Final Project

CMPE 255 Data Mining

Instructor: Gheorgi Ghuzun

Department of Computer Engineering

Payment Behavior Analysis: Analytics and Predictions for Customers' Delinquency Data

Group 8

1. Nanthana Thanonklin nanthana.thanonklin@sjsu.edu SID 010316228
2. Sameer Ali Hannan sameerali.hannan@sjsu.edu SID 013751842
3. Weihua Zhao weihua.zhao@sjsu.edu SID 017467346

Code Repository: https://github.com/thinkpos2022/255-delinquency-telecom-prediction

DATASET INFORMATION:

The Delinquency Telecom Dataset provides insights into how customers handle their phone credits, with a focus on payments and recharging. It reveals if a user repaid a borrowed credit within 5 days. This dataset has 209,593 rows and 36 columns, showing users' mobile numbers, how much they spend daily and their average credit left with particular attention to 30-day and 90-day intervals. It further highlights credit borrowing patterns, including the number of credits taken, the borrowed amounts, and when they are paid back. The dataset also includes the customer's telecom region. All monetary figures are presented in Indonesian currency. This comprehensive data assists telecom companies in forecasting potential payment challenges.

Dataset Link:- https://www.kaggle.com/code/sivakrishna3311/delinquency-telecom-model

Section 1 Introduction

- ***Motivation***

Having learned the theories and developed technical skills, this project presents an excellent opportunity for us to apply what we've learned in a practical setting. The varied and complex nature of financial data, with its large amounts of information, quick changes, and diverse elements, is an ideal testing ground for various machine learning methods, from basic regression to more sophisticated neural networks. Our goal is to try out different algorithms, enhance their effectiveness and really understand what they can do in real-life scenarios.

- ***Objective***

The primary goal of this project is to perform a comprehensive data mining analysis to help telecom companies gain better insights into whether their customers might be late or miss payments. Given the large number of attributes in the dataset, we aim to identify the most significant features that impact customer delinquency. By building a model to predict potential payment issues, telecom companies can understand what causes these delays, allowing them to design specific strategies to address and reduce delinquency.

Section 2 System Design & Implementation details

- ***Data Preprocessing Algorithms***

  - EDA is conducted using an algorithm that categorizes account balances into 'low', 'average', and 'high' based on percentile thresholds. It then computes the prevalence of a specific 'label' across these categories and expresses it as percentages. These percentages are visualized through bar charts.
  - *Standardization and Outlier* Outliers are addressed by replacing values that are considered extreme (based on the z-score) with the median of the column.
  - *Transformation* To stabilize variance and make the data more normal distribution-like, a cube root transformation is applied to the data.
  - *MaxAbsScaler* is used to scale each feature by its maximum absolute value. This is done so that all the features contribute equally to the model's performance.

- *Dimensionality Reduction* Principal Component Analysis (PCA) is implemented for dimensionality reduction. Before PCA, the data is scaled using StandardScaler from the sklearn.preprocessing module. The PCA is configured to retain 20 components initially.
- *Visualization of PCA Results* The explained variance ratio of each principal component is calculated and plotted to visualize the cumulative explained variance. By looking at how much variance each component adds to the model, we can use this information to determine the number of components to keep. From this analysis, we chose to select 3 principal components as they collectively maximize the explained variance while maintaining model simplicity and efficiency.

- ***Machine Learning Algorithms***

We used the following Machine Learning algorithms for our project:-
1. *Logistic Regression*: Logistic Regression models the probability of loan delinquency based on input features. It estimates the odds of a customer defaulting on a loan, providing a clear and interpretable framework for understanding the impact of various factors on delinquency.

2. *K Nearest Neighbors:* K Nearest Neighbors assesses loan delinquency by considering the similarity between a given loan transaction and its neighboring transactions. It classifies a loan based on the majority class of its k-nearest neighbors, making it effective for capturing local patterns in the delinquency data.

3. *Decision Tree Classification*: Decision Tree Classification employs a tree-like structure to evaluate different conditions and features of loan transactions, leading to a final decision regarding delinquency. It is adept at capturing complex relationships and providing a transparent decision-making process.

4. *Random Forest Classification*: Random Forest combines multiple decision trees to enhance predictive accuracy and robustness. It leverages the collective insights from a multitude of trees to improve the model's generalization and resilience to overfitting, making it effective for delinquency prediction in telecom data.

