**Loan Approval Prediction**

Nibin Joseph

ID: 0282482

Email: cbu22cfdv@cbu.ca

Data Source: Kaggle

Link to data: <https://www.kaggle.com/competitions/playground-series-s4e10/data?select=test.csv>

**Table of Contents**

1. Introduction --------------------------------------------------------------------------------------------3

2. Literature Review -------------------------------------------------------------------------------------7

3. Methodology -------------------------------------------------------------------------------------------9

4. Result --------------------------------------------------------------------------------------------------18

5. Discussion ---------------------------------------------------------------------------------------------20

6. Conclusion --------------------------------------------------------------------------------------------22

7. References ---------------------------------------------------------------------------------------------23

**Introduction**

Today, banks are the engine of economic growth as they lend money to people and companies (Lamoreaux, N. R., 1996). They are used to get any kind of project funded ranging from buying a house or opening a business to getting an education or undertaking a massive undertaking. Banks extend credit, not only allowing individuals and businesses to grow but also making the economy work (Turner, A., 2010). But granting a loan application is more than one thing, and the applicant’s creditworthiness and finances are largely evaluated. Currently, manual and subjective procedures of loan approval is often the traditional way of getting approved which is costly and error prone.

Machine learning is growing by leaps and bounds in its recent days in providing solutions to prediction problems. With predictive analytics, banks can analyze vast amounts of data to make more accurate and efficient loan approval decisions. The machine learning models such as logistic regression, random forests and neural networks can analyze historical loan applicants for patterns and predict loan repayment probabilities. This is big step towards automation and refining the process for the loan approval which benefits both borrowers and lender (Thekkethil, M. S., Shukla, V. K., Beena, F., & Chopra, A., 2021).

The objective of this project is to develop a machine learning model that will help predict the loan approval status for a customer loan application. The dataset taken is of one of the competition datasets from Kaggle.com. The data contains total of 97742 rows and 13 dimensions which is split into two different datasets as training dataset and test dataset. Training dataset has 58645 rows and test has 13 columns. The project will follow PACE framework. PACE stands for Plan, Analyze, Construct and Execute.

A diagram of a diagram

Description automatically generated

Image source: Google Advanced Data Analytics Professional Certificate from Coursera

PACE framework allows to conduct the project work through a systematic way with clear set goals and objectives. All objectives of the project are listed in the PACE framework and its associated PACE stage as below

|  |  |  |
| --- | --- | --- |
| **Milestones** | **Tasks** | **PACE stages** |
| 1 | Define the problem | Plan |
| 2 | Data exploration and data cleaning | Plan, Analyze |
| 3 | Determine which models are most appropriate | Analyze, Construct |
| 4 | Construct the model | Construct |
| 5 | Confirm model assumptions | Analyze, Construct |
| 6 | Evaluate model results | Analyze |
| 7 | Interpret results and share actionable steps with stakeholders | Execute |

**Defining the problem**

The objective is to develop a machine learning model that will help predict the loan approval status of different customer loan applications.

The data contains total of 97742 rows and 13 dimensions/features. The dataset has two parts. Train and Test sets. Training dataset has 58645 rows and 13 dimensions. Each of these dimensions are listed below with its brief description

* **id**: Unique identifier for each record.
* **person\_age**: Age of the individual in years
* **person\_income**: Income of the individual in dollars
* **person\_home\_ownership**: Homeownership status, which includes categories like 'RENT', 'OWN', etc.
* **person\_emp\_length**: Employment length of the individual in years.
* **loan\_intent**: The purpose of the loan, with categories such as 'EDUCATION', 'MEDICAL', etc.
* **loan\_grade**: The credit grade of the loan, such as 'A', 'B', etc.
* **loan\_amnt**: Loan amount applied
* **loan\_int\_rate**: Loan interest rate
* **loan\_percent\_income**: Percentage of the individual’s income that the loan represents
* **cb\_person\_default\_on\_file**: Whether the person has a history of loan default, with values 'true' or 'false'
* **cb\_person\_cred\_hist\_length**: Length of the individual’s credit history
* **loan\_status**: Status of the application whether the loan is approved or not.

