Credit Score Classification Based on Supervised Machine Learning ECS784P

Abstract— The objective of this paper is to predict and classify credit scores to analyze and evaluate candidates' performance distributions (good, standard, and poor) in financial institutions by using two machine learning algorithms named K-Nearest Neighbors and Logistic Regression. The dataset consists of 28 features, a training set, and a test set. This report discusses data exploration, data preprocessing steps and a literature review. In the final stage, we evaluate the accuracy model's performance and analyze the results.

Keywords-* *credit score classification, K-Nearest Neighbors, Logistic regression

I. Introduction

Finance institutions encounter challenging tasks, whether applications are good or bad candidates for loans, extending credit, mortgages, or other financial services. Eligibility for credit has become increasingly difficult over the years. The failure of prediction models affects incorrect decision-making and risks in business. Credit scores are not static; they can change over time based on individual financial behaviors such as paying bills on time and maintaining creditability, which can positively impact credit scores. Therefore, credit ratings are essential to determining prospective customers.

This paper discusses two supervised machine learning algorithms to classify credit scores into different credit score distributions (good, standard, and poor). If the model predicts accuracy and is trustworthy, a bank or financial institution can evaluate a person's ability to repay a loan and their creditworthiness.

Objective

I. Exploration and preprocessing data with visualizing.

II. Using a minimum of two classification models by using supervised machine learning algorithms named K-Nearest Neighbors and Logistic Regression to predict credit score distribution.

III. Evaluating the accuracy of models' performance with train and test dataset.

IV. To identify the strengths and weakness of KNN and Logistic Regression.

V. To identify the challenge, limitation, and further study.

II. Literature review

Several research and journal papers have been discussed in relation to credit score classification. In this section, five research will be reviewed, as follows:

Firstly, the research titled 'Credit Score Prediction using Genetic Algorithms: LSTM Technique' by Juliana Adisa, Samuel Ojo, Pius Owolawi, and Agnieta Pretorius.[1] This study will apply LSTM for credit scoring prediction to overcome the limitations of conventional ANN. In addition, it presents a method developed by employing a genetic algorithm to optimize long-short-term memory (LSTM) for credit scoring. Long Short-term Memory (LSTM), one of the most advanced deep learning algorithms, is seldom used for credit score prediction but is naturally suitable for the financial sector. It is a type of RNN that employs feedback connections within the network. Therefore, LSTM can represent temporal sequence data, and it is beneficial for tasks such as financial predictions, speech recognition, and natural language processing. The research determined using the LSTM parameters includes epochs, batch size, number of neurons, learning rate, and dropout. In addition, this research also investigated the performance of single classifiers and compared them with ensemble models and hybrid models. A genetic algorithm was employed as an optimization scheme to find the optimal parameters for the LSTM model. The result of the optimized model shows better prediction performance and an improved loss than an ordinary LSTM model. To sum up, the hybrid LSTM performs better than all the models applied in this research.

The second research introduces an ensemble-based method for credit score classification to improve performance and robustness, titled 'A Decision Trees Classifier-Based Ensemble Approach to Credit Score Classification' by Ashok Maurya and Shivam Gaur.[2] Ensemble-based methods are widely used in various machine learning tasks, including those in the financial industry, that contribute to credit ecosystem stability and inform financial decision-making. This research represents an ensemble framework that combines various algorithms to be more robust and less prone to overfitting, reducing bias and variance. The various algorithms are used in the ensemble model as follows: 1) Bagging Classifier: to leverage the variance reduction. 2) Extra Tree Classifier: to enhance diversity 3) Random Forest: The Bias-Variance Tradeoff 4) Histogram Gradient Boosting Classifier—Histogram Approach 5) XGB Classifier: optimized gradient descent 6) The stacking classifier. After that, combine all these advantages to make a final and reliable prediction. In the data preprocessing stage, the data was cleaned by dropping various unnecessary columns and applying the Synthetic Minority Oversampling Technique (SMOTE) to highly tackle the imbalanced dataset. In addition, encoded and normalized datasets were used before training the model. The results show high rates of accuracy like previous works, but ensemble models contribute to variance reduction, enhanced diversity, optimized gradient descent, and the bias-variance trade-off.

