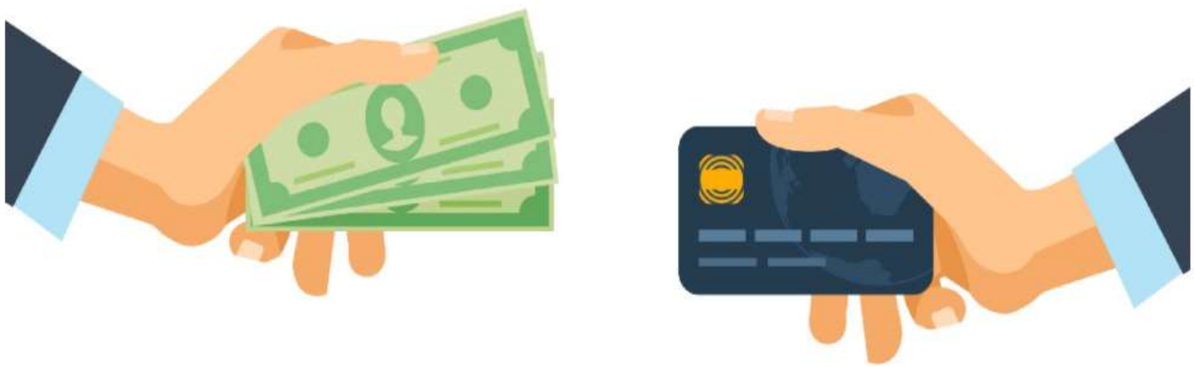


MITA Capstone project
Credit card defaulter analysis
Project by: Nupoor Joshi

Under the guidance of Prof Michail Xyntarakis.

Background
Credit Card Defaults have become much common than before these days.
Credit card defaults occur when one becomes severely neglectful about credit card payments. Defaults can not only harm credit card ratings within a bank but can impact their overall credit scores.
To address the issue of potential default, prediction of credit card defaults is essential. The predictive analysis can help prevent financial losses to the credit card companies. It can also help in being proactive by providing financial advice to customers that are more likely to default.

Goal
To predict whether the customer will default or not by carrying out an analysis of the Taiwanese dataset.

Software used

- **OS**: Windows 10
- **Programming language:** Python
- **Environment**: Anaconda Navigator
- **Libraries:** Scikit-learn, NumPy, pandas, matplotlib, Plotly

Dataset and Data description
About the dataset
The dataset consists of credit card payments, demographic factors, history of the credit card user in Taiwan from April 2005 to September 2005.
The total number of rows are 30000.
Features:
There are 25 features:
ID: ID of each client
LIMIT_BAL: Amount of given credit in NT dollars (includes individual and family/supplementary credit
SEX: Gender (1=male, 2=female)
EDUCATION: (1=graduate school, 2=university, 3=high school, 4=others, 5=unknown, 6=unknown)
MARRIAGE: Marital status (1=married, 2=single, 3=others)
AGE: Age in years
PAY_0: Repayment status in September, 2005 (-1=pay duly, 1=payment delay for one month, 2=payment delay for two months, ... 8=payment delay for eight months, 9=payment delay for nine months and above)
PAY_2: Repayment status in August, 2005 (scale same as above)
PAY_3: Repayment status in July, 2005 (scale same as above)
PAY_4: Repayment status in June, 2005 (scale same as above)
PAY_5: Repayment status in May, 2005 (scale same as above)
PAY_6: Repayment status in April, 2005 (scale same as above)
BILL_AMT1: Amount of bill statement in September, 2005 (NT dollar)
BILL_AMT2: Amount of bill statement in August, 2005 (NT dollar)