A snapshot of all columns, its datatype are in the exhibits (figure 1) which shows that out of 13 features, 9 of them are numerical features and 4 of them are categorical values including the target variable. The target variable, ‘loan\_status’ is either a 0 for loan not approved or 1 for approved for each customer loan application. A summary statistics table of all the numeric features are below, which is a great way to start understanding the data and its distribution.

The summary statistics (figure 2, exhibits) provide much more clarity about all the numerical data and its distribution. Primary reading shows some indication of outliers in the ‘person\_age’ and ‘person\_emp\_length’ variables as the max value for both is 123 years which are unusual values considering the nature of these two variables. This point will be checked thoroughly in the next phase of data cleaning process

**Literature Review**

Literature review has been conducted on the previous works related with predicting customer loan application using various machine learning models. Many scholars have examined the use of machine learning models in the related domains. Following is a brief analysis on those literatures reviewed for the purpose of my study.

Sheikh, M. A., Goel, A. K., and Kumar, T. used logistic regression model to predict loan defaulters. The model was trained and tested on the data collected from Kaggle. Authors used sensitivity and specificity as the model evaluation parameters. The model was trained using a data set of 1500 cases with 10 numerical and 8 categorical attributes. They found that the model is marginally offering better, and it has incorporated variables such age of the customer, purpose of the loan, credit history, credit amount and credit duration. Authors states that not considering other important variables regarding to customer account specific information is a limitation to the study. The study concluded that banks should not solely target affluent customers but should also consider other personal attributes that significantly impact credit decisions. By doing so, banks can effectively identify the right customers for loan approval and manage the risk of non-payment towards the loan.

The study on ‘Loan Approval Prediction Using Machine Learning’ by Diwate, Y., Rana, P., Chavan, P. was based on the data from different financial institutions with past records of loan advances passed to individuals and analysed its performances on its repayment history to develop the machine learning model. Authors considered data features like CIBIL score (score by a credit rating agency in India), business value, Assets of customer etc. The data modeling was done using SVM (Support Vector Machine) model and the precision score is used as the model evaluation criteria. The model produced a precision score of 81%. Authors identified that high credit score and high salary has most impact on the positive prediction on the data.

The paper by G. Arutjothi, Dr. C. Senthamarai is situated around predicting the credit defaulters. The authors used KNN machine learning model to predict the credit defaulters and the experiment was run using R- studio. The data collected was cleaned and pre-processed before the model building and they have used Min-Max scaler to transform the categorical data features into numerical values as the model they used works best on the numerical data values. They have chose the model with highest accuracy as the champion model.

The paper ‘Overdue Prediction of Bank Loans Based on LSTM-SVM’ by Li, X., Long, X., Sun, G., Li, H. is based on the DNN (Deep Neural Network) and the authors proposed to analyse the dynamic behaviour of users by using LSTM algorithm and SVM (Support Vector Machines). The study was conducted using customers basic information, bank records, user browsing behaviour, credit card billing records, and loan time information to evaluate whether users will be a defaulter. The final model results is built by calculating the average of both SVM and LSTM outputs

Dosalwar, S,. Kinkar, K., Sannat, R., Pise, N. states that the bank’s Non-Performing Assets can be reduced by forecasting loan defaulters. The dataset for the study was collected from Kaggle. Logistic Regression, Decision Tree Classifier, K Neighbors Classifier, Naïve Bayes, Random Forest Classifier, Support Vector Machine, and XGBoost Classifier models are trained on the data to select the final model with best result.. Authors used sensitivity and specificity as the model evaluation criteria. And the Logistic Regression model emerged as the champion model with 78.5% accuracy. The model was trained on data features like customer age, objective of the loan, credit score, credit amount, and credit period etc.