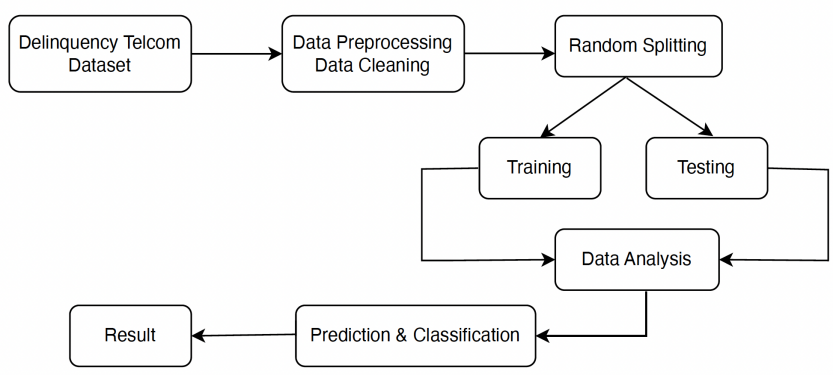
5. *Adaptive Boosting*: Adaptive Boosting (AdaBoost) focuses on weak learners and iteratively adjusts their weights to create a strong ensemble model. It is particularly useful for emphasizing the importance of misclassified instances, potentially improving the model's performance in capturing nuanced patterns in delinquency behavior.

6. *Gradient Boosting*: Gradient Boosting builds a series of decision trees sequentially, with each tree aiming to correct the errors of its predecessor. It excels at capturing complex relationships in the data and is effective in improving predictive accuracy for delinquency prediction in telecom scenarios.

- ***Technologies & Tools Used***

- **Python** was chosen as our programming language due to its vast array of specialized libraries that streamlines data science tasks like analysis, visualization and machine learning.
- **GitHub** is used as a centralized platform for storing both our dataset and the final code. We collaborated using Google Colab, a decision that gave us access to powerful GPUs and TPUs making it easier to handle our large dataset without downloads.
- **Python libraries:** *Pandas* for data analysis, *NumPy* for complex calculations, and *Matplotlib/Seaborn* for data visualization. *Scipy* was our choice for advanced math and stats.

- ***System design/architecture/data flow***



Section 3 Experiments / Proof of concept evaluation

- ***Dataset***

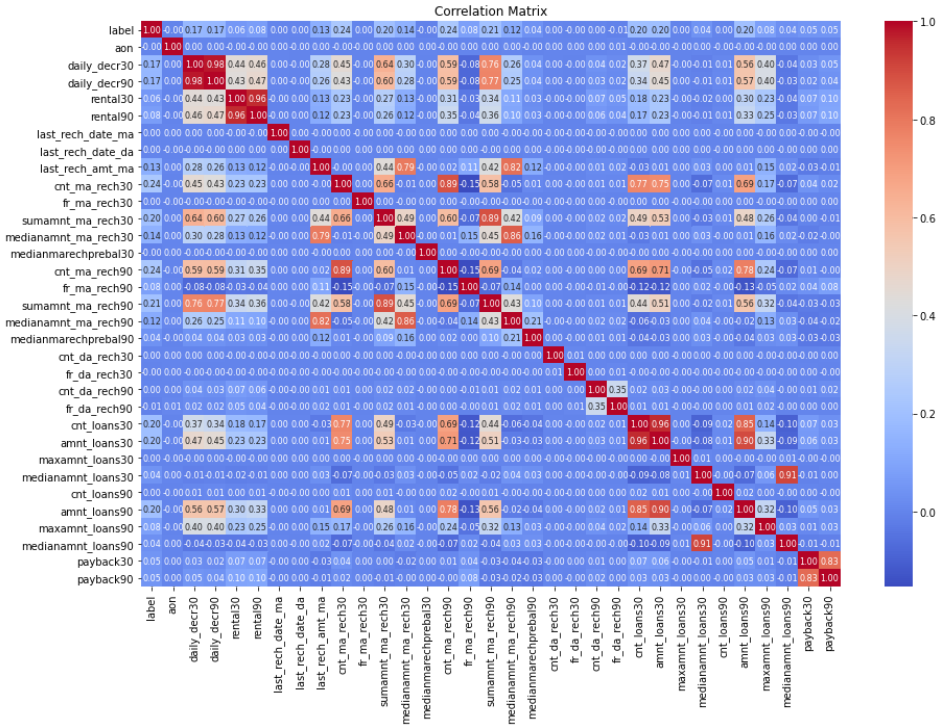
  - *Dataset Name*: Delinquency Telecom Dataset (Historic Delinquency Telecom Data)
  - *Source*: https://www.kaggle.com/datasets/sivakrishna3311/delinquency-telecom-dataset
  - *Type of Data*: The dataset contains a mix of numerical and categorical data.
  - *Size of Data*: The download size of the file is 30.1 MB. It occupies approximately 96.2 MB of memory.

