIX MOBILITY LIMITED' 'LAVA' 'SYMPHONY' 'motorola' 'Sony' 'HTC'
 'Maximus' 'Lava' 'joyar' 'Walton' 'coolpad' 'Helio' 'LGE' 'Sanmu' 'benco'
 'Lenovo' 'roco' 'mobiistar' 'WIKO' 'HMD Global Oy' 'Fortune Ship'
 'lenovo' 'TECNO' 'TP-LINK' 'Mycell' 'WE' 'SMILE' 'GANGCHEN' 'BlackBerry'
 'Hisense' 'TWINMOS' 'Infinix' 'INONE' 'WINSTAR' 'Google' 'Itel' '5STAR'
 'JIO' 'TCL' 'Walton Digitech' 'L18']
bi business type:['tailor and fabrics business' 'mfs-mobile recharge store' 'grocery'
 ' business' 'pharmacy' 'hardware business' 'personal'
 'sweets and confectionary' 'fabrics and cloths' 'tea-coffee store'
 'electronics store' 'servicing centre' 'mill and factory' 'wholeseller'
 'pesticide/fish/agro business' 'hospital and clinic service'
 'company-industry and factory' 'stationery' 'construction raw material'
 'cosmetics and perlour' 'automobile business servicing/rent'
 'telecom_agent' 'shoe store' 'computer accessories' 'market & supershop'
 'hotel and restaurant business' 'furniture shop'
 'household_and_furniture' 'banking service' 'jewellary store'
 'mobile showroom']
area type:['Urban' 'City Corporation' 'Rural' 'Missing']
status:['Not Eligible' 'Eligible']
```

After applying necessary encoding techniques on the dataset now we see that all the data types are numerical.



In [11]:
```
# Device Manufacturer Encoading :
import category_encoders as ce
manufacturer_dictionary = [{'col':'device manufacturer','mapping':{'vivo':1,'Symphony':3,'HMDGlobal':3,'ITELMOBILELIMITED':3
ordinal_device_menufacturer_encoder = ce.OrdinalEncoder(cols='device manufacturer',mapping=manufacturer_dictionary)
data_set_wn_1 = ordinal_device_menufacturer_encoder.fit_transform(data_set_wn,replace=True)

```

In [12]:
```
# Business Area Type Encoading:

area_type_dictionary = [{'col':'area type','mapping':{'City Corporation':1,'Urban':2,'Rural':3,'Missing':3}}]
ordinal_area_type_encoder = ce.OrdinalEncoder(cols='area type',mapping=area_type_dictionary)
data_set_wn_2 = ordinal_area_type_encoder.fit_transform(data_set_wn_1,replace=True)
```

In [13]:
```
# Status Encoading:
status_dictionary = [{'col':'status','mapping':{'Not Eligible':0,'Eligible':1}}]
status_encoder = ce.OrdinalEncoder(cols='status',mapping=status_dictionary)
data_set_wn_3 = status_encoder.fit_transform(data_set_wn_2,replace=True)
```

In [14]:
```
# Applying One Hot Encoading Techniques to bi business type column.

dataset_1=pd.get_dummies(data=data_set_wn_3,columns=['bi business type'])
```

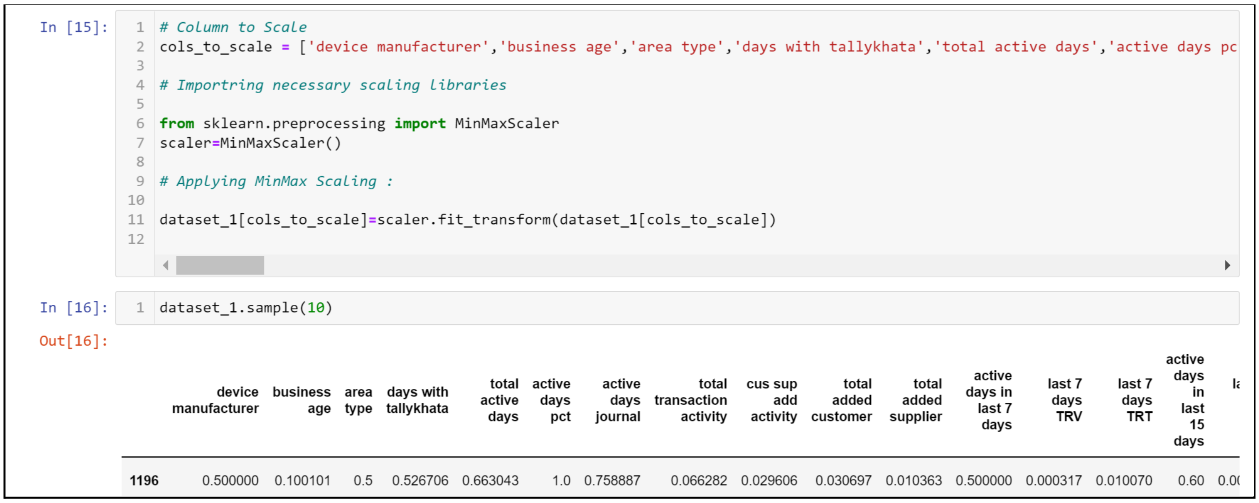
dtypes: float64(2), int32(2), int64(59), uint8(103)

memory usage: 4.0 MB

Note: Because of the application of one Hot Encoding we see that the number of columns increased.

- Standardization of Dataset

For the standardization we used Min Max Scaling Techniques.



In [15]:
```
# Column to Scale
cols_to_scale = ['device manufacturer','business age','area type','days with tallykhata','total active days','active days pc

# Importring necessary scaling libraries

from sklearn.preprocessing import MinMaxScaler
scaler=MinMaxScaler()