BILL_AMT3: Amount of bill statement in July, 2005 (NT dollar)
BILL_AMT4: Amount of bill statement in June, 2005 (NT dollar)
BILL_AMT5: Amount of bill statement in May, 2005 (NT dollar)
BILL_AMT6: Amount of bill statement in April, 2005 (NT dollar)
PAY_AMT1: Amount of previous payment in September, 2005 (NT dollar)
PAY_AMT2: Amount of previous payment in August, 2005 (NT dollar)
PAY_AMT3: Amount of previous payment in July, 2005 (NT dollar)
PAY_AMT4: Amount of previous payment in June, 2005 (NT dollar)
PAY_AMT5: Amount of previous payment in May, 2005 (NT dollar)
PAY_AMT6: Amount of previous payment in April, 2005 (NT dollar)
default.payment.next.month: Default payment (1=yes, 0=no)
default payment is the target variable(Y)

Lets have a look at the dataset:

	ID	LIMIT_BAL	SEX	EDUCATION	MARRIAGE	AGE	PAY_1	PAY_2	PAY_3	PAY_4	...	is Average greater than 30k and less than 50k	is Average greater than 50k and less than 70k	is Average greater than 70k and less than 100k	DUE_1	DUE_2	DUE_3	DUI
0	1	20000.0	2	2	1	24	2	2	-1	-1	...	0	0	0	3913	2413	689	
1	2	120000.0	2	2	2	26	-1	2	0	0	...	0	0	0	2682	725	1682	2
2	3	90000.0	2	2	2	34	0	0	0	0	...	0	0	0	27721	12527	12559	13
3	4	50000.0	2	2	1	37	0	0	0	0	...	0	0	0	44990	46214	48091	27
4	5	50000.0	1	2	1	57	-1	0	-1	0	...	0	0	0	6617	-31011	25835	11
5	6	50000.0	1	1	2	37	0	0	0	0	...	1	0	0	61900	55254	56951	18
6	7	500000.0	1	1	2	29	0	0	0	0	...	0	0	0	312965	372023	407007	522

Descriptive statistics associated with the data :
The following images provide statistical descriptions of data like:
Mean
Standard deviation
Quartiles
Mode
Frequency

	ID	LIMIT_BAL	SEX	EDUCATION	MARRIAGE	AGE	PAY_0	PAY_2	PAY_3	PAY_4	...
count	30000.000000	30000.000000	30000.000000	30000.000000	30000.000000	30000.000000	30000.000000	30000.000000	30000.000000	30000.000000	...
mean	15000.500000	167484.322667	1.603733	1.853133	1.551867	35.485500	-0.016700	-0.133767	-0.166200	-0.220667	...
std	8660.398374	129747.661567	0.489129	0.790349	0.521970	9.217904	1.123802	1.197186	1.196868	1.169139	...
min	1.000000	10000.000000	1.000000	0.000000	0.000000	21.000000	-2.000000	-2.000000	-2.000000	-2.000000	...
25%	7500.750000	50000.000000	1.000000	1.000000	1.000000	28.000000	-1.000000	-1.000000	-1.000000	-1.000000	...
50%	15000.500000	140000.000000	2.000000	2.000000	2.000000	34.000000	0.000000	0.000000	0.000000	0.000000	...
75%	22500.250000	240000.000000	2.000000	2.000000	2.000000	41.000000	0.000000	0.000000	0.000000	0.000000	...
max	30000.000000	1000000.000000	2.000000	6.000000	3.000000	79.000000	8.000000	8.000000	8.000000	8.000000	...