**Methodology**

**Data Exploration and Data Cleaning**

Data Cleaning involved three steps,

1. Checking for missing values in the dataset
2. Checking for duplicate rows in the dataset
3. Checking for outliers in the columns

As seen in the figure 3, exhibits, no column had any missing values to deal with. Data has been checked for duplicated rows using duplicated() from pandas which confirmed that there are no duplicate rows.

In the preliminary data exploration with summary statistics, it was found with couple of unusual values in the ‘person\_age’ and ‘person\_emp\_length’ columns. Hence need to check these two columns in details for outliers. Here, I am using box-plot diagram from seaborn library to check the data distribution with these two variables.

A screenshot of a computer screen

Description automatically generated

As it seemed in the summary statistics, both the columns have outliers which will be removed from the dataset as some of the classification models like logistic regression are sensitive to outliers. To remove these outliers, Boolean masking method is used.

**Model Selection and Model Building**

The objective of this project is to build a machine learning model to successfully predict the approval of customer loan applications. It is basically a binary classification. There are many supervised learning techniques to choose in this scenario like, logistic regression, Naïve Bayes classifier, KNN classifier, Decision Tree classifier, Random Forest, Gradient Boosting algorithms etc. In this project, three models will be compared with its results. The models chosen are, logistic regression, Random Forest and XGBoost. Logistic Regression model will be used as a base model as it’s simple in nature and can be used to compare the results with other models to check the improvements. Second model will be Random Forest, which is an ensemble of decision trees. This model is based on the decision tree, but it trains on bootstrapped data with randomly selected features which can produce better results than a decision tree. Random forest model reduces variances or in other words it reduces overfitting problem and faster runtime and scales well on big data. And the third model will be XGBoost. Its an advanced model in machine learning based on gradient boosting technique. It is one of the most widely used model because of its high accuracy. It requires minimal preprocessing of data and works well even with outliers and missing data.

**Checking Class Balance**

Before start building the model, its important to check the class balance of the target variable ‘loan\_status’. Class balance refers to the ratio in which each class of target variable is present in the dataset. The result showed that 85% of data are of loan rejected status class and 15% data is of loan approved status. It is a common scenario in like loan approval datasets that more loan applications will get rejected compared to approved. Even though it’s not a perfect class balance with 85% to 15% but it is not the worst. So, we are not doing any up-scaling or down-scaling of the target variable in this case.

**Converting all categorical values into numerical values for modeling**

Machine learning model works best on numerical values and hence the categorical variables ‘person\_home\_ownership’, ‘loan\_intent’, ‘loan\_grade’, ‘cb\_person\_default\_on\_file’ should be converted into numerical values. Here the dummy encoding process is used with the help of pd.get\_dummies().

**Splitting data into target variable and predictor variables**

To build the machine learning model, the dataset is split into two set, one with the target variable y and the other is the predictor variables X. Column ‘id’ has been excluded in the X dataset as it’s no longer needed for modeling.

Further, the datasets have been split into train and test sets with 70% to 30% ratio as using the train\_test\_split() from sklearn.model\_selection package with a test size of 30% of total data. An additional parameter random\_state=42 is used while split which will be useful to regenerate the same sample in future. Also, the data has been stratified based on the class balance in the target variable to improve its accuracy of prediction.

Two instances of logistic regression have been performed. one without scaling the data. After the result without scaling, the second logistic regression learning has been performed on the scaled data which in-turn improved the result. StandardScaler() from sklearn.preprocessing were used to perform the modeling. The accuracy has been improved from 82% to 91%, precision increased from 69% to 77%, recall from 30% to 54% and f1 from 42% to 64%. A pipeline from sklearn.pipeline is used to run a series of operation of StandarScaler and LogisticRegression models

The results are given below compared in a table and with the confusion matrix plot(see figure 4, exhibits) for the model results.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Machine Learning Model** | **Accuracy** | **Precision** | **Recall** | **f1** |
| Logistic Regression without Scaling | 0.882 | 0.699 | 0.309 | 0.429 |
| **Logistic Regression with Scaling** | **0.912** | **0.773** | **0.546** | **0.64** |