The third research to predict the probability of default (PD) and to assign a credit score based on PD to minimize loss is titled 'Credit Risk Assessment' by Danish Shaikh and Aakash Vishwakarma. [3] The objective of this research is to introduce a novel term called 'Xscore' and show performance metrics with 22 machine learning models, for example, HistGradient Boosting, Random Forest, Neural Network, Gradient Boosting, Decision Tree, and so on. These models will be analyzed with evaluation techniques including F1-score, recall, accuracy, precision, AUC, and Xcore. The results show that the model can achieve relatively high levels of accuracy. However, the models may struggle with positive and negative outcomes in the dataset, resulting in low AUC scores. Further research is required to determine whether the models can generalize well to new data and validate their performance in the real world.

The fourth research is titled 'An Augmentation of Credit Card Fraud Detection Using Random Under sampling' by Vipin Khattri and Sandeep Kumar Nayak.[4] The study analysed credit card fraud and problems with an imbalanced dataset that degraded the credit card fraud detection system's performance. Therefore, a random undersampling technique was applied to the datasets to make them balanced. The detection of credit card fraud can be done using one of two different paths. The first way is by using an authentication process, and the second is to use a credit card fraud detection model (CCFDM). The CCFDM is an automatic process to categorize payments between genuine and fraudulent using previous transaction patterns related to a machine learning algorithm. The experiment performed a comparative analysis by comparing the CCFDM's performance using random forest before and after implementing the random Undersampling technique on an imbalanced dataset. The results of CCFDM's performance have improved in standard performance metrics such as the area under curve (AUC), receiver operating curve, geometric mean, f-score, recall, and precision after implementing random under sampling on an imbalanced dataset. In the future, the study will work further on performance augmentation of the CCFDM by handling the outliers and feature engineering of an imbalanced dataset.

The final research is titled "Credit Risk Analysis using LightGBM and a comparative study of popular algorithms" by Dr. J. Godwin Ponsam and Dr. S. Karpaselvi.[5] Currently, random forest and linear support vector models are popular and widely used in machine learning algorithms. This research will explore more choices and introduce an ensemble model called 'LightGBM', which is an open source framework developed by Microsoft in 2017. CatBoost, on the other hand, is a decision-tree gradient boosting algorithm. The result is less noisy than a single model, and LightGBM can perform better than in Logistic Regression, XgBoost, and CatBoost. Accuracy is high and faster than other models.

III. DATA MANAGEMENT

A. Data Collection

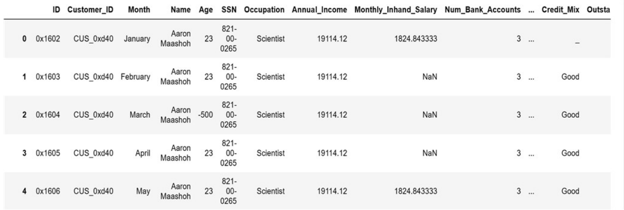
Credit Score Classification dataset is from Kaggle. Over the years, the company has collected basic bank details and gathered a lot of credit-related information.

B. Feature Description

Dataset contains 28 features. It consists of 2 files which training set and test set. The Training set has shape (49999*28) and the shape of the test set is 24999*27) (Table 1) and dataset sample (Table 2)

Feature	Description
ID	a unique identification of an entry
Customer_ID	a unique identification of a person
Month	the month of the year
Name	the name of a person
Age	the age of the person
SSN	the social security number of a person
Occupation	the occupation of the person
Annual_Income	the annual income of the person
Monthly_Inhand_Salary	he monthly base salary of a person
Num_Bank_Accounts	the number of bank accounts a person holds
Num_Credit_Card	the number of other credit cards held by a person
Interest_Rate	the interest rate on credit card
Num_of_Loan	the number of loans taken from the bank
Type_of_Loan	the types of loan taken by a person
Delay_from_due_date	the average number of days delayed from the payment date
Num_of_Delayed_Payment	the average number of payments delayed by a person
Changed_Credit_Limit	the percentage change in credit card limit
Num_Credit_Inquiries	the number of credit card inquiries
Credit_Mix	the classification of the mix of credits
Outstanding_Debt	the remaining debt to be paid (in USD)
Credit_Utilization_Ratio	the utilization ratio of credit card
Credit_History_Age	the age of credit history of the person
Payment_of_Min_Amount	the minimum amount was paid by the person
Total_EMI_per_month	the monthly EMI payments (in USD)
Amount_invested_monthly	the monthly amount invested by the customer (in USD)
Payment_Behaviour	the payment behavior of the customer (in USD)
Monthly_Balance	the monthly balance amount of the customer (in USD)
Credit_Score(Target)	rating credit score

