Professor Mingdi Xin

BANA 273 Machine Learning for Analytics

30 November 2021

Machine Learning Final Project Report   
Classifying Credible Loan Applicants

Team 23:

Kate Zagrebneva

Prashasti Sharma

My Han Mai

Noor Zia

Samira Shafiei

**Table of Contents**

Executive Summary

Business Idea

Data Summary

Selected Machine Learning Techniques

Naiive Bayes

Decision Trees

Logistic Regressions

Process of Analysis

Results of Analysis

Appendix

Executive Summary

## Business Idea

## Distribution of the loans is important for many. For example, the main portion of the bank’s assets comes directly from the profit earned from the loans it distributes.The primary objective in the many financial environments is to invest their assets in situations where they can be assured of profit or being paid back. Today many financial companies such as banks, mortgage providers, loan companies, et cetera approve loans after an aggressive process of verification and validation but still there is no surety whether the chosen applicant is the most dependable applicant out of all applicants. Through this project we hope to make more accurate predictions on whether or not a particular applicant is credible or not through the validation of attributes by applying machine learning techniques. Loan Prediction is very integral to the activities of lenders and also greatly impacts applicants.

## The aim of this project is to provide a quick, immediate and easy way to identify the applicants who are most likely to be approved. It can provide many benefits to the lenders and applicants who are looking to be approved. The Loan Prediction System can automatically calculate the weight of each feature taking part in loan processing and on new test data the same features are processed with respect to their associated weight. It can allow jumping to specific applications so that it can be checked on priority basis. This replacement results against particular Loan Id can also be sent to various departments of banks/firms or households so that they can take appropriate action on application.

## Data Summary

The data we are using for this project is pulled from Vikas Ukani’s ‘Loan Eligible Dataset’ off of the free datasets provided on Kaggle. The dataset consists of two parts: loan-test.csv and loan-train.csv. The dataset is based off of a fictitious loan company, Dream House Finance company, which offers home loans for urban, semi-urban, and rural areas. The data is collected from the company’s home loan applications in order to verify the applicant’s eligibility to be approved for a loan.

The application form requests information on gender, marriage status, dependents, education level, status of employment (self-employed or not), income level, co-applicant income level, amount of loan requested, length of loan, credit history of applicant, and area in which the property is located (urban, semi-urban, rural). It also provides applicants with a unique ID number that corresponds with the loan application. The loan amount is in units of thousands of US dollars and the loan length is in units of months.

Overall, the data provided is a partial list of 367 customers in the test set and 614 customers in the train set. The goal of using this dataset is to identify specific customer segments in order to predict and target those customers who are more likely to be eligible for loan approval.

Variables:

Descriptive statistics of the numerical variables within the dataset:

Summary of overall dataset:

## Observations of Data:

1. Around eighty applicants (8%) are without a credit history.
2. There are null values in the loan amount and loan term column indicating that some applicants submitted an application without completing all the required information.
3. The variable, Loan Amount, has outliers that skews the entire dataset to the right.
4. The general applicant type is a married, male graduate with no children.

## **Selected Machine Learning Techniques**

Decision Tree

One of the most important goals of the project is to determine which class a new client with specific characteristics will fall into. The provided dataset is structured and supervised. Based on different variables, two classes of interest are defined; approved and not approved. For example, we would like to predict that the loan application for a new client who is male, married, not graduate, self-employed, and with $75,000 income would be approved or not. To do so, the best Model to perform this classification is the Decision Tree classifier. Based on the different attributes, the decision Tree algorithm identifies whether the client loan should be approved or not.

One of the reasons that we chose the Decision Tree technique is that our dataset contains both categorical and continuous variables and the Decision Tree algorithm works well with both types within a dataset. Another reason is that it is easy to understand and interpret and does not require very complex and precise data. Decision Tree does not require much preprocessing to get output.

One of the challenges that we faced with the Decision Tree algorithm is selecting the right attributes. We need to choose the best attributes as the root of the different levels of the decision Tree. In order to tackle this problem, we measure the information gain and entropy for each variable. We set the variable with the highest information gain as the root of the decision Tree and split the data that gives the best classification and purer classes.

Before Preprocessing:

To run decision trees in Weka, we had to remove load id and dependents:

We used J48 and cross validation with 10 folds

***1.Raw data- remove Loan-ID, Dependents***

Tree visualization:

***2. Post Preprocessing: Removing Nulls, Loan\_ID, Converting all numeric attributes to nominal, Discretize ApplicantIncom, CoapplicantIncom, LoanAmount***

Tree visualization:

***3. Post Processing: Replacing Nulls with mean & mode, removing Loan\_ID, Converting all numeric attributes to nominal, Discretize ApplicantIncom, CoapplicantIncom, LoanAmount***

Tree visualization:

***4. Post Processing: Replacing Nulls with mean & mode, removing Loan\_ID, removing Dependents????? Keep all variables as numeric***

**Random Forest**

The Decision Tree is a simple classification model, however, in terms of prediction accuracy, it is not as robust as other models. We chose the Random Forest classifier to improve the prediction accuracy of the Decision Tree model. After growing some individual decision trees on the training dataset, the Random Forest model combines all the predictions and produces the outcome with the highest accuracy.

***1.Raw data- remove Loan-ID, Dependents***

***2. Post Processing: Removing Nulls, Loan\_ID, Converting all numeric attributes to nominal, Discretize ApplicantIncom, CoapplicantIncom, LoanAmount***

***3. Post Processing: Replacing Nulls with mean & mode, removing Loan\_ID, Converting all numeric attributes to nominal, Discretize ApplicantIncom, CoapplicantIncom, LoanAmount***

**Naive Bayes**

Naive Bayes is a probabilistic classifier and is one of the most powerful and simple classification techniques to make fast predictions. Naive Bayes performs better than other classifiers when the variables are independent. It makes the assumption that the presence of the predictor variables is unrelated to each other in the dataset. We chose this technique because the attributes in the dataset are independent of each other. To make sure we have chosen the right technique and to have a robust analysis we examined the interaction between the variables in the dataset. To do so, we analyzed the correlations between variables in Python. Also, we created a heatmap visualization in Python to be able to explain the correlation between the attributes. According to the below plot, the dataset’s attributes have either very weak correlations or almost no correlation between themselves. So, the first assumption needed to use the Naive Bayes technique is fulfilled, which is that the predictors are independent.

Moreover, the provided dataset is structured and supervised, and some of the attributes are categorical. We chose the Naive Bayes technique because it works better with categorical variables compared to numerical variables. Subsequently, we changed the numerical variables to categorical to improve the model’s accuracy

*To run naive bayes in Weka we had to remove loan id and dependents. We used cross validation with 10 folds.*

***1.Raw data- remove Loan-ID, Dependents***

***2. Post Processing: Removing Nulls, Loan\_