Table 1: Comparison of logistic regression before scaling and after scaling

After the logistic regression, Random Forest model has been built with cross validation and hyperparameter tuning. Random forest model’s base learners are decision trees. The tuned parameters are given below

cv\_params = {'max\_depth': [5, None],

'max\_features': [0.3, 0.6],

'max\_samples': [0.7],

'n\_estimators': [75, 100, 125, 150],

}

GridSearchCV() from sklearn.model\_selection has been used to run the cross validation with ‘precision’ as the refit parameter. Cross validation has also been performed with 5 folds as the parameter for cv argument. The result is quite an improvement from the logistic regression results. Accuracy has been increased to 95% meaning that 95% of the total predictions are accurate. Precision score is increased from 77% to 92% which means, out of total positive predictions, 92% are actual positives rest 8% are false positives. Recall score has been increased from 55% to 72%. Recall score gives the model’s ability to correctly identify the actual positives. f1 score is improved from 64% to 81%. This score is a harmonic mean of both precision and recall scores (see figure 5, exhibits for the confusion matrix plot of the Random Forest model)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Machine Learning Model** | **Accuracy** | **Precision** | **Recall** | **f1** |
| Logistic Regression without Scaling | 0.882 | 0.699 | 0.309 | 0.429 |
| Logistic Regression with Scaling | 0.912 | 0.773 | 0.546 | 0.64 |
| **Random Forest** | **0.952** | **0.921** | **0.723** | **0.81** |

Table 2: Comparison of logistic regression results with Random Forest Model results

Next, the XGBoost model is built. XGBoost stands for extra gradient boosting, which is one of the most widely used model in supervised learning methods. Many of the prize winning models in Kaggle competitions are gradient boosting machines. XGBClassifier from xgboost library is used train the model. Hyperparameter tuning is done with following parameters

{'max\_depth': [7,8,9,None],

'min\_child\_weight': [3, 5],

'learning\_rate': [0.01, 0.1],

'n\_estimators': [100,150,175]

}

Model also performed with cross validation of 5 folds. The confusion matrix and model results are shown below. Accuracy is same as random forest model with 95%, precision score is decreased by 0.4%, recall score has shown an improvement from random forest model by 0.8% and the f1 score is almost the same as compared in the table below (see figure 6, exhibit for the confusion matrix plot of XGBoost model).

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Machine Learning Model** | **Accuracy** | **Precision** | **Recall** | **f1** |
| Logistic Regression without Scaling | 0.882 | 0.699 | 0.309 | 0.429 |
| Logistic Regression with Scaling | 0.912 | 0.773 | 0.546 | 0.64 |
| Random Forest | 0.952 | 0.921 | 0.723 | 0.81 |
| **XGBoost** | **0.953** | **0.917** | **0.735** | **0.816** |

Table 3: Comparison of previous results with xgboost model results

Further, both the Random Forest model and XGBoost models are pickled using the pickle module. “Python [pickle module](https://www.geeksforgeeks.org/pickle-python-object-serialization/) is used for serializing and de-serializing a Python object structure. Any object in Python can be pickled so that it can be saved on disk. What Pickle does is it “serializes” the object first before writing it to a file” (Python Pickle — Python object serialization, Geeks for Geeks).

**Result**

After running 3 different machine learning models with different hyperparameters and cross-validation, Random Forest model is selected as the champion model. Both Random Forest and xgboost models have given close scores with random forest model has better precision score whereas the xgboost model has better recall score. But since the problem here is to predict the loan approval of customer’s loan application, the precision score has more meaning and importance. Because high precision score indicates that when the model predicts the positive class, it is most often an actual positive one. The table comparing the evaluation metrics of all four models are given below.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Machine Learning Model** | **Accuracy** | **Precision** | **Recall** | **f1** |
| Logistic Regression without Scaling | 0.882 | 0.699 | 0.309 | 0.429 |
| Logistic Regression with Scaling | 0.912 | 0.773 | 0.546 | 0.64 |
| **Random Forest** | **0.952** | **0.921** | **0.723** | **0.81** |
| XGBoost | 0.953 | 0.917 | 0.735 | 0.816 |