Table 1 Feature description



	ID	Customer_ID	Month	Name	Age	SSN	Occupation	Annual_Income	Monthly_Inhand_Salary	Num_Bank_Accounts	...	Credit_Mix	Outsta
0	0x1602	CUS_0xd40	January	Aaron Maashoh	23	821-00-0265	Scientist	19114.12	1824.843333	3	...	_	
1	0x1603	CUS_0xd40	February	Aaron Maashoh	23	821-00-0265	Scientist	19114.12	NaN	3	...	Good	
2	0x1604	CUS_0xd40	March	Aaron Maashoh	-500	821-00-0265	Scientist	19114.12	NaN	3	...	Good	
3	0x1605	CUS_0xd40	April	Aaron Maashoh	23	821-00-0265	Scientist	19114.12	NaN	3	...	Good	
4	0x1606	CUS_0xd40	May	Aaron Maashoh	23	821-00-0265	Scientist	19114.12	1824.843333	3	...	Good	

Table 2: Dataset Sample

C. Data Exploration

To understand a dataset and gain insights into relationships. We can explore data by visualising it.The objective of this paper is to classify credit scores into different credit score distributions (good, Standard and Poor) to assess and evaluate candidates' performance. (Fig.1)

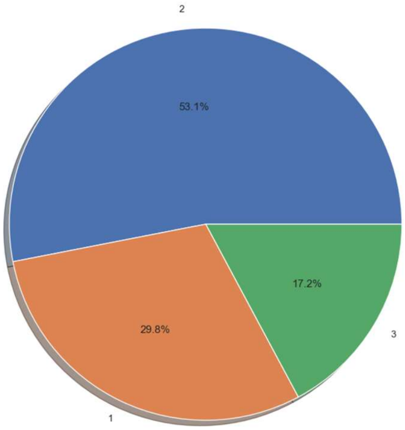


Fig. 1 Credit Score (Target Feature)

After observing the dataset, we found that some features appear invalid value or erroneous such as !@9#%8, '__-333333333333. At the pre-processing method, we will drop raw invalid values. (Fig.2)

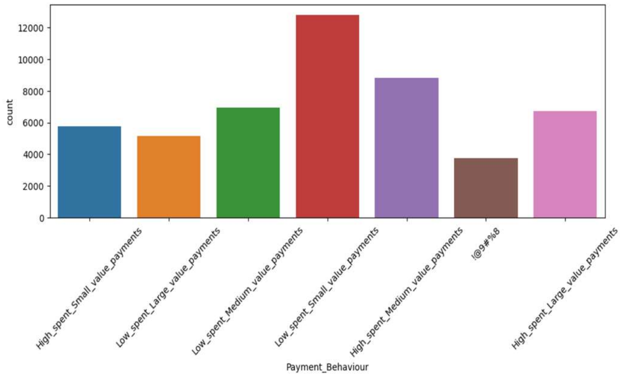
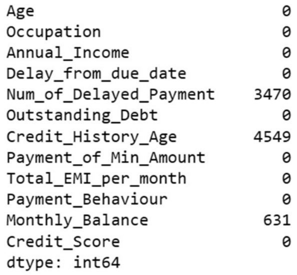


Fig. 2 Invalid data

D. Data Preprocess

i) Missing Data: Dealing with missing data is a important because it effect on model performance. These datasets have some missing values both numerical and categorical features. (Tabel 3)

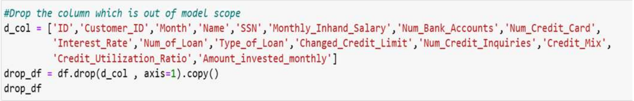


```
Age                         0
Occupation                  0
Annual_Income               0
Delay_from_due_date         0
Num_of_Delayed_Payment   3470
Outstanding_Debt            0
Credit_History_Age       4549
Payment_of_Min_Amount       0
Total_EMI_per_month         0
Payment_Behaviour           0
Monthly_Balance           631
Credit_Score                0
dtype: int64
```