# Applying MinMax Scaling :

dataset_1[cols_to_scale]=scaler.fit_transform(dataset_1[cols_to_scale])

```

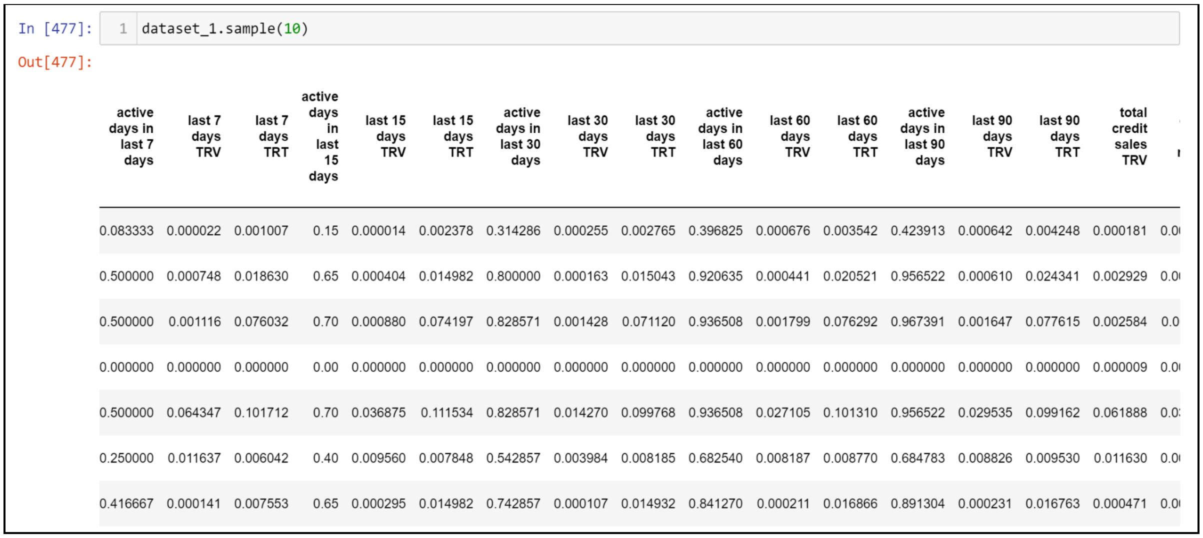
In [16]:
```
dataset_1.sample(10)
```

Out[16]:

	device manufacturer	business age	area type	days with tallykhata	total active days	active days pct	active days journal	total transaction activity	cus sup add activity	total added customer	total added supplier	active days in last 7 days	last 7 days TRV	last 7 days TRT	active days in last 15 days	la
1196	0.500000	0.100101	0.5	0.526706	0.663043	1.0	0.758887	0.066282	0.029606	0.030697	0.010363	0.500000	0.000317	0.010070	0.60	0.00

Data standardization helps improve the quality of your data by transforming and standardizing it. Think of it like a uniform for your databases. By taking this step, you are formatting your records in a way that creates consistency across your systems and makes it easy for businesses to use.

Here for the standardization we use the MinMaxScaler method.MinMaxScaler preserves the shape of the original distribution. It doesn't meaningfully change the information embedded in the original data. Note that MinMaxScaler doesn't reduce the importance of outliers. The default range for the feature returned by MinMaxScaler is 0 to 1.



In [477]:
```
dataset_1.sample(10)
```

Out[477]:

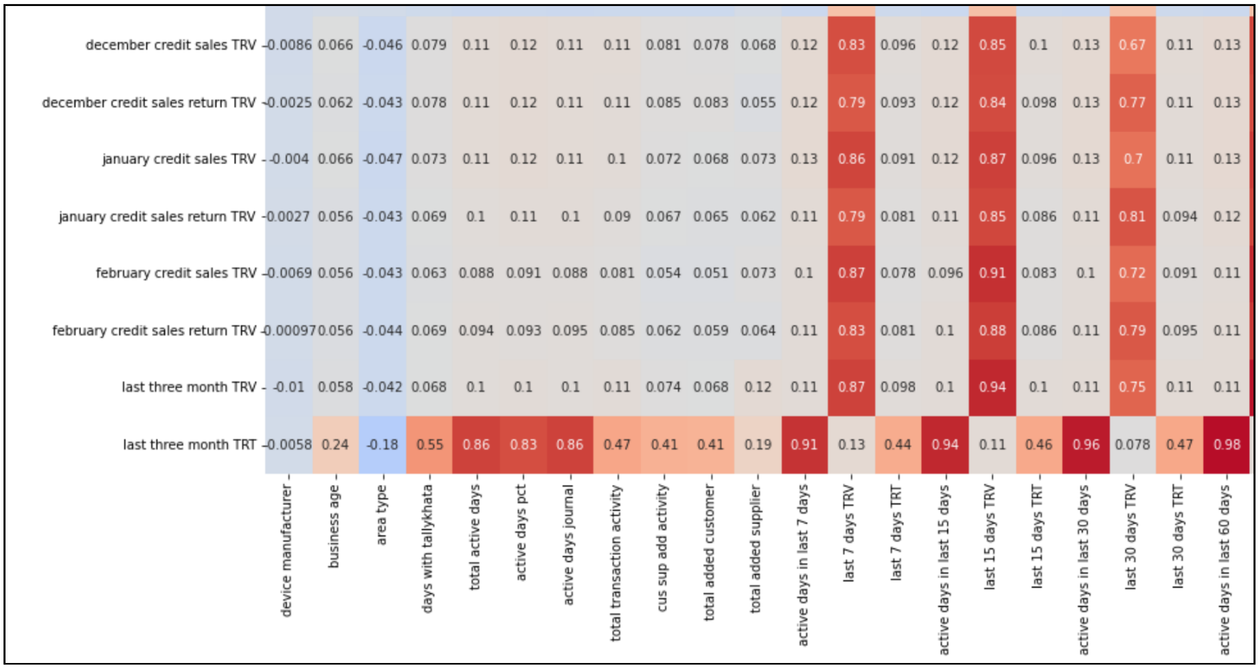
active days in last 7 days	last 7 days TRV	last 7 days TRT	active days in last 15 days	last 15 days TRV	last 15 days TRT	active days in last 30 days	last 30 days TRV	last 30 days TRT	active days in last 60 days	last 60 days TRV	last 60 days TRT	active days in last 90 days	last 90 days TRV	last 90 days TRT	total credit sales TRV	
0.083333	0.000022	0.001007	0.15	0.000014	0.002378	0.314286	0.000255	0.002765	0.396825	0.000676	0.003542	0.423913	0.000642	0.004248	0.000181	0.0
0.500000	0.000748	0.018630	0.65	0.000404	0.014982	0.800000	0.000163	0.015043	0.920635	0.000441	0.020521	0.956522	0.000610	0.024341	0.002929	0.0
0.500000	0.001116	0.076032	0.70	0.000880	0.074197	0.828571	0.001428	0.071120	0.936508	0.001799	0.076292	0.967391	0.001647	0.077615	0.002584	0.0
0.000000	0.000000	0.000000	0.00	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000009	0.0
0.500000	0.064347	0.101712	0.70	0.036875	0.111534	0.828571	0.014270	0.099768	0.936508	0.027105	0.101310	0.956522	0.029535	0.099162	0.061888	0.0
0.250000	0.011637	0.006042	0.40	0.009560	0.007848	0.542857	0.003984	0.008185	0.682540	0.008187	0.008770	0.684783	0.008826	0.009530	0.011630	0.0