- ***About dataset:***

  - *Phone Credits Management* The dataset includes various attributes that could be used to analyze how customers manage their phone credits. Attributes like 'daily_decr30', 'daily_decr90', 'rental30', 'rental90', and various recharge-related features suggest a focus on phone credit usage and recharging patterns.

  - *Repayment of Borrowed Credit* The dataset includes attributes related to loans, such as 'cnt_loans30', 'cnt_loans90', 'amnt_loans30', 'amnt_loans90' that could be used to analyze whether users repay borrowed credits within a specific period.

  - The combination of loan, repayment, and usage data in this dataset could indeed assist telecom companies in forecasting potential payment challenges.

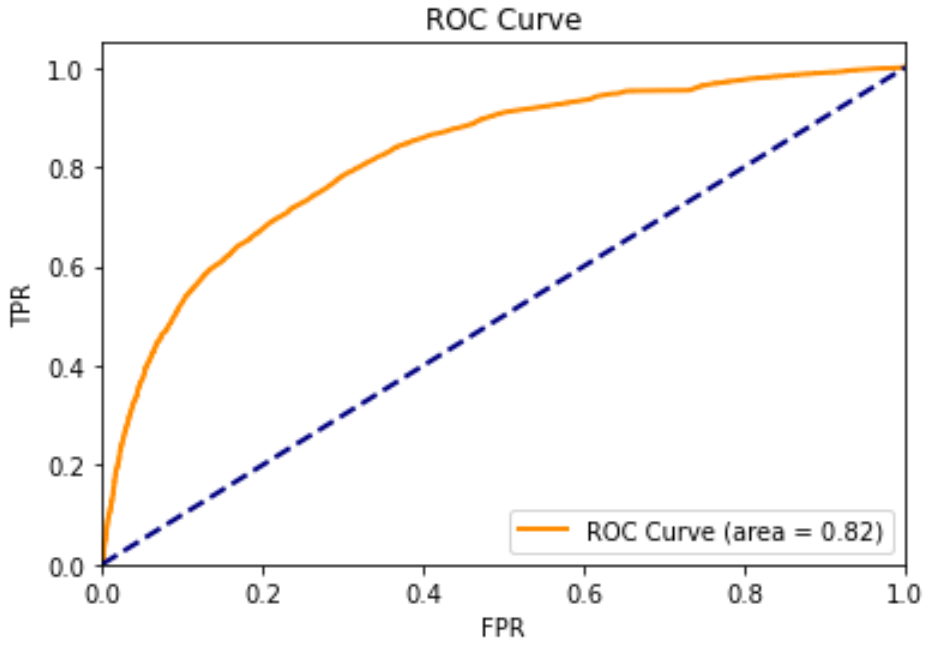


- ***Data Inspection***

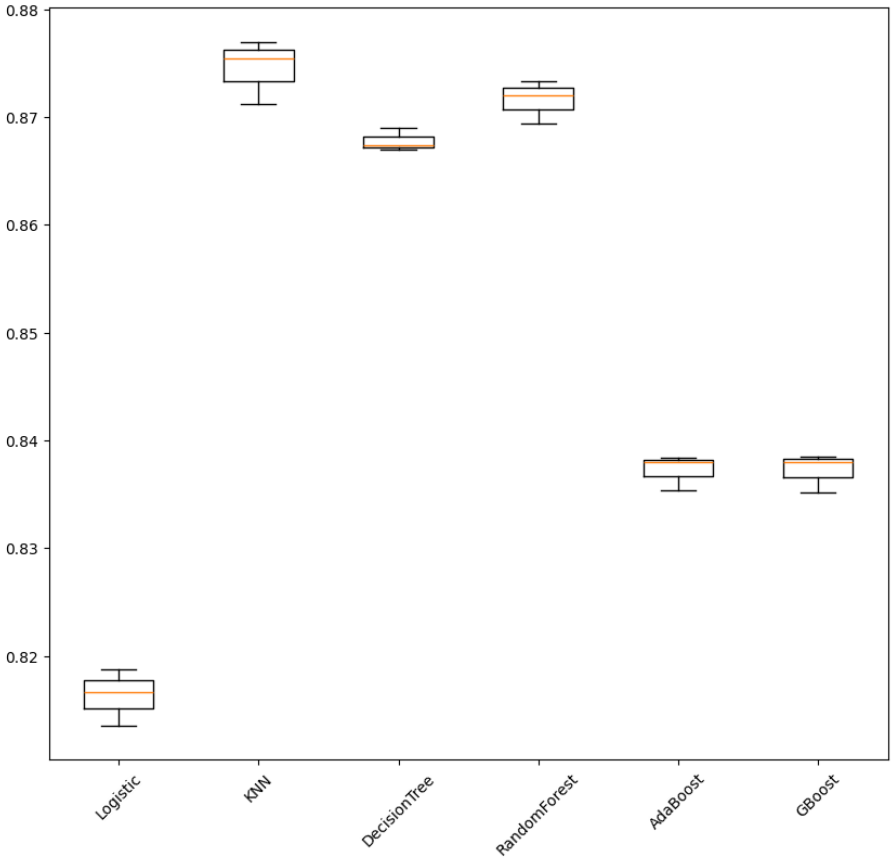
    - The `info()` method is useful for addressing any missing values, converting data types, or renaming columns.
    - The `head()` function is being used to quickly inspect the first ten entries of the DataFrame to gain an understanding of what the data looks like (numerical, categorical, textual, etc.) If there are any obvious inconsistencies or anomalies in the data that might need cleaning or further investigation including the column names and the type of data they hold.
    - The `unique()` method is used to count unique values in each column of a data form to assess data variety and cardinality. Columns with a high number of unique values could be identifiers, while those with fewer unique values may be categorical. Columns that have only one unique value (low variance) across all rows are not useful for analysis as they do not contribute any information that can differentiate between rows.