Table 3 missing value

ii) Dropping unnecessary feature and invalid data

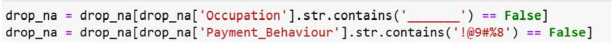
The data was cleaned by dropping various unnecessary features such as ID, Customer ID and so on (Fig.3)



```
#Drop the column which is out of model scope
d_col = ['ID','Customer_ID','Month','Name','SSN','Monthly_Inhand_Salary','Num_Bank_Accounts','Num_Credit_Card',
         'Interest_Rate','Num_of_Loan','Type_of_Loan','Changed_Credit_Limit','Num_Credit_Inquiries','Credit_Mix',
         'Credit_Utilization_Ratio','Amount_invested_monthly']
drop_df = df.drop(d_col , axis=1).copy()
drop_df
```

Fig. 3 Dropping unnecessary features

In the next step, we will drop invalid and erroneous data in the 'Occupation and Payment Behavior' column. (Fig.4)



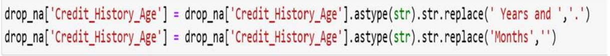
```
drop_na = drop_na[drop_na['Occupation'].str.contains('_______') == False]
drop_na = drop_na[drop_na['Payment_Behaviour'].str.contains('!@9#%8') == False]
```

Fig. 4 Dropping Invalid and erroneous row

iii) Data Transformation

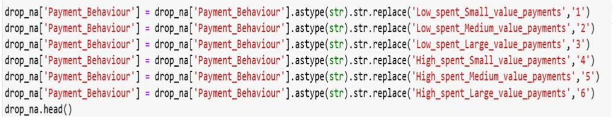
After exploration dataset we observed that 'Credit History Age' column contain both string and numerical value like '22 year and 6 month'. We have to remove sub data 'year' 'and' 'month' (Fig.5). Additionally, replace with numerical values in 'Payment Behaviour', Credit Score and Payment of Min Amount' (Fig.6). In term of 'Age' column, we select age range between 0 to 150 years old.(Fig.7)

.



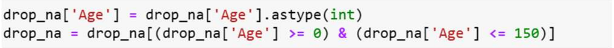
```
drop_na['Credit_History_Age'] = drop_na['Credit_History_Age'].astype(str).str.replace(' Years and ','.')
drop_na['Credit_History_Age'] = drop_na['Credit_History_Age'].astype(str).str.replace('Months','')
```

Fig. 5 Remove sub data



```
drop_na['Payment_Behaviour'] = drop_na['Payment_Behaviour'].astype(str).str.replace('Low_spent_Small_value_payments','1')
drop_na['Payment_Behaviour'] = drop_na['Payment_Behaviour'].astype(str).str.replace('Low_spent_Medium_value_payments','2')
drop_na['Payment_Behaviour'] = drop_na['Payment_Behaviour'].astype(str).str.replace('Low_spent_Large_value_payments','3')
drop_na['Payment_Behaviour'] = drop_na['Payment_Behaviour'].astype(str).str.replace('High_spent_Small_value_payments','4')
drop_na['Payment_Behaviour'] = drop_na['Payment_Behaviour'].astype(str).str.replace('High_spent_Medium_value_payments','5')
drop_na['Payment_Behaviour'] = drop_na['Payment_Behaviour'].astype(str).str.replace('High_spent_Large_value_payments','6')
drop_na.head()
```

