

CMP4011 Big Data and Cloud Computing

Project Report

Team 9

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Sec** | **B.N** | **Code** |
| Ahmed Hosny | 1 | 2 | 9202077 |
| Ahmed Sabry | 1 | 4 | 9202119 |
| Basma Elhoseny | 1 | 16 | 9202381 |
| Zeinab Moawad Fayez | 1 | 28 | 9202611 |

## Problem Statement:

To assess the risk associated with approving a loan request from a customer, bankers rely on evaluating the customer's transactional history. We propose the development of an intelligent system that would analyze customer credit information, encompassing payment history, credit utilization, and length of credit. It would then assign a credit score to the customer, which the bank would use to determine loan approval based on a predefined threshold.

## Data Set:

• **Main:** <https://www.kaggle.com/datasets/parisrohan/credit-score-classification> [31.14 MB 150K Example] [25 useful feature excluding name, id, ssn, and month

## Project Pipeline:

A diagram of a flowchart

Description automatically generated

## Analysis and Solutions:

### Data Preprocessing:

*DataPreprocessing* Class includes all pre/post processing done on data: [data\_preprocessing.py]

###### Preprocessing:

1. Drop unimportant features such as:
   1. ID
   2. Customer\_ID
   3. Name
   4. SSN
   5. Month
   6. Credit\_History\_Age (for time series analysis)
2. Correct and Cleaning of some values like age containing 29\_ & Conversion of numerical values read from csv as string back to numerical such as num\_of\_loans.
3. Handling Missing Values based on series of 8 (because 8 record represents same customer in 8 different months (Local [Per Customer])
   1. Categorial Missed Data are replaced by the most frequent within the 8 records of the customer.
   2. Continuous Missed Data are replaced by the mean of the 8 records of the customer.
   3. Continuous [non-float] Missed Data are replaced by the mod of the 8 records of the customer (such as age)
4. Convert Categorial Data to One hot encoder or Label encoded [According to the model used]
5. Normalization for Numerical Data [For Kmeans]
6. Standardization for Numerical Data [For other Predictive Models]

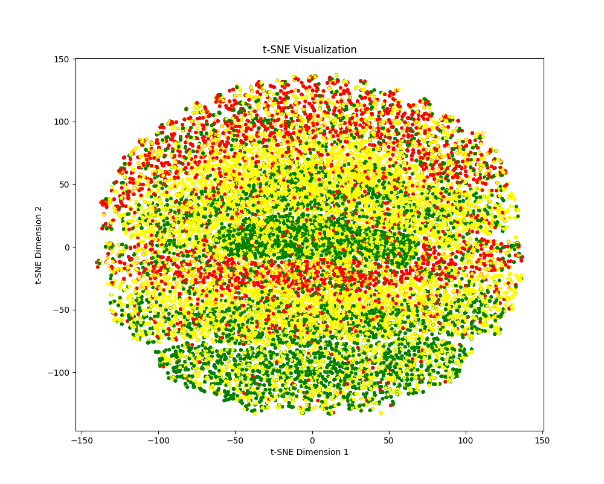
###### Postprocessing:

1. Further Cleaning for the data observed from visualizations 😉 (Global [Not per customer])
   1. Replace !@9#%8 in Payment\_Behaviour by Low\_spent\_Small\_value\_payments.
   2. Replace **\_** in Credit\_Mix by Standard which is the value for most of the customers.
2. Removing outliers which are out of bound of the normal distribution of the feature [mean ± 3\*std]
3. Split the type\_of\_loan column into 9(unique) columns such that each is binary 0 or 1 [later we find out that these features are useless, they are the least features contributing to the credit\_score decision (This idea is clear below 😊)]

### Data Visualization:

After postprocessing Step we have used different visualizations to take insights from the data [preprocessing\_visualizations.ipynb]

We have applied T-SNE on the 3 clusters of the data (good-standard-poor) to see clusters. [descrpitive\_analysis.ipynb]

 A diagram of a brain

Description automatically generated

Figure 1 T-SNE for 3 Classes (With most of features) Figure 2 T-SNE for 3 Classes (After Dropping features of importance < 0.04)

### Insights From Data:

* …………………………
* No Correlation between Continuous Features [Insight from the correlation Matrix of between all features]

##### **Business Insights**

* The most important feature in the credit score decision is the outstanding debt, giving loan to a customer who has a many outstanding loans is risky 😃 [insight from important feature diagram of random forest]
* Occupation & Age aren’t and effective features for credit Score Classification [insight from EDA histogram]
* Num of banks accounts & credit cards are correlated together [Insight from EDA & Features correlation matrix]
* The customers of good credit score are mostly the ones with high delay from due date this might be due to the types of their payments which is thought to be high [Insight from EDA]

### Model/Classifier training:

We have defined *Trainer* class which is common for training and evaluation for all predictive analysis models[trainer.py]

1. Splitting training data into train and test [NB: We haven’t used the test subset provided with the data set because it is unlabeled]
2. …………………………………………………………………………………

## Results and Evaluations:

### SVM:

…………………………………………………………………

### Random Forest:

1. We used Postprocessed data.
2. Convert Categorial Data using Label Encoder (Label Encoder)
3. Standardization for Continuous Data (Standard Scaler)
4. From the trained Model we could plot the features Importance

A graph of a number of blue and white bars

Description automatically generated

Figure 3 Feature Importance from Random Forest Classifier

The most important feature for credit\_score prediction is Outstanding\_debt (Represents the remaining debt to be paid in USD) which is logic 😎

……………………………………….