Table 4 : Comparison of all four models

A yellow and blue squares with numbers

Description automatically generated with medium confidence

Confusion Matrix plotted for the Champion Model (Random Forest)

Here in this problem, predicting positive class means predicting the customers who are likely to be approved for the loan. Here the cost of approving a loan application wrongly which supposed to be rejected is high compared to the cost of rejecting a loan application which is supposed to be approved. That is the reason the precision score is considered as the most important factor to select the champion model among the two best performing models.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Machine Learning Model** | **Accuracy** | **Precision** | **Recall** | **f1** |
| **Random Forest** | **0.952** | **0.921** | **0.723** | **0.81** |

Table 5: Random Forest Model’s metrics results

The champion model, in this case Random Forest model has an overall accuracy of 95% meaning that out of 100 predictions, the model can predict 95 of them correctly as either true positive class or true negative class. Precision score measures the accuracy of the positive predictions made by the model. The precision score of the champion model is 92% which means that out of 100 predictions as positive class, 92 of them are actual positive and rest 8 are falsely identified as positives. Recall score also known as sensitivity measures the model’s ability to correctly identify all positive instances. It is the ratio of true positive predictions to the total actual positives. Here, model’s recall score is 72% which means, out of 100 actual positive class, 72 are identified as positive and rest 28 are identified falsely as negative class. The f1 score is a harmonic mean of both precision and recall scores. It is useful in certain cases where, predicting both positive class and negative class is equally important. Here the model has an f1 score of 81% which is a good score, and it underlines the fact that the model is stable and reliable for predicting both positive class and negative class.

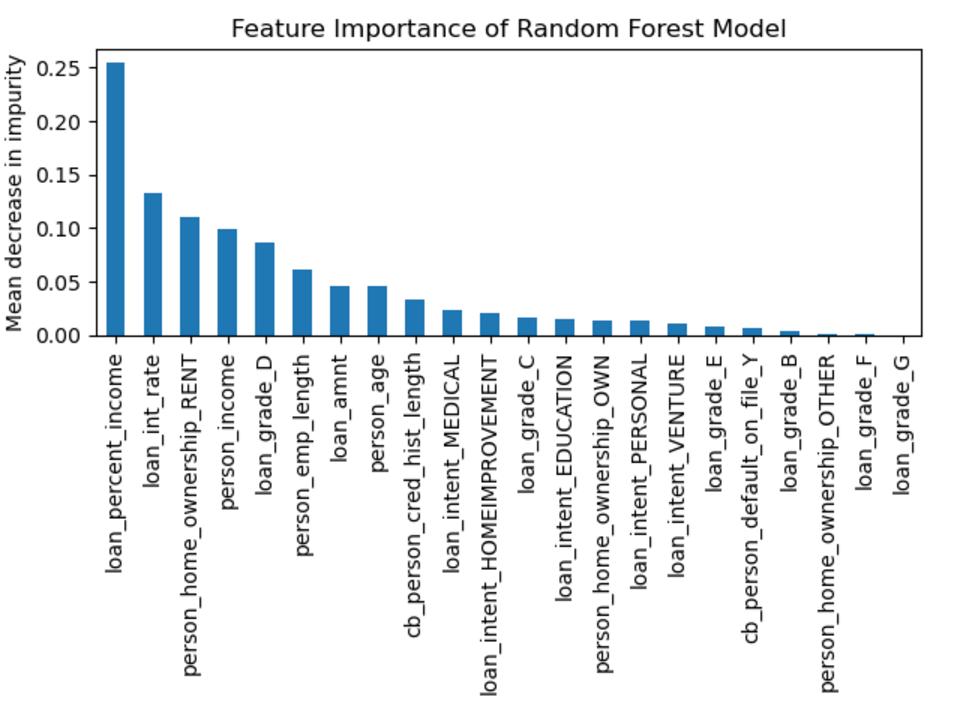


Figure 9: Random Forest Model’s metrics results

The feature importance plot has been plotted using feature\_importances\_ attribute of the model object of random forest model as given above. The top 6 features contributing to the prediction model are percentage of loan amount to the applicant’s income, interest rate of the loan, whether the applicant own their house, applicant’s income, grade of the loan type and how long has the applicant been employed for. Almost 70 percentage of the model’s prediction is dependent on these six features of the data.