Fig. 6 Example replacing with numerical values

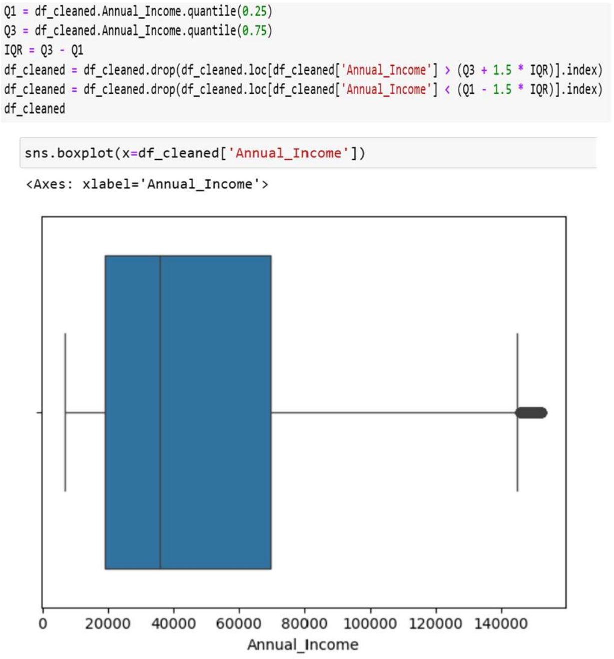


```
drop_na['Age'] = drop_na['Age'].astype(int)
drop_na = drop_na[(drop_na['Age'] >= 0) & (drop_na['Age'] <= 150)]
```

Fig. 7 Selecting age range 1-150 years old

We use the Interquartile Range (IQR) method to perform outlier removal for the Annual Income column. It computes the Interquartile Range(IQR) as the difference between Q3 and Q1.(Fig.8)

.



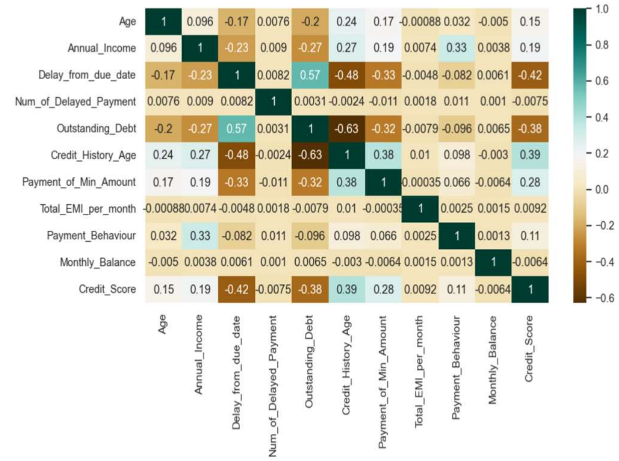
```
Q1 = df_cleaned.Annual_Income.quantile(0.25)
Q3 = df_cleaned.Annual_Income.quantile(0.75)
IQR = Q3 - Q1
df_cleaned = df_cleaned.drop(df_cleaned.loc[df_cleaned['Annual_Income'] > (Q3 + 1.5 * IQR)].index)
df_cleaned = df_cleaned.drop(df_cleaned.loc[df_cleaned['Annual_Income'] < (Q1 - 1.5 * IQR)].index)
df_cleaned
```

```
sns.boxplot(x=df_cleaned['Annual_Income'])
```

```
<Axes: xlabel='Annual_Income'>
```

Fig. 8 Outlier removal

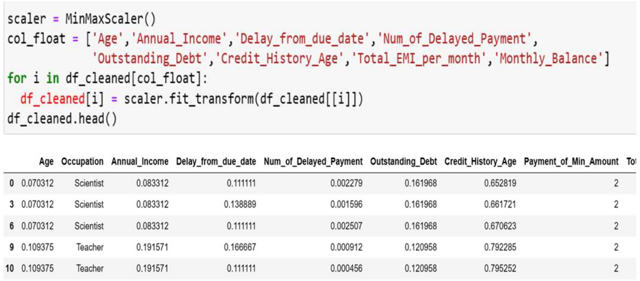
The correlation matrix of the dataset is explored to find out the interrelated features. We observed that 'Delay From Due Date 'and 'Outstanding Debt have a high negative correlation with credit score.(Fig.9)