- ***Methodology followed***

    The methodology for model evaluation involves splitting the dataset into a training set comprising 70% (188,100 rows) of the data and a test set containing 30% (20,900 rows). This is a common practice known as a 70/30 train-test split. The size of the dataset allows for a robust evaluation of model performance. To further enhance reliability, information about the use of techniques like n-fold cross-validation would be beneficial.



Algorithm Comparison



Based on the aforementioned outcomes, it is evident that the K Nearest Neighbor model outperforms the others. Through a comprehensive comparison of bias error and variance error across all algorithms, it is concluded that KNN is the most effective, and therefore, it will be employed for predicting loan defaulters.

- ***Analysis of results***

```
Logistic: 0.816300 (0.000007)
KNN: 0.874537 (0.000009)
DecisionTree: 0.867824 (0.000001)
RandomForest: 0.871614 (0.000004)
AdaBoost: 0.837251 (0.000003)
GBoost: 0.837215 (0.000003)
```

Model	Training	Test
Logistic Regression	0.8163	0.9474
K Nearest Neighbors	0.8745	0.9730
Decision Tree	0.8678	0.9775
Random Forest	0.8716	0.9889
AdaBoost	0.8372	0.9524
GBoost	0.8372	0.9524

These are accuracy scores for different machine learning models obtained through a process such as cross-validation. The numbers in parentheses represent the standard error associated with each accuracy estimate, providing a measure of the variability or precision of the model's performance. Among the models, K Nearest Neighbors (KNN) achieved the highest accuracy at 0.874537, followed by Random Forest at 0.871614, Decision Tree at 0.867824, Logistic Regression at 0.816300, and both AdaBoost and Gradient Boosting (GBoost) at 0.837251 and 0.837215, respectively. The lower standard error values suggest a relatively tight confidence interval around each accuracy estimate.
When testing the model, a dataset consisting of 100 random rows of the original dataset was used. Different from training, random forest has the best f1 score, which is 0.9889. We concluded the

Section 4 Discussion & Conclusions

• Difficulties faced

During our initial model training, we faced various data preprocessing challenges like dealing with missing or inconsistent data and handling outliers. Our initial results showed unusually high accuracy, which seemed suspicious. Upon revisiting our preprocessing steps, we realized we had included the label column in our predictions and hadn't handled outliers properly. After correcting these mistakes and refining our preprocessing, we were able to achieve more realistic results.

• Things that worked

Our methods for data splitting, data preprocessing, and model training all worked after careful inspections.

• Things that didn't work well

The current solution utilizes all available features in the dataset for modeling, lacking any feature selection or dimensionality reduction methods. This approach may result in overfitting and a potential decline in model performance. Advanced feature engineering techniques, such as time-series decomposition or incorporation of lagged features, have not been explored. Implementation of these techniques could furnish the model with supplementary information, enhancing its predictive accuracy.

• Future work

- Explore advanced modeling techniques

- Incorporate additional features like loans and financial details

- Expand the dataset to include a broader range of financial information

- Enhance the predictive capabilities of the models

• Conclusion

The delinquency dataset proves highly insightful, enabling through visualization and analysis of numerous data attributes. Graphs and visual aids contribute to a clearer understanding of data and its interrelationships. The preprocessing and analysis stages offer valuable experience in managing raw data. Working with multiple models and their training proves to be an enriching experience, allowing the identification of various data patterns and gaining insights into the workings of diverse machine learning models. The application of metrics, including mean squared error, instills confidence in the solution.