8 rows × 38 columns

Is Average greater than 10k and less than 30k	Is Average greater than 30k and less than 50k	Is Average greater than 50k and less than 70k	Is Average greater than 70k and less than 100k	DUE_1	DUE_2	DUE_3	DUE_4	DUE_5	
30000.000000	30000.000000	30000.000000	30000.000000	30000.000000	3.000000e+04	3.000000e+04	30000.00000	30000.000000	30000.
0.126500	0.009633	0.000400	0.000400	45559.750400	4.325791e+04	4.178747e+04	38436.87210	35512.013333	33656.
0.332418	0.097677	0.019996	0.019996	73173.789447	7.256594e+04	6.929536e+04	64200.61083	60553.370054	60151.
0.000000	0.000000	0.000000	0.000000	-733744.000000	-1.702347e+06	-8.546410e+05	-667000.00000	-414380.000000	-684896.
0.000000	0.000000	0.000000	0.000000	745.000000	3.295000e+02	2.627500e+02	230.00000	0.000000	0.
0.000000	0.000000	0.000000	0.000000	18550.500000	1.810250e+04	1.776900e+04	16970.00000	15538.000000	13926.
0.000000	0.000000	0.000000	0.000000	62241.500000	5.907775e+04	5.629425e+04	50259.50000	46961.500000	46067.
1.000000	1.000000	1.000000	1.000000	913727.000000	9.332080e+05	1.542258e+06	841586.00000	877171.000000	911408.

<u>Predictive Analytics.</u>

Steps performed:

1.Data preprocessing

A.Data cleaning.

The data was checked for null values.

```
In [6]: cdata = creditdata.isnull().sum()
        cdata

Out[6]: ID                                                0
        LIMIT_BAL                                         0
        SEX                                               0
        EDUCATION                                         0
        MARRIAGE                                          0
        AGE                                               0
        PAY_0                                             0
        PAY_2                                             0
        PAY_3                                             0
        PAY_4                                             0
        PAY_5                                             0
        PAY_6                                             0
        BILL_AMT1                                         0
        BILL_AMT2                                         0
        BILL_AMT3                                         0
        BILL_AMT4                                         0
        BILL_AMT5                                         0
        BILL_AMT6                                         0
        PAY_AMT1                                          0
        PAY_AMT2                                          0
        PAY_AMT3                                          0
        PAY_AMT4                                          0
        PAY_AMT5                                          0
        PAY_AMT6                                          0
        default.payment.next.month                        0
        Number of missed payments                         0
        Average Bill Amount (TD)                          0
         Is Average Bill Amount less than 10K?            0
        Is Average greater than 10k and less than 30k     0
        Is Average greater than 30k and less than 50k     0
```

The data had no missing rows and columns.

B. Data transformation

Specific columns were renamed to carry out data standardization.

```
: new_data = new_data.rename(columns = {'PAY_0':'PAY_1','default.payme

: new_data.dtypes

: ID               int64
  LIMIT_BAL      float64
  SEX              int64
  EDUCATION        int64
  MARRIAGE         int64
  AGE              int64
  PAY_1            int64
  PAY_2            int64
  PAY_3            int64
  PAY_4            int64
  PAY_5            int64
  PAY_6            int64
```

2.Feature Engineering

Feature engineering uses knowledge related to the data to create some additional features that can be then used to train the model.
Feature engineering is necessary when the available features aren't sufficient to train the model.
The below features where added to the dataset:

- Number of missed payments.
- Average Bill amount.
- Is average amount less than 10k?
- Is average amount greater than 10k and less than 30K?
- DUE_1
- DUE_2
- DUE_3
- DUE_4
- DUE_5
- DUE_6

The dataset after adding the above features contains in total 38 features.

Z	AA	AB	AC	AD	AE	AF	AG	AH	AI	AJ	
Number of missed paym	Average Bill Amount (TD)	Is Average Bill Amount less than 10K?	Is Average greater than 10k and less than 30k	Is Average great	Is Average gre	Is Average greater t	DUE_1	DUE_2	DUE_3	DUE_4	DUI
2	128.4	1	0	0	0	0	3913	2413	689	0	
2	284.6166667	1	0	0	0	0	2682	725	1682	2272	
0	1694.216667	1	0	0	0	0	27721	12527	12559	13331	
0	3855.566667	1	0	0	0	0	44990	46214	48091	27214	
0	1822.316667	1	0	0	0	0	6617	-31011	25835	11940	
0	3968.566667	1	0	1	0	0	61900	55254	56951	18394	
0	45409.91667	0	0	0	0	0	312965	372023	407007	522414	4
0	224.7666667	1	0	0	0	0	11496	-221	601	-360	
1	1086.866667	1	0	0	0	0	7956	14096	11676	11211	
0	448.65	1	0	0	0	0	0	0	0	-13007	