0.416667	0.000141	0.007553	0.65	0.000295	0.014982	0.742857	0.000107	0.014932	0.841270	0.000211	0.016866	0.891304	0.000231	0.016763	0.000471	0.0

So after scaling the dataset this is the shape of the dataset now, we clearly see that all the attributes range converted to 0 to 1.

- Advance Feature Engineering

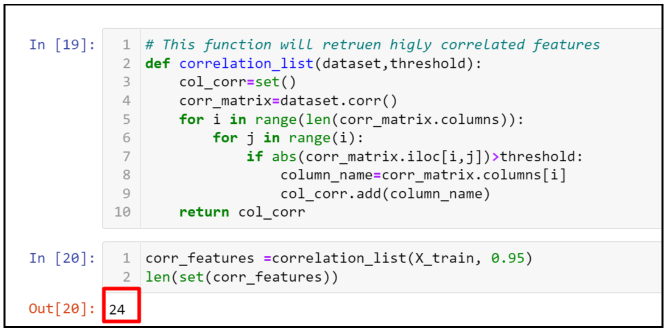
Here I analyzed the correlation between the independent variables and tried to figure out what list of variables is strongly correlated. These highly correlated attributes may cause multicollinearity in the dataset and overfitting.

Part of correlation Matrix between independent variables:



	device manufacturer	business age	area type	days with tallykhata	total active days	active days pct	active days journal	total transaction activity	cus sup add activity	total added customer	total added supplier	active days in last 7 days	last 7 days TRV	last 7 days TRT	active days in last 15 days	last 15 days TRV	last 15 days TRT	active days in last 30 days	last 30 days TRV	last 30 days TRT	active days in last 60 days
december credit sales TRV	-0.0086	0.066	-0.046	0.079	0.11	0.12	0.11	0.11	0.081	0.078	0.068	0.12	0.83	0.096	0.12	0.85	0.1	0.13	0.67	0.11	0.13
december credit sales return TRV	-0.0025	0.062	-0.043	0.078	0.11	0.12	0.11	0.11	0.085	0.083	0.055	0.12	0.79	0.093	0.12	0.84	0.098	0.13	0.77	0.11	0.13
january credit sales TRV	-0.004	0.066	-0.047	0.073	0.11	0.12	0.11	0.1	0.072	0.068	0.073	0.13	0.86	0.091	0.12	0.87	0.096	0.13	0.7	0.11	0.13
january credit sales return TRV	-0.0027	0.056	-0.043	0.069	0.1	0.11	0.1	0.09	0.067	0.065	0.062	0.11	0.79	0.081	0.11	0.85	0.086	0.11	0.81	0.094	0.12
february credit sales TRV	-0.0069	0.056	-0.043	0.063	0.088	0.091	0.088	0.081	0.054	0.051	0.073	0.1	0.87	0.078	0.096	0.91	0.083	0.1	0.72	0.091	0.11
february credit sales return TRV	-0.00097	0.056	-0.044	0.069	0.094	0.093	0.095	0.085	0.062	0.059	0.064	0.11	0.83	0.081	0.1	0.88	0.086	0.11	0.79	0.095	0.11
last three month TRV	-0.01	0.058	-0.042	0.068	0.1	0.1	0.1	0.11	0.074	0.068	0.12	0.11	0.87	0.098	0.1	0.94	0.1	0.11	0.75	0.11	0.11
last three month TRT	-0.0058	0.24	-0.18	0.55	0.86	0.83	0.86	0.47	0.41	0.41	0.19	0.91	0.13	0.44	0.94	0.11	0.46	0.96	0.078	0.47	0.98

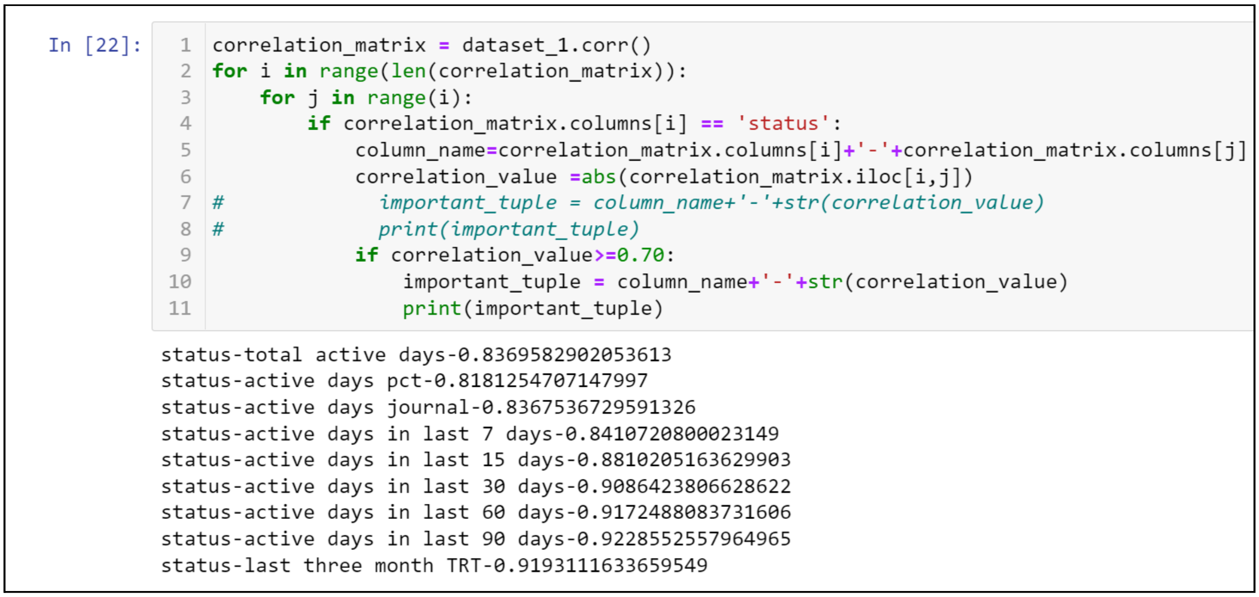
This a partial screenshot of correlation matrix, from here we see a lot of variables are highly correlated with each other, Hence there is a scope of eliminating the highly correlated features by keeping only one set.