In conclusion, our findings suggest that KNN is a more suitable option for predicting the probability of loan transactions. However, further experimentation and fine-tuning of models may be necessary to achieve optimal results.

For model KNN

	precision	*recall*	*f1-score*	*support*
0	*1.00*	*0.50*	*0.67*	*10*
1	*0.95*	*1.00*	*0.97*	*90*
accuracy			*0.95*	*100*
macro avg	*0.97*	*0.75*	*0.82*	*100*
weighted avg	*0.95*	*0.95*	*0.94*	*100*

Section 5 Project Plan / Task Distribution

We followed a standard ML pipeline consisting of Data Cleaning and Preparation, Research and Modeling, and Evaluation stages. We divided the tasks among our team members and shared our findings to complete the project efficiently.

- ***Nanthana*** was responsible for setting up the GitHub repository, and project related files. She worked on data inspection, data preprocessing, data cleaning, and Exploratory Data Analysis (EDA), data visualization through plotting.
- ***Sameer*** was responsible for lead code development of our project where he worked on testing and training of data as well as visualization. He also contributed to making powerpoint presentations as well as the final report of the project.
- ***Weihua*** was responsible for fixing the bug in feature engineering and model training. He also was responsible for code reviews and model testing. He also offered help with data preprocessing.

Reference:

sivakrishna3311. "Delinquency Telecom Model." Kaggle, 2019, www.kaggle.com/code/sivakrishna3311/delinquency-telecom-model.

Appendix

Variable	Definition	Comment
label	Flag indicating whether the user paid back the credit amount within 5 days of issuing the loan{1:success, 0:failure}	
msisdn	mobile number of user	
aon	age on cellular network in days	
daily_decr30	Daily amount spent from main account, averaged over last 30 days (in Indonesian Rupiah)	
daily_decr90	Daily amount spent from main account, averaged over last 90 days (in Indonesian Rupiah)	
rental30	Average main account balance over last 30 days	Unsure of given definition
rental90	Average main account balance over last 90 days	Unsure of given definition
last_rech_date_ma	Number of days till last recharge of main account	
last_rech_date_da	Number of days till last recharge of data account	
last_rech_amt_ma	Amount of last recharge of main account (in Indonesian Rupiah)	
cnt_ma_rech30	Number of times main account got recharged in last 30 days	
fr_ma_rech30	Frequency of main account recharged in last 30 days	Unsure of given definition
sumamnt_ma_rech30	Total amount of recharge in main account over last 30 days (in Indonesian Rupiah)	
medianamnt_ma_rech30	Median of amount of recharges done in main account over last 30 days at user level (in Indonesian Rupiah)	
medianmarechprebal30	Median of main account balance just before recharge in last 30 days at user level (in Indonesian Rupiah)	
cnt_ma_rech90	Number of times main account got recharged in last 90 days	
fr_ma_rech90	Frequency of main account recharged in last 90 days	Unsure of given definition
sumamnt_ma_rech90	Total amount of recharge in main account over last 90 days (in Indonesian Rupiah)	
medianamnt_ma_rech90	Median of amount of recharges done in main account over last 90 days at user level (in Indonesian Rupiah)	
medianmarechprebal90	Median of main account balance just before recharge in last 90 days at user level (in Indonesian Rupiah)	
cnt_da_rech30	Number of times data account got recharged in last 30 days	
fr_da_rech30	Frequency of data account recharged in last 30 days	
cnt_da_rech90	Number of times data account got recharged in last 90 days	
fr_da_rech90	Frequency of data account recharged in last 90 days	
cnt_loans30	Number of loans taken by user in last 30 days	
amnt_loans30	Total amount of loans taken by user in last 30 days	
maxamnt_loans30	maximum amount of loan taken by the user in last 30 days	There are only two options: 5 & 10 Rupiah, for which the user needs to pay back 6 & 12 Rupiah respectively
medianamnt_loans30	Median of amounts of loan taken by the user in last 30 days	
cnt_loans90	Number of loans taken by user in last 90 days	
amnt_loans90	Total amount of loans taken by user in last 90 days	
maxamnt_loans90	maximum amount of loan taken by the user in last 90 days	
medianamnt_loans90	Median of amounts of loan taken by the user in last 90 days	
payback30	Average payback time in days over last 30 days	
payback90	Average payback time in days over last 90 days	
pcircle	telecom circle	
pdate	date	