3. Data Modeling.

The main features of the credit card data that were used for model building are:

A total of 23 features were used

```
SEX
EDUCATION
MARRIAGE
AGE
BILL_AMT1
BILL_AMT2
BILL_AMT3
BILL_AMT4
BILL_AMT5
BILL_AMT6
PAY_AMT1
PAY_AMT2
PAY_AMT3
PAY_AMT4
PAY_AMT5
PAY_AMT6
Number of missed payments
 Is Average Bill Amount less than 10K?
DUE_2
DUE_3
DUE_4
DUE_5
DUE_6
dtype: int64
```

Over Sampling and Under Sampling of data to overcome the data imbalance.

```
:  from collections import Counter
   from imblearn.over_sampling import RandomOverSampler
   from imblearn.under_sampling import RandomUnderSampler
   from imblearn.over_sampling import SMOTE
   # summarize class distribution
   over = RandomOverSampler(sampling_strategy=0.5)
   X,y = over.fit_resample(xdata,ydata)
   print(X.shape)
   print(y.shape)
   # define undersampling strategy
   under = RandomUnderSampler(sampling_strategy=0.5)
   # fit and apply the transform
   X, y = under.fit_resample(X, y)
   print(X.shape)
   print(y.shape)
   # summarize class distribution
   print(Counter(y))

   (35046, 23)
   (35046, 1)
   (35046, 23)
   (35046, 1)
   Counter({'default': 1})
```

Train test split

The data is split into training and testing sets with 70%: Training and 30%: Testing sets.

```
In [38]: from sklearn.model_selection import train_test_split
         X_train, X_test, Y_train, Y_test = train_test_split( X, y, test_size=0.3)

         from sklearn.preprocessing import StandardScaler
         scX = StandardScaler()
         X_train = scX.fit_transform( X_train )
         X_test = scX.transform( X_test )
```

Various classification models were implemented using the sci-kit-learn package in python, and the best model was determined by carrying out a comparative study of the model.

Overview

The following models are implemented to predict the credit card defaulters.

- 1. Logistic Regression
- 2. Naive Bayes
- 3. Decision Tree Classifier
- 4. Random forest classifier
- 5. AdaBoost classifier
- 6. Support Vector Classifier

The overview of the performance of these models is:

Comparision of various algorithms.

Classifier	Precision	Recall	Accuracy	ROC_AUC score
Naive Bayes	0.4	0.87	0.46	0.54
Logistic Reg	0.73	0.35	0.737	0.64
Decision tree	0.62	0.55	0.735	0.68
Random forest	0.69	0.55	0.76	0.71
Ada Boost	0.65	0.48	0.74	0.67
Support Vector	0.70	0.38	0.74	0.64

After carrying out a comparative analysis of data, it was determined that the Random forest Classifier is the best-suited model for the data with a ROC_AUC score of 0.71 and an accuracy rate of 0.76.