```
In [19]:
# This function will retruen higly correlated features
def correlation_list(dataset,threshold):
    col_corr=set()
    corr_matrix=dataset.corr()
    for i in range(len(corr_matrix.columns)):
        for j in range(i):
            if abs(corr_matrix.iloc[i,j])>threshold:
                column_name=corr_matrix.columns[i]
                col_corr.add(column_name)
    return col_corr

In [20]:
corr_features =correlation_list(X_train, 0.95)
len(set(corr_features))

Out[20]: 24
```

After developing a customer python function and analysis I see there are 24 variables that are strongly correlated. With correlation value of abs() >=0.95

After that I tried to identify correlation between dependent and independent variables and here is the code and result screenshot of Dependent and Independent variable correlation with abs() >=0.70



```
In [22]:
correlation_matrix = dataset_1.corr()
for i in range(len(correlation_matrix)):
    for j in range(i):
        if correlation_matrix.columns[i] == 'status':
            column_name=correlation_matrix.columns[i]+'-'+correlation_matrix.columns[j]
            correlation_value =abs(correlation_matrix.iloc[i,j])
#             important_tuple = column_name+'-'+str(correlation_value)
#             print(important_tuple)
            if correlation_value>=0.70:
                important_tuple = column_name+'-'+str(correlation_value)
                print(important_tuple)
```

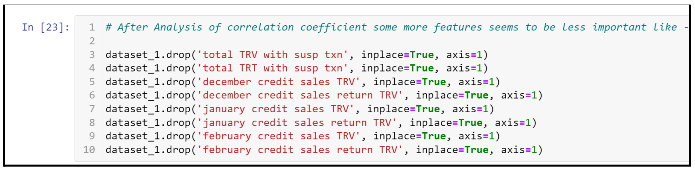
```
status-total active days-0.8369582902053613
status-active days pct-0.8181254707147997
status-active days journal-0.8367536729591326
status-active days in last 7 days-0.8410720800023149
status-active days in last 15 days-0.8810205163629903
status-active days in last 30 days-0.9086423806628622
status-active days in last 60 days-0.9172488083731606
status-active days in last 90 days-0.9228552557964965
status-last three month TRT-0.9193111633659549
```

Now we have two different scenarios,

- Correlation matrix among independent variables.
- Correlation Matrix between Dependent and Independent variables.

Except for the common variables we dropped highly correlated variables to avoid the overfitting and multicollinearity issue.

Excluded Variables:



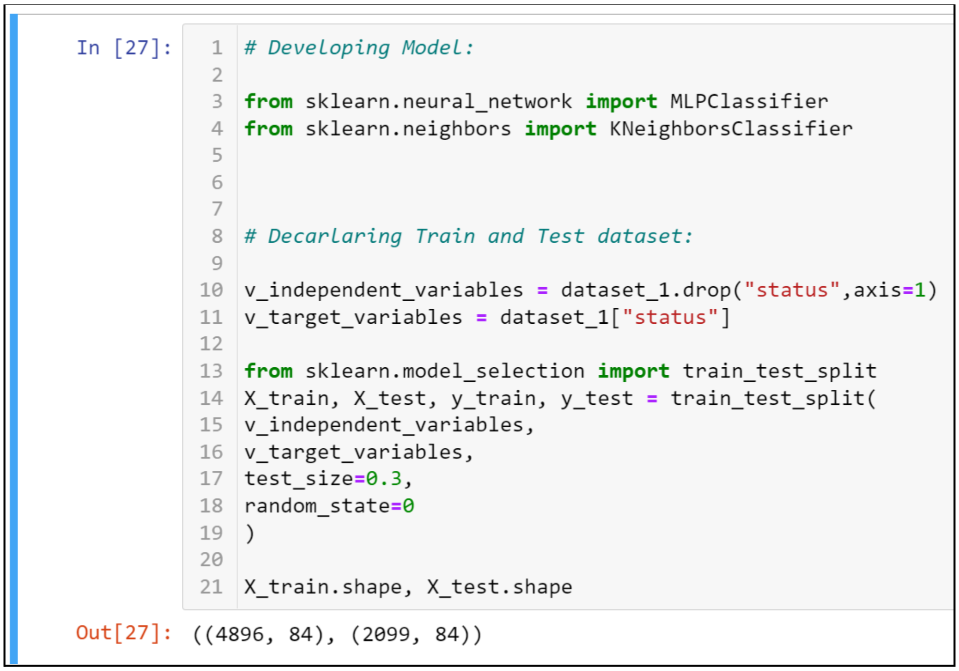
```
In [23]:
# After Analysis of correlation coefficient some more features seems to be less important like -

dataset_1.drop('total TRV with susp txn', inplace=True, axis=1)
dataset_1.drop('total TRT with susp txn', inplace=True, axis=1)
dataset_1.drop('december credit sales TRV', inplace=True, axis=1)
dataset_1.drop('december credit sales return TRV', inplace=True, axis=1)
dataset_1.drop('january credit sales TRV', inplace=True, axis=1)
dataset_1.drop('january credit sales return TRV', inplace=True, axis=1)
dataset_1.drop('february credit sales TRV', inplace=True, axis=1)
dataset_1.drop('february credit sales return TRV', inplace=True, axis=1)
```

3. Splitting Data set into train and test dataset

Data set is splitted into two parts.

1. Independent Variables - All the variables except the ‘Status’ Column.
2. Dependent Variables - Status Column.

And then by using the sklearn library 30% considered as test data and 70% considered as training data. Random State setted as =0.



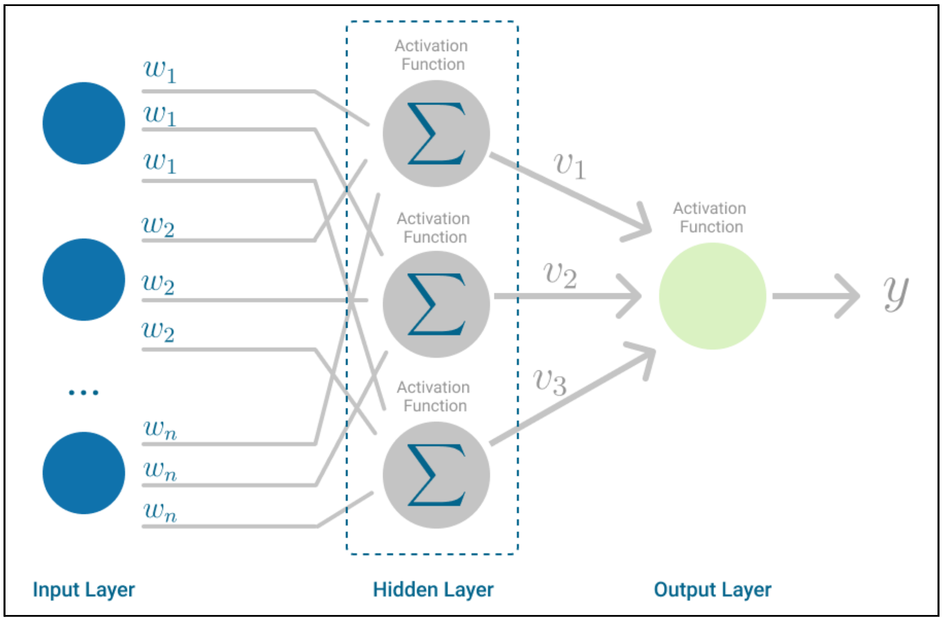
```
In [27]:
# Developing Model:

from sklearn.neural_network import MLPClassifier
from sklearn.neighbors import KNeighborsClassifier



# Decarlaring Train and Test dataset:

v_independent_variables = dataset_1.drop("status",axis=1)
v_target_variables = dataset_1["status"]

from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(
v_independent_variables,
v_target_variables,
test_size=0.3,
random_state=0
)

X_train.shape, X_test.shape
```

```
Out[27]: ((4896, 84), (2099, 84))
```

4. Choose a Model

- Initially for the An ANN model (MLP) classifier selected to solve this binary classification problem.
- Then KNN, Decision Tree and Naive Bayes Classifier also applied on the same dataset.
- After that Multi-Layer perceptron (MLP) Classification model is tuned to get the maximum output

MLPClassifier stands for **Multi-layer Perceptron classifier** which in the name itself connects to a Neural Network. Unlike other classification algorithms such as Support Vectors or Naive Bayes Classifier, MLPClassifier relies on an underlying Neural Network to perform the task of classification.



`MLPClassifier` default parameter used:

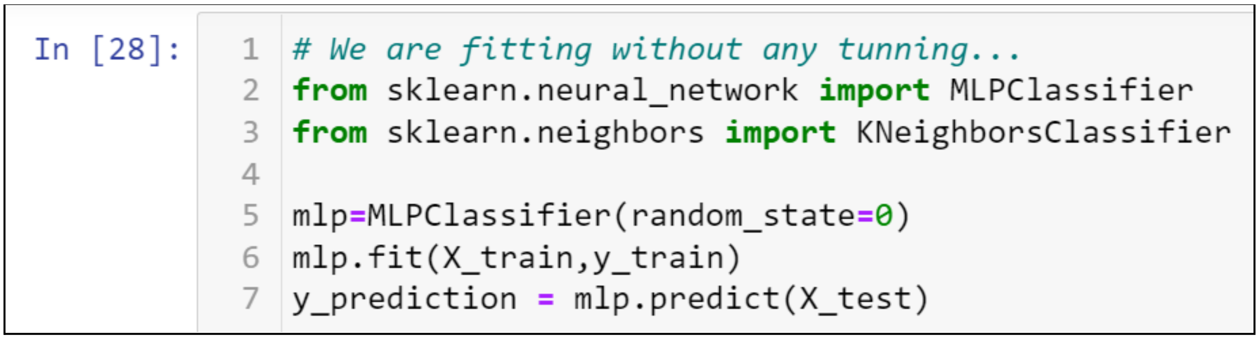
- hidden_layer_sizes=(100,) - Default

  default(100,) means **if no value is provided for hidden_layer_sizes then default architecture will have one input layer, one hidden layer with 100 units and one output layer**.

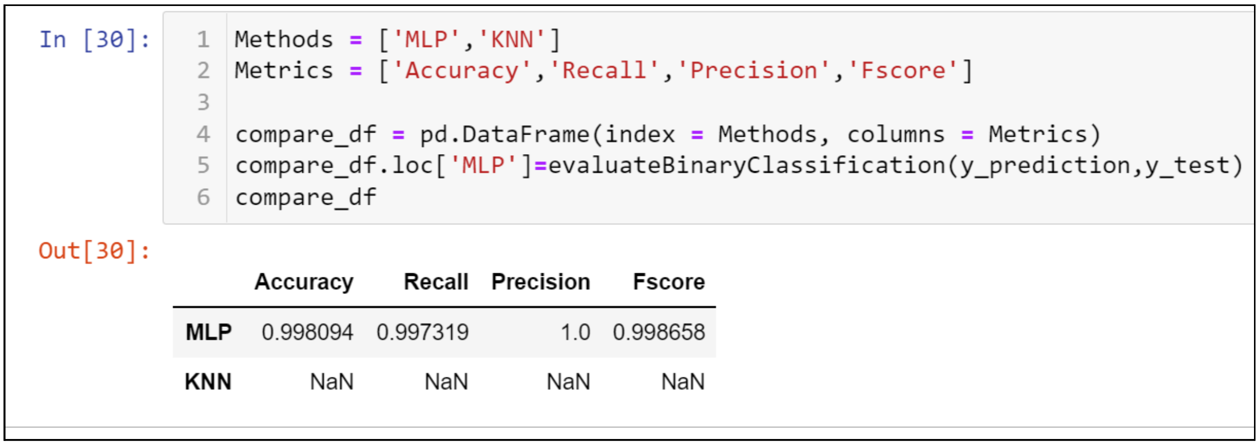
- Activation='relu'
- solver='adam'
- learning_rate_init=0.001

5. Training data to ML Model

Initially By using these parameters the model is trained with MLP classifiers. Hence we were able to product result as:



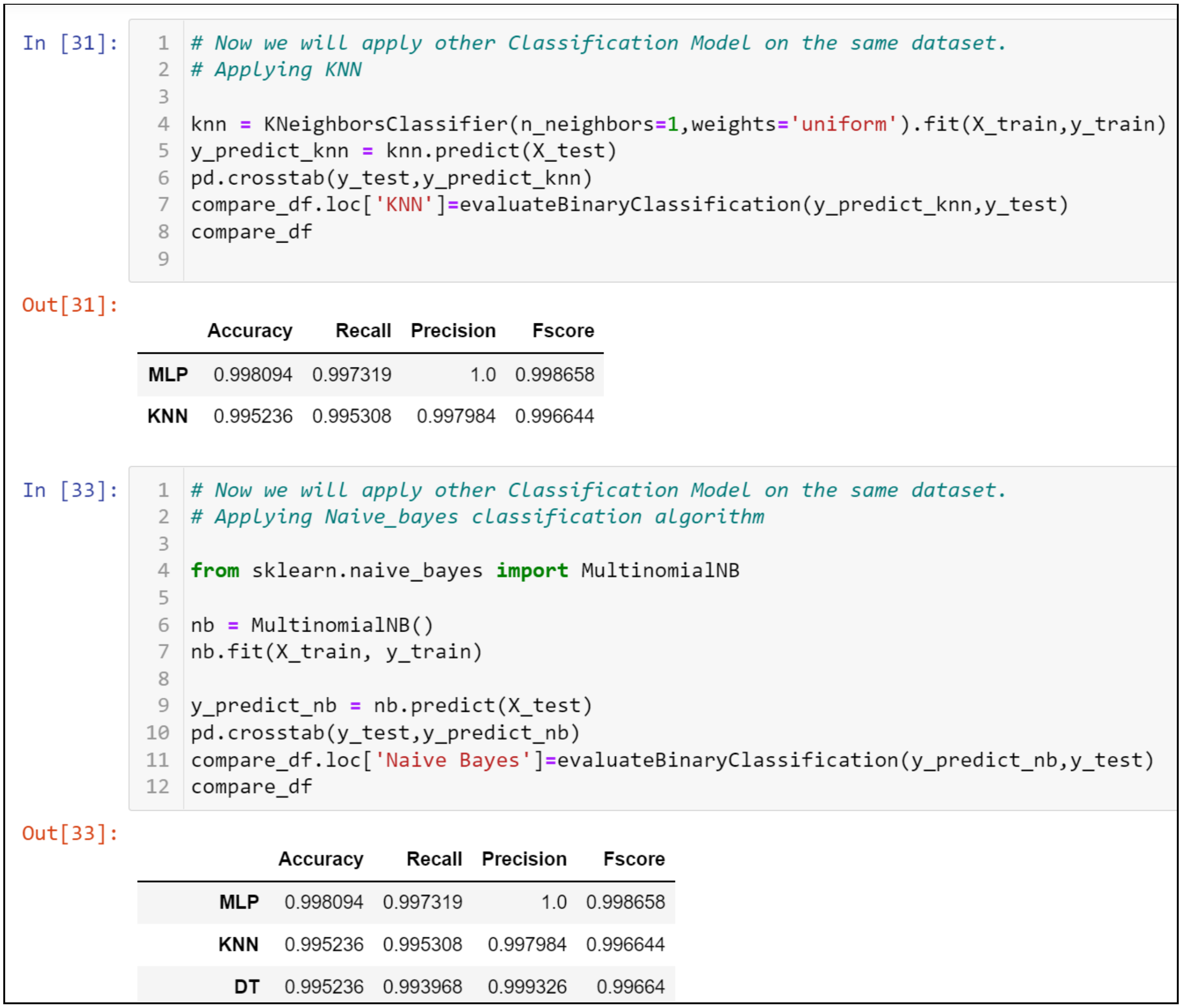
```
In [28]:
# We are fitting without any tunning...
from sklearn.neural_network import MLPClassifier
from sklearn.neighbors import KNeighborsClassifier