	Age	Annual_Income	Delay_from_due_date	Num_of_Delayed_Payment	Outstanding_Debt	Credit_History_Age	Payment_of_Min_Amount	Total_EMI_per_month	Payment_Behaviour	Monthly_Balance	Credit_Score
Age	1	0.096	-0.17	0.0076	-0.2	0.24	0.17	-0.00088	0.032	-0.005	0.15
Annual_Income	0.096	1	-0.23	0.009	-0.27	0.27	0.19	0.0074	0.33	0.0038	0.19
Delay_from_due_date	-0.17	-0.23	1	0.0082	0.57	-0.48	-0.33	-0.0048	-0.082	0.0061	-0.42
Num_of_Delayed_Payment	0.0076	0.009	0.0082	1	0.0031	-0.0024	-0.011	0.0018	0.011	0.001	-0.0075
Outstanding_Debt	-0.2	-0.27	0.57	0.0031	1	-0.63	-0.32	-0.0079	-0.096	0.0065	-0.38
Credit_History_Age	0.24	0.27	-0.48	-0.0024	-0.63	1	0.38	0.01	0.098	-0.003	0.39
Payment_of_Min_Amount	0.17	0.19	-0.33	-0.011	-0.32	0.38	1	0.00035	0.066	-0.0064	0.28
Total_EMI_per_month	-0.00088	0.0074	-0.0048	0.0018	-0.0079	0.01	-0.00035	1	0.0025	0.0015	0.0092
Payment_Behaviour	0.032	0.33	-0.082	0.011	-0.096	0.098	0.066	0.0025	1	0.0013	0.11
Monthly_Balance	-0.005	0.0038	0.0061	0.001	0.0065	-0.003	-0.0064	0.0015	0.0013	1	-0.0064
Credit_Score	0.15	0.19	-0.42	-0.0075	-0.38	0.39	0.28	0.0092	0.11	-0.0064	1

Fig. 9 Correlation matrix

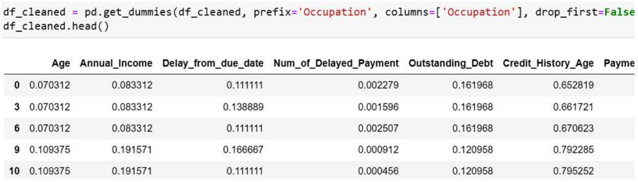
Before training the model, we use the MinMaxScaler to scale numerical features to a range between 0 and 1 (Fig. 10). Further, perform one hot encoding using the get_dummies() function (Fig. 11).



```
scaler = MinMaxScaler()
col_float = ['Age','Annual_Income','Delay_from_due_date','Num_of_Delayed_Payment',
             'Outstanding_Debt','Credit_History_Age','Total_EMI_per_month','Monthly_Balance']
for i in df_cleaned[col_float]:
  df_cleaned[i] = scaler.fit_transform(df_cleaned[[i]])
df_cleaned.head()
```

	Age	Occupation	Annual_Income	Delay_from_due_date	Num_of_Delayed_Payment	Outstanding_Debt	Credit_History_Age	Payment_of_Min_Amount	Tot
0	0.070312	Scientist	0.083312	0.111111	0.002279	0.161968	0.652819	2	
3	0.070312	Scientist	0.083312	0.138889	0.001596	0.161968	0.661721	2	
6	0.070312	Scientist	0.083312	0.111111	0.002507	0.161968	0.670623	2	
9	0.109375	Teacher	0.191571	0.166667	0.000912	0.120958	0.792285	2	
10	0.109375	Teacher	0.191571	0.111111	0.000456	0.120958	0.795252	2	

Fig. 10 Scaler MinMaxScaler



```
df_cleaned = pd.get_dummies(df_cleaned, prefix='Occupation', columns=['Occupation'], drop_first=False
df_cleaned.head()
```

	Age	Annual_Income	Delay_from_due_date	Num_of_Delayed_Payment	Outstanding_Debt	Credit_History_Age	Payme
0	0.070312	0.083312	0.111111	0.002279	0.161968	0.652819	
3	0.070312	0.083312	0.138889	0.001596	0.161968	0.661721	
6	0.070312	0.083312	0.111111	0.002507	0.161968	0.670623	
9	0.109375	0.191571	0.166667	0.000912	0.120958	0.792285	
10	0.109375	0.191571	0.111111	0.000456	0.120958	0.795252	

Fig. 11 One Hot Encoding

IV METHODOLOGY

A . K-Nearest Neighbours (K-NN)

- The k-nearest neighbours (KNN) algorithm is a non-parametric, supervised learning classifier, which uses proximity to make classifications or predictions about the grouping of an individual data point. There are several distance measures for example Euclidean distance , Manhattan distance**,** Minkowski distance and Hamming distance. Euclidean distance method is the most commonly used distance measure, it measures a straight line between the query point and the other point. This can be represented with the following formula:

$$\text{distance}(x, X_i) = \sqrt{\sum_{j=1}^{d}(x_j - X_{i_j})^2]}$$

- K-NN algorithm compute k value that defining how many neighbours will be checked to determine the classification of a s specific query point.