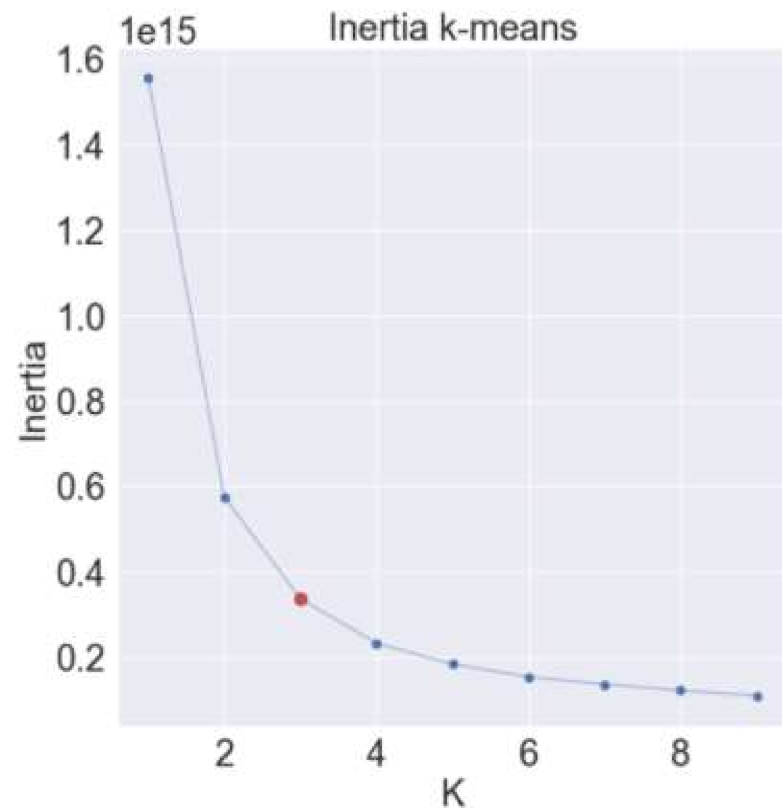
Descriptive Analytics

Clustering is an unsupervised learning technique in which objects with similar characteristics are grouped together within the same cluster and with dissimilarities placed in different clusters. Various clustering algorithms were implemented on the dataset namely:
K-means clustering
Spectral clustering
Clustering based on TSNE

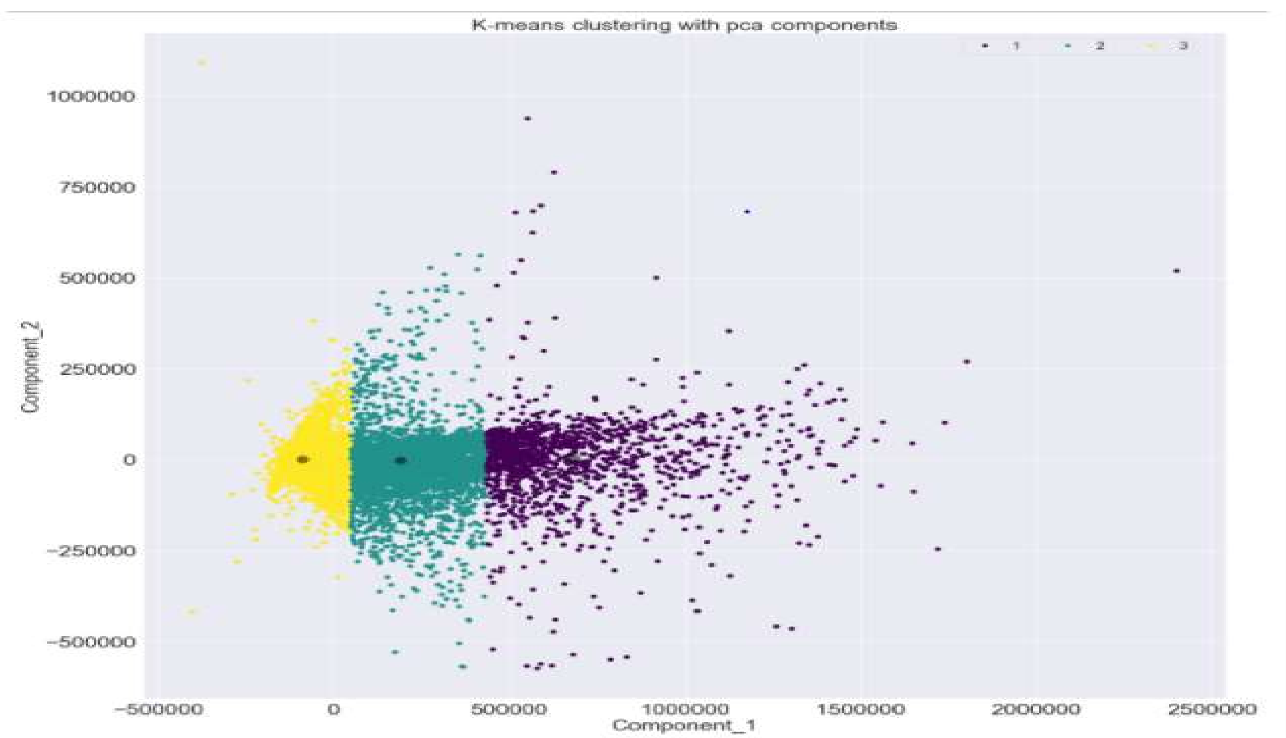
1.K- means clustering
Initially, PCA decomposition was performed on the dataset, resulting in 2 components.
K- means clustering was performed on the transformed dataset with the value of
k = 3.