mlp=MLPClassifier(random_state=0)
mlp.fit(X_train,y_train)
y_prediction = mlp.predict(X_test)
```



```
In [30]:
Methods = ['MLP','KNN']
Metrics = ['Accuracy','Recall','Precision','Fscore']

compare_df = pd.DataFrame(index = Methods, columns = Metrics)
compare_df.loc['MLP']=evaluateBinaryClassification(y_prediction,y_test)
compare_df
```

```
Out[30]:
        Accuracy    Recall  Precision    Fscore
MLP     0.998094  0.997319        1.0  0.998658
KNN          NaN       NaN        NaN       NaN
```

- 0.9980 Accuracy (Seems default parameters of the algorithms gives us best result, Laiter on comparison with other algorithm and optimized MLP algorithm will say weather this is best or not)



In [31]:

```python
# Now we will apply other Classification Model on the same dataset.
# Applying KNN

knn = KNeighborsClassifier(n_neighbors=1,weights='uniform').fit(X_train,y_train)
y_predict_knn = knn.predict(X_test)
pd.crosstab(y_test,y_predict_knn)
compare_df.loc['KNN']=evaluateBinaryClassification(y_predict_knn,y_test)
compare_df

```

Out[31]:

	Accuracy	Recall	Precision	Fscore
MLP	0.998094	0.997319	1.0	0.998658
KNN	0.995236	0.995308	0.997984	0.996644

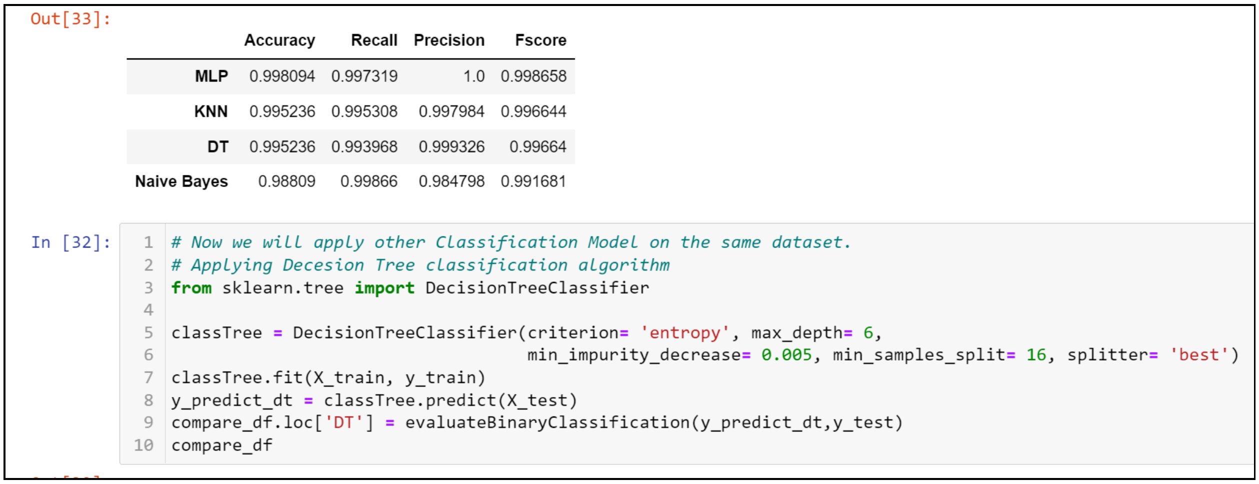
In [33]:

```python
# Now we will apply other Classification Model on the same dataset.
# Applying Naive_bayes classification algorithm

from sklearn.naive_bayes import MultinomialNB

nb = MultinomialNB()
nb.fit(X_train, y_train)

y_predict_nb = nb.predict(X_test)
pd.crosstab(y_test,y_predict_nb)
compare_df.loc['Naive Bayes']=evaluateBinaryClassification(y_predict_nb,y_test)
compare_df
```

Out[33]:

	Accuracy	Recall	Precision	Fscore
MLP	0.998094	0.997319	1.0	0.998658
KNN	0.995236	0.995308	0.997984	0.996644
DT	0.995236	0.993968	0.999326	0.99664

KNN and Naive_bayes algorithm Used on the same dataset and here we see that compared with MLP (Multi-Layer Perceptron) Classifiers these two give us less accuracy.



Out[33]:

	Accuracy	Recall	Precision	Fscore
MLP	0.998094	0.997319	1.0	0.998658
KNN	0.995236	0.995308	0.997984	0.996644
DT	0.995236	0.993968	0.999326	0.99664
Naive Bayes	0.98809	0.99866	0.984798	0.991681

In [32]:

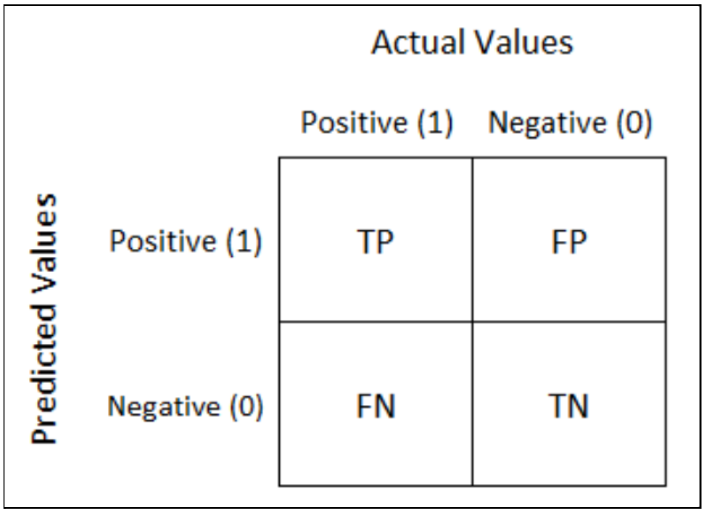
```python
# Now we will apply other Classification Model on the same dataset.
# Applying Decesion Tree classification algorithm
from sklearn.tree import DecisionTreeClassifier