- The strength of KNN is that it requires no training time and is straightforward to understand and implement. Additionally, it can adapt to use complex data. On the other hand, it requires feature scaling and is sensitive to noise and irrelevant features.

B. Logistic Regression

Logistic regression is the most popular under the Supervised Learning technique for predicting the categorical variables using a given set of independent variables. It is used for solving the classification problem. The sigmoid function also call logistic function gives an 'S' shaped curve which map a value between 0 and 1. If the output of the sigmoid function is more than 0.5, we can classify the outcome as 1 or YES, and if it is less than 0.5, we can classify it as 0 or NO. (Fig.12)

$$f(x) = \frac{1}{1 + e^{-x}}$$

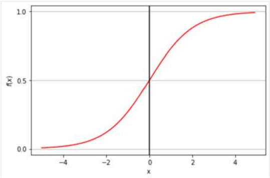


Fig. 12 Sigmoid Function

The strength of logistic regression is that it is robust to overfitting, and it can handle large datasets with many features, making it easy to understand the relationship between features and the target. Conversely, it struggles with highly imbalanced datasets and is primarily used for binary classification problems; it is not suitable for multi-class classification.

V.RESULT AND EVALUTION

Before training KNN and Logistic Regression model , the data was split 80% for training and 20% for testing data and cross validation was used to determine the model accuracy . In addition, we drop 'Credit Score' targeted feature in train_x data. We use evaluation technique as follow.

- *Evaluation Techique*

▪ Accuracy is a metric for evaluation classification model. Formally, accuracy has the following definitions.

$$\text{Accuracy} = \frac{\text{Number of correct predictions}}{\text{Total number of predictions}}$$

▪ F1 score computes the average of precision and recall, The best value of F1 score is 1 and the worst is 0. The mathematical formula for the F1 score is as follows.

$$F1 = 2 * \frac{Precision * Recall}{Precision + Recall}$$

▪ Precision is a metric how model predict the positive class. The formular for precision as follows.

$$\text{Precision} = \frac{TP}{TP + FP}$$

▪ Recall is a metric how model identified true positive from all the actual positive samples in the dataset. The formular for recall as follows.

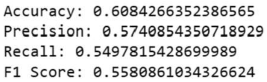
$$\text{Recall} = \frac{TP}{TP + FN}$$

▪ Cross Validation is one of the technique use to test the effectiveness of models. It used to evaluate a model with a limited data.

- Results

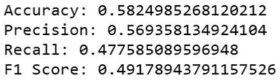
After training the KNN and logistic regression models, the performance of the KNN and logistic regression models shows that the F1 score and accuracy are quite moderate, approximately 49%–60%. Additionally, KNN-cross-validation and logistic cross-validation show the F1 score and accuracy range of 48%–58%. In conclusion, the results demonstrate that the KNN model performed better than the logistic regression model with both the F1 score (55%, 49%) and accuracy (60%, 58%), respectively. (Table 4)

KNN



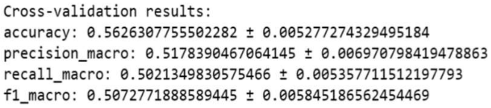
```
Accuracy: 0.6084266352386565
Precision: 0.5740854350718929
Recall: 0.5497815428699989
F1 Score: 0.5580861034326624
```

Logistic Regression



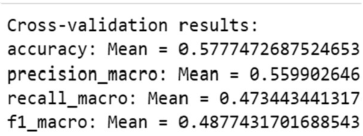
```
Accuracy: 0.5824985268120212
Precision: 0.569358134924104
Recall: 0.477585089596948
F1 Score: 0.49178943791157526
```

KNN- Cross- Validation



```
Cross-validation results:
accuracy: 0.5626307755502282 ± 0.005277274329495184
precision_macro: 0.5178390467064145 ± 0.006970798419478863
recall_macro: 0.5021349830575466 ± 0.005357711512197793
f1_macro: 0.5072771888589445 ± 0.005845186562454469
```