The optimal value of k is 3 and is determined using the elbow method.

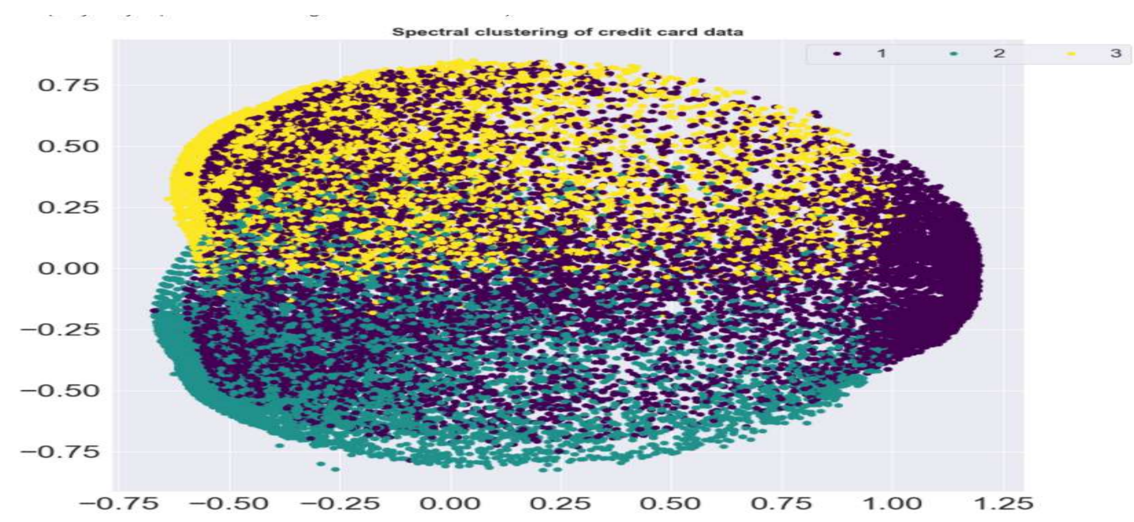
```
Out[94]: Text(0.5, 1.0, 'Inertia k-means')
```



The final clustering graph obtained(K-means):