classTree = DecisionTreeClassifier(criterion= 'entropy', max_depth= 6,
                                   min_impurity_decrease= 0.005, min_samples_split= 16, splitter= 'best')
classTree.fit(X_train, y_train)
y_predict_dt = classTree.predict(X_test)
compare_df.loc['DT'] = evaluateBinaryClassification(y_predict_dt,y_test)
compare_df
```

Hence, the decision tree applied on the dataset and found that same data set.

** MLP classifier gives us highest accuracy compared to other classification algorithms like- KNN, Naive Bayes, Decision Tree etc.

6. Measuring Accuracy

Accuracy simply measures how often the classifier correctly predicts. We can define accuracy as the ratio of the number of correct predictions and the total number of predictions.



		Actual Values	
		Positive (1)	Negative (0)
Predicted Values	Positive (1)	TP	FP
	Negative (0)	FN	TN

True Positive: We predicted positive and it’s true. In the image, we predicted that a woman is pregnant and she actually is.

True Negative: We predicted negative and it’s true. In the image, we predicted that a man is not pregnant and he actually is not.

False Positive (Type 1 Error)- We predicted positive and it’s false. In the image, we predicted that a man is pregnant but he actually is not.

False Negative (Type 2 Error)- We predicted negative and it’s false. In the image, we predicted that a woman is not pregnant but she actually is.

We discussed Accuracy, now let’s discuss some other metrics of the confusion matrix

1. Precision—Precision explains how many of the correctly predicted cases actually turned out to be positive. Precision is useful in the cases where False Positive is a higher concern than False Negatives. The importance of Precision is in music or video recommendation systems, e-commerce websites, etc. where wrong results could lead to customer churn and this could be harmful to the business.

Precision for a label is defined as the number of true positives divided by the number of predicted positives.

$$Precision = \frac{TruePositive}{TruePositive + FalsePositive}$$

2. Recall (Sensitivity)—Recall explains how many of the actual positive cases we were able to predict correctly with our model. It is a useful metric in cases where False Negative is of higher concern than False Positive. It is important in medical cases where it doesn’t matter whether we raise a false alarm but the actual positive cases should not go undetected!

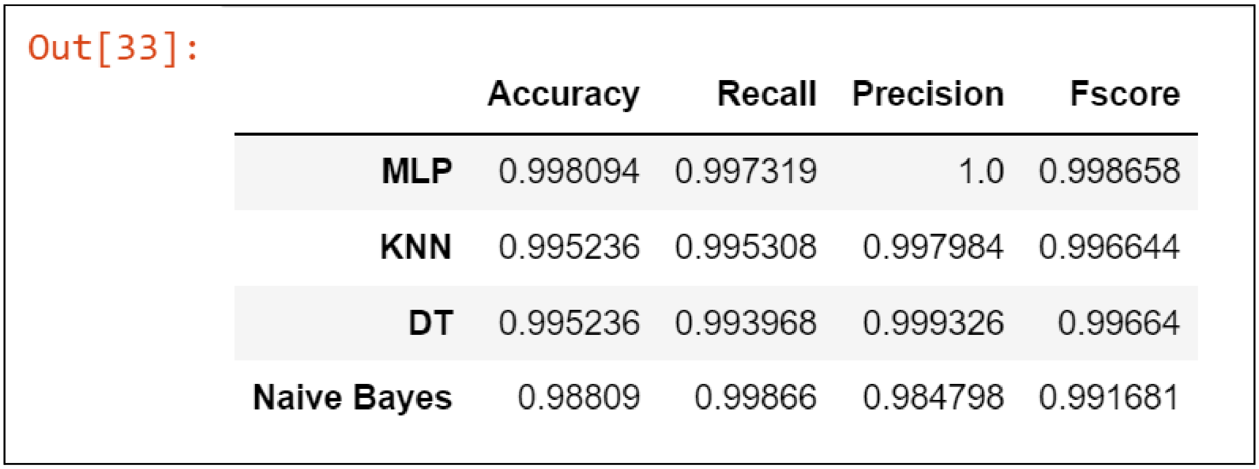
Recall for a label is defined as the number of true positives divided by the total number of actual positives.

$$Recall = \frac{TruePositive}{TruePositive + FalseNegative}$$

3. F1 Score—It gives a combined idea about Precision and Recall metrics. It is maximum when Precision is equal to Recall.

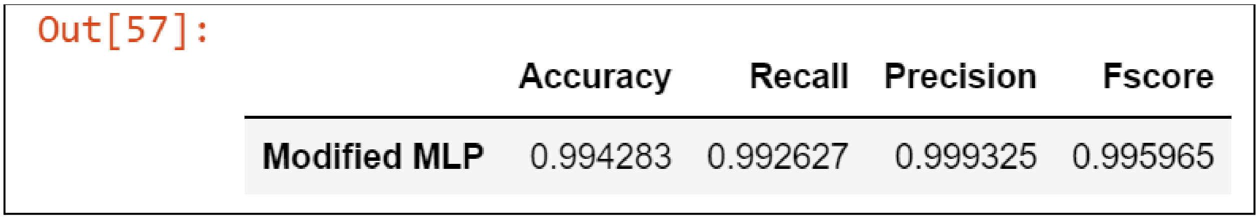
F1 Score is the harmonic mean of precision and recall.

$$F1 = 2.\frac{Precision \ \times Recall}{Precision + Recall}$$



Out[33]:

	Accuracy	Recall	Precision	Fscore
MLP	0.998094	0.997319	1.0	0.998658
KNN	0.995236	0.995308	0.997984	0.996644
DT	0.995236	0.993968	0.999326	0.99664
Naive Bayes	0.98809	0.99866	0.984798	0.991681



Out[57]:

	Accuracy	Recall	Precision	Fscore
Modified MLP	0.994283	0.992627	0.999325	0.995965

Here I represent the accuracy of the MLP classifier, K-nearest Neighbor, Decision Tree, Naive Bayes and finally Optimized MLP.

Chapter - 05

Findings

Core Findings:

After doing analysis on some reports and papers and several articles and doing analysis on the collected data set I have come up with following findings -

- By Usings MLP Classifier we are able to detect the eligibility of credit with accuracy 99.80%

- Compared to Other Model KNN, Decision Tree, Naive Bayes, MLP can perform better at detecting the creditworthiness

- MLP performs better with default parameters like -Relu activation in the hidden layer and logistics function in binary classification output layer

- Poor quality of collateral, Inadequate formal documentation and Ill-defined business plans are the core challenges of getting loan in MSME sector

Chapter - 05

Recommendations

Recommendations:

- Here in this report we are able to figure out that if we have digital data available we are still able to detect the credit eligibility with the accuracy level of 99%. This may derive new ways of distributing the loan/ digital loan in this sector.

- One of the important causes of high interest rates is Loan distribution cost. Because if a bank recruits several employees, spending an amount on visiting the MSMEs shop multiple times and collecting traditional documents is very expensive.

- As our Banks and NBFIs primary target is the credit return capability, hence their data can be an asset instead of imposed high value collateral. Data Analysis and ML can help the Bank and NBFIs to judge the person.

- Government should also focus on the potentiality of usings ANN in this sector rather than imposing the high collateral assert.

Chapter - 05

Conclusions

Conclusion

In conclusion I can say that this project report can help our Banks, NBFIs and NGOs who continuously distribute the loand the MSME sector. As technology grasps the generation day by day we need to cope with this using different data science techniques. A sector with 25% of contribution in the GDP like MSME , we can leverage them really by assessing the digital using ANN and other ML/AI and statistical methods.

Chapter - 06

References

References:

1. https://ieeexplore.ieee.org/document/6567409/authors#authors
    - Fast Neural Network Algorithm for solving the Classification Problem.
2. https://www.youtube.com/watch?v=Aq-F0ADkHGI
    - Use MLP in a specific way to be able to fit for classification problems.
3. https://analyticsindiamag.com/a-beginners-guide-to-scikit-learns-mlpclassifier/ (Example of MLP Classification)
4. https://www.datasciencecentral.com/credit-risk-prediction-using-artificial-neural-network-algorithm/
5. https://www.thedailystar.net/business/news/the-sme-loan-conundrum-1715272
6. https://www.bb.org.bd/sme/smepolicye.pdf
7. https://documents1.worldbank.org/curated/en/995331545025954781/Financing-Solutions-for-Micro-Small-and-Medium-Enterprises-in-Bangladesh.pdf
8. https://www.adb.org/sites/default/files/publication/214476/adbi-smes-developing-asia.pdf
9. https://documents1.worldbank.org/curated/en/995331545025954781/Financing-Solutions-for-Micro-Small-and-Medium-Enterprises-in-Bangladesh.pdf
10. https://www.unescap.org/sites/default/d8files/knowledge-products/MSME%20financing%20Bangladesh_10%20May%202021_share_0.pdf
11. http://www.diva-portal.org/smash/get/diva2:829365/FULLTEXT01.pdf (ANN Scoring)
12. https://www.datasciencecentral.com/credit-risk-prediction-using-artificial-neural-network-algorithm/ (ANN Credit Risk)
13. https://www.mecs-press.org/ijmecs/ijmecs-v10-n5/IJMECS-V10-N5-2.pdf (ANN in Credit risk prediction)
14. https://ieeexplore.ieee.org/document/6567409/authors#authors (Classification Algorithm ANN)

Chapter - 08

Appendix

Dataset + Python script

https://drive.google.com/drive/folders/1DUYAMwzyMHDtO9aGdz10_UOP29vgNJxo?usp=sharing

Anybody can use these resources for further research and analysis.