Logistic- Cross- Validation

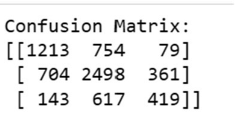


```
Cross-validation results:
accuracy: Mean = 0.5777472687524653
precision_macro: Mean = 0.559902646
recall_macro: Mean = 0.473443441317
f1_macro: Mean = 0.4877431701688543
```

Table 4 comparison model performance

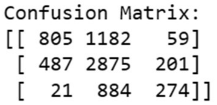
Confusion Matrix is evaluate the performance that represents how many predictions are correct and incorrect per class. The following matrices were shown as followed:(Table 5)

- K-NN



```
Confusion Matrix:
[[1213  754   79]
 [ 704 2498  361]
 [ 143  617  419]]
```

Logistic Regression



```
Confusion Matrix:
[[ 805 1182   59]
 [ 487 2875  201]
 [  21  884  274]]
```

Table 5 Confusion Matrix

The credit score has 3 classes: Class 1 = Good credit, Class 2 = Standard Credit, Class 3 = Poor Credit

The confusion matrix for KNN performs that class 2 has the highest number of correct predictions, while class 3 has the least. For Logistic Regression confusion matrix shows a strong prediction performance for Class 2 with a high number of correct predictions. To sum up, the KNN model shows balance prediction capabilities across the classed, with class 2 being predicted the highest accuracy and perform better than Logistic Regression.

VI. CONCLUSION

The objectives of this paper are data exploration, data preprocessing, and predicting model performance. We found that this dataset has imbalanced data and is erroneous. After we preprocessed the dataset and predicted model performance with KNN and logistic regression, the result in F1score and accuracy was at a moderate level, approximately 49%–60%. Although we use the cross-validation technique to test the effectiveness of models, the results demonstrate that the KNN model performed better than the logistic regression model with both the F1 score, accuracy, and confusion matrix.

The challenge of this paper is that we have to work towards improving the accuracy model. Optimal model selection is a factor to be considered because different model algorithms have different strengths and weaknesses, such as Decision trees, Support Vector Machines (SVM), and Random forests.

The limitation of this dataset is that there are imbalanced classes. The model may be biased, leading to poor performance, and the evaluation technique can be misled.

In further study, we can employ deep learning and genetic algorithms to perform more accurately and effectively and apply the synthetic minority oversampling technique (SMOTE) or the random undersampling technique to tackle the imbalanced dataset. In addition, applying the ensemble classification approaches with consideration of feature selection may be the best classification accuracy with the datasets.

REFERENCES

[1] "Credit Score Prediction using Genetic Algorithms: LSTM Technique" Juliana Adisa, Samuel Ojo, Pius Owolawi, and Agnieta Pretorius 2022

URL: https://ieeexplore-ieee- org.ezproxy.library.qmul.ac.uk/stamp/stamp.jsp?tp=&arnumber=9744714

[2] "A Decision Trees Classifier-Based Ensemble Approach to Credit Score Classification" Ashok Maurya and Shivam Gaur -2023

URL: https://ieeexplore-ieee- org.ezproxy.library.qmul.ac.uk/stamp/stamp.jsp?tp=&arnumber=10425039

[3] "Credit Risk Assessment" Danish Shaikh and Aakash Vishwakarma-2023

URL: https://ieeexplore-ieee-org.ezproxy.library.qmul.ac.uk/stamp/stamp.jsp?tp=&arnumber=10393778

[4] 'An Augmentation of Credit Card Fraud Detection Using Random Undersampling' by Vipin Khattri and Sandeep Kumar Nayak-2021

URL: https://eds-p-ebscohostcom.ezproxy.library.qmul.ac.uk/eds/detail/detail?vid=0&sid=b28fb00f302147ec84962b0385412d1d%40redis&bdata=JnNpdGU9

[5] Credit Risk Analysis using LightGBM and a comparative study of popular algorithms" Dr. J. Godwin Ponsam and Dr. S. Karpaselvi.-2021

URL: https://ieeexplore-ieee-org.ezproxy.library.qmul.ac.uk/stamp/stamp.jsp?tp=&arnumber=9711896