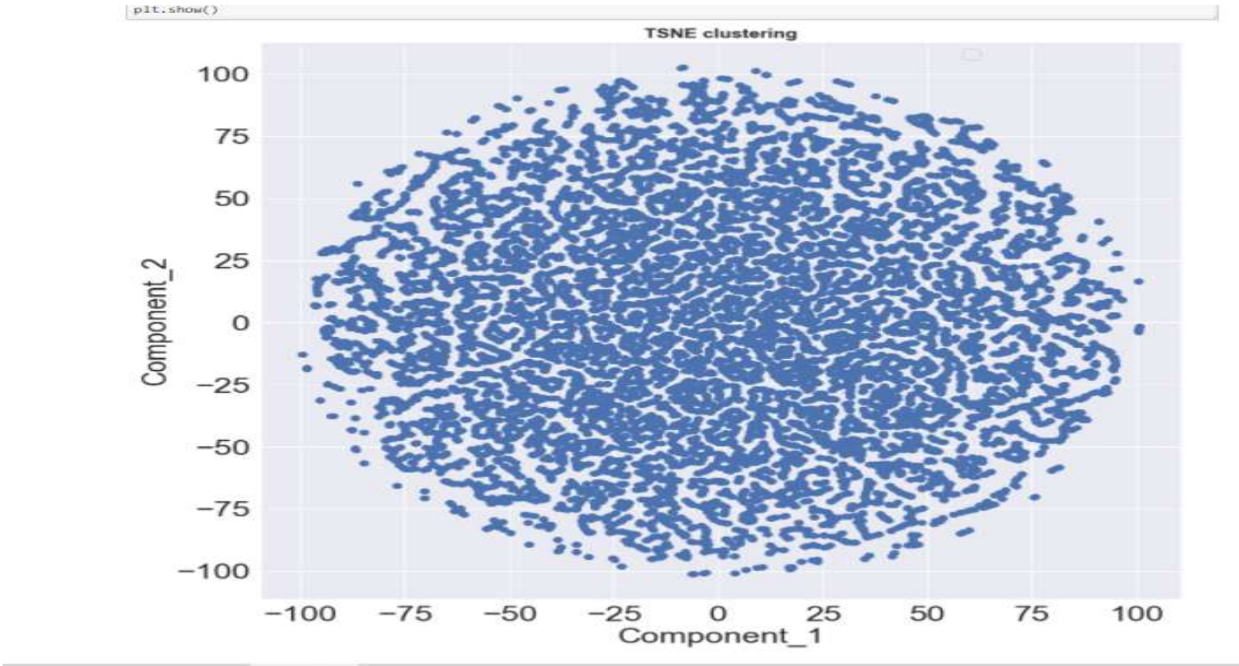


2.Spectral clustering
Spectral clustering is a special type of clustering algorithm that takes into consideration, the eigen vectors to carry out clustering in an n-dimensional space.
Spectral clustering of credit card data resulted into 3 overlapping clusters. The silhouette score of the clustering is 0.12, thus indicating that the clusters are overlapping.



3.Clustering based on TSNE

TSNE is a dimensionality reduction technique, such as the Principle Component Analysis(PCA). The main aim of TSNE is to take a set if points in the high dimensional dataset and represent them to a lower dimensional dataset like a 2D plane.
TSNE takes an important parameter called as perplexity, which determines the number of close neighbors.
The typical values of perplexity are between 5 and 50.
The algorithm gives different results on successive runs.
Below is the TSNE output for the credit cards client dataset.



Conclusion:

It can be concluded that for the given credit card dataset, random forest algorithm works best for carrying out predictive analytics followed by decision tree classifier.
The available dataset wasn't efficient to carry out predictions, so new features had to be added to improve the efficiency of the models.
Clustering analysis helped in understanding the behavior of data and thus drawing conclusions.
Predictive analysis of the credit card defaulters is essential as it can help the financial institutions in dealing with such clients and to save huge amounts.

References

1. https://towardsdatascience.com/k-means-clustering-with-scikit-learn-6b47a369a83c
2. https://365datascience.com/pca-k-means/
3. https://www.kdnuggets.com/2020/05/getting-started-spectral-clustering.html
4. https://www.kaggle.com/rpsuraj/classifying-credit-card-defaulters-deployment/notebook
5. https://towardsdatascience.com/building-a-logistic-regression-in-python-step-by-step-becd4d56c9c8
6. https://www.kdnuggets.com/2020/04/visualizing-decision-trees-python.html
7. https://towardsdatascience.com/support-vector-machine-introduction-to-machine-learning-algorithms-934a444fca47
8. https://machinelearningmastery.com/metrics-evaluate-machine-learning-algorithms-python/