**Discussion**

The comparison among the results from different models shows that the Random Forest model modified with hyperparameters set to {'max\_depth': None, 'max\_features': 0.6, 'max\_samples': 0.7, 'n\_estimators': 100}.

The context of the problem required to prioritize the precision score as the primary model evaluation metric as the cost of approving a loan that should be rejected (false positive) is comparatively higher than the cost of rejecting a loan application that should be approved (false negative). This will help the financial institution to minimizes the risk of potential defaults and get the most approved loans get repaid.

The Random Forest model has achieved a high precision score of 92% which is an indication of high accuracy in predicting the positive class of customers whose loan should be approved. This result is aligned with the financial institutions objective of minimizing the approval of risky loans.

Random Forest model’s feature\_importances\_ attribute has also identified that the factors such as loan to income ratio, interest rate, home ownership status, income of the applicant, loan grade and the length of the employment of the applicant have significant impact on the model’s predictions as shown in the figure below. This is a great insight that the loan officers and financial analysts who are the decision makers in this case can depend on making informed decisions.

A graph with blue and black text

Description automatically generated

Figure 10: Top 6 features contributing the model prediction

One limitation with the champion model’s result is that the recall score is 72% which is relatively a low score meaning that the model does miss some of the positive cases. However, as we are better interested on the precision score over recall score, the trade-off is justifiable. The class balance of loan approved cases and loan rejected cases is 15% to 85% which is considerably not the best class balance but considering the number of positive cases were around 8400 applications, the model did have enough cases to predict the positive class.

Future improvements could include getting more data for positive classes or using some over-sampling techniques to synthetically generate a greater number of positive class data points. Also, can explore other ensemble methods which can combine different algorithms unlike Random Forest model which is an ensemble of just the decision tree algorithm. More relevant features can also be added to the dataset to further improve the results. Continuous update to the model is a necessary thing with relevant new data and regular retraining can help the model’s performance over time.

**Conclusion**

The study included building three different machine learning models to predict the loan applications such as Logistic Regression (with non-scaled data and scaled data), Random Forest and XGBoost among which Random Forest model with hyperparameter tuned emerged as the champion model with high precision score. Even though the recall score indicated that there is room for improvement to predict the positive class with better accuracy, the precision score focused approach is aligned with the project’s primary objective of ensuring approved loans are likely to be repaid.

The study highlights the importance of using the right evaluation metric based on the specific context and objectives of the problem. It also underlines significance of feature importance analysis to identify the key drivers of the predictive model

Even though the model has given successful results, the model also has a few challenges like the potential changes in market conditions may have an impact on the model’s accuracy to predict results accurately over time. It is essential to add regular updates and re-evaluations to maintain the model effective and useful to deliver its purpose. It is also essential to identify and add additional features and better algorithms to further improve the accuracy of the prediction.

No model is perfect and requires continuous revisit and revalidation to maintain the effectiveness. Over-fitting can be a challenge overtime as the model may work better on the training data but it may fail to give accurate results on the new test data sets or data with different values which are out of the range of values of the training data.

**References**

A Review On Loan Approval Prediction Based On Machine Learning Approaches. (2024).

*Journal of Computer and Theoretical Applications (JCTA),* 12(1), 45-60. Retrieved

from https://journals.stmjournals.com/jocta/article=2024/view=148406/

Accurate loan approval prediction based on machine learning Approach (2020).

*Journal of Emerging Technologies in Advanced Computing (JETAC),* 10(4), 1-10.