### XgBoost:

……………………………………………………………..

|  |  |  |  |
| --- | --- | --- | --- |
| **Metric** | **SVM** | **Random Forest** | **XgBoost** |
| **Train** | | | |
| **Accuracy** | 0.787 | 1.0 | 1.0 |
| **Recall** | 0.758 0.777 0.802 | 1.0 1.0 1.0 | 1.0 1.0 1.0 |
| **Precision** | 0.679 0.797 0.823 | 1.0 1.0 1.0 | 1.0 1.0 1.0 |
| **F1-score** | 0.716 0.789 0.812 | 1.0 1.0 1.0 | 1.0 1.0 1.0 |
| **Test** | | | |
| **Accuracy** | 0.7546 | 0.804 | 0.791 |
| **Recall** | 0.710 0.743 0.777 | 0.758 0.835 0.802 | 0.738 0.803 0.803 |
| **Precision** | 0.641 0.772 0.790 | 0.775 0.786 0.825 | 0.767 0.783 0.803 |
| **F1-score** | 0.674 0.757 0.783 | 0.766 0.810 0.813 | 0.752 0.793 0.803 |

### Kmeans:

We have used Sklearn built-in Kmeans Clustering for further segmentation of the customers.

We have seen in Figures (1&2) that segmenting the customers into 3 classes according to the credit score isn’t enough, there is a lot of overlapping between classes. So, we tried out different no of classes such as 5.

A colorful diagram of a brain

Description automatically generated

Figure 4 Segmenting into 5 Clusters

### Kmeans (Map-Reduce):

##### Mapper:

Computes the closest Centroid for each point.

**Point(P) 🡪 (P, Ci), i is the index of the closest centroid.**

##### Combiner:

Combine the points that belong to the same centroid (Partial Sum) [Per Machine]

**(P1,C1),(P3,C1) 🡪 (C1, P1+P3), i is the index of the closest centroid.**

##### Reducer:

##### Compute the new centroids by dividing the partial summation per centroid by the no of points in the newly computed centroid!

**(C1, P1+P3) 🡪 (C1, (P1+P3)/2), i is the index of the closest centroid.**

A diagram of a diagram

Description automatically generated with medium confidence

Figure 5 K means Clustering Map-Reduce

A close-up of a diagram

Description automatically generated

Figure 6 Kmeans Clustering Map-Reduced 5 Clusters

The same scatter plot for the TSNE Reduction is obtained from Sklearn 😎

* Converged after 25 iterations.

### Kmeans Bench Marking: (5 Classes Clustering):

|  |  |  |
| --- | --- | --- |
| **POC** | **Sklearn** | **Pyspark Map-Reduce**  **From Scratch** |
| **Silhouette Score** | 0.25215559235815665  (Cases of all features used we got 0.09 🙂) | 0.25276471972112385  (Same as Sklearn 🙂) |
| **Training Time** | 0.6300568580627441 | 124.21908068656921 |

The Sklearn version is very fast compared to the map-reduced one this is due to the parallelization implemented by the built-in version by Sklearn.

## Bench Marking Problem:

Others on Kaggle

|  |  |  |  |
| --- | --- | --- | --- |
|  |  |  |  |
|  |  |  |  |

## Extra Work (Bonus):

* Training/Testing of Random Forest on ML Studio

## Future Works:

* Apply Map Reduce on Multi Machine

A diagram of a diagram

Description automatically generated

Idea (1) Credit Score Classification Problem

## Problem Statement:

To assess the risk associated with approving a loan request from a customer, bankers rely on evaluating the customer's transactional history. We propose the development of an intelligent system that would analyze customer credit information, encompassing payment history, credit utilization, and length of credit. It would then assign a credit score to the customer, which the bank would use to determine loan approval based on a predefined threshold.

## Questions we will Answer:

* What is the Segment of this Customer? [Poor-Standard-Good]
* How are factors such as age and occupation correlated with monthly salary and average balance?

## Data Set:

• **Main:** <https://www.kaggle.com/datasets/parisrohan/credit-score-classification> [31.14 MB 150K Example] [25 useful feature excluding name, id, ssn, and month]

## Approach:

##### EDA (exploratory Data Analysis) Phase:

1. Carry out Statistical Analysis on the data set computing mean std ……
2. Anomalies and outliers Detections
3. Plotting Distributions [Data Visualization]
4. Data Cleaning and Handling missing values.
5. Checking correlations between features [Correlation Analysis]
6. We may need feature space reduction as PCA [to be checked later when we start the analysis phase]

##### Descriptive Analysis Methods:

1. KMeans Clustering to segment customers into clusters based on their credit scores. [Further Clustering]
   1. This idea is insighted from <https://www.kaggle.com/code/jayrdixit/credit-scoring> [Unsupervised]
2. Association Rules Between Features
   1. Such as {Occupation Doctor} => {Credit Score High}.
   2. Such as {Occupation Developer} => {Credit Score Low}.

##### Predictive Analysis Methods

1. Random Forest [**Map Reduced**]
2. KNN [**Map Reduced**]
3. Naïve Baye’s Classifier [**Map Reduced**]
4. Logistic Regression
5. SVM