[https://jespublication.com/upload/2020-110471.pdf](https://jespublication.com/upload/2020-110471.pdf?form=MG0AV3)

Adieltheanalyst. (n.d.). *Loan-Approval-Prediction: A machine learning project predicting loan*

*approval based on customer features like income, loan grade, and credit history.*

<https://github.com/Adieltheanalyst/Loan-Approval-Prediction>

Arutjothi, G., & Senthamarai, C. (2017). Prediction of loan status in commercial bank using

machine learning classifier. In *2017 international conference on intelligent sustainable*

*systems (ICISS)* (pp. 416-419). IEEE.

Diwate, Y., Rana, P., & Chavan, P. (2021). Loan approval prediction using machine learning.

*International Research Journal of Engineering and Technology (IRJET)*, *8*(05), 1741-

1745.

Fawcett, T. (2006). *An introduction to ROC analysis* (Vol. 14).

Springer Science & Business Media.

Garcia, R., Liu, E., Shetty, N., Dandamudi, A., Gonzalez, J. E., & Hellerstein, J. M. Choose

What to Log, After You Execute.

Lamoreaux, N. R. (1996). *Insider lending: Banks, personal connections, and economic*

*development in industrial New England*. Cambridge University Press.

Li, X., Long, X., Sun, G., Yang, G., & Li, H. (2018). Overdue prediction of bank loans based

on LSTM-SVM. In *2018 IEEE SmartWorld, Ubiquitous Intelligence & Computing,*

*Advanced & Trusted Computing, Scalable Computing & Communications, Cloud &*

*Big Data Computing, Internet of People and Smart City Innovation*

*(SmartWorld/SCALCOM/UIC/ATC/CBDCom/IOP/SCI)* (pp. 1859-1863). IEEE.

Patel, B., Patil, H., Hembram, J., & Jaswal, S. (2020). Loan default forecasting using data

mining. In *2020 international conference for emerging technology (INCET)* (pp. 1-4). IEEE.

Powers, D. M. (2011). *Evaluation: From precision, recall and F-measure to ROC,*

*informedness, markedness and correlation*. Journal of Machine Learning Technologies,

2(1), 37-63.

Sharayu Dosalwar, Ketki Kinkar, Rahul Sannat, Dr Nitin Pise4 (2021). Analysis of Loan

Availability using Machine Learning Techniques

Sheikh, M. A., Goel, A. K., & Kumar, T. (2020, July). An approach for prediction of loan

approval using machine learning algorithm. In *2020 international conference on*

*electronics and sustainable communication systems (ICESC)* (pp. 490-494). IEEE.

Thekkethil, M. S., Shukla, V. K., Beena, F., & Chopra, A. (2021). Robotic process automation

in banking and finance sector for loan processing and fraud detection.

In *2021 9th international conference on reliability, infocom technologies and*

*optimization (trends and future directions)(ICRITO)* (pp. 1-6). IEEE.

Turner, A. (2010). What do banks do? Why do credit booms and busts occur and what can

public policy do about it. *The Future of finance*, *5*.

Yajasarora. (n.d.). *Loan Prediction with Machine Learning.*

<https://github.com/yajasarora/Loan-Prediction-with-Machine-Learning>

Python Pickle — Python object serialization

https://www.geeksforgeeks.org/understanding-python-pickling-example/

**Exhibits**

A screenshot of a computer

Description automatically generated

Figure 1: Basic details about the data features

A screenshot of a computer

Description automatically generated

Figure 2: Summary statistics of numerical features

A screenshot of a computer code

Description automatically generated

Figure 3: Checking for missing values

A screenshot of a computer program

Description automatically generated

Figure 4: Code executed for Logistic Regression with Scaling using pipeline

A screenshot of a computer screen

Description automatically generated

Figure 5: Confusion Matrix for the Logistic Regression model with scaled data

A screenshot of a computer program

Description automatically generated

Figure 6: Code Executed for Random Forest model training with cross validation and hyperparameter tuning

A yellow and blue squares with numbers

Description automatically generated with medium confidence

Figure 7: Confusion Matrix for the Random Forest model with scaled data

A chart of a test set

Description automatically generated with medium confidence

Figure 8: Confusion Matrix for the XGBoost model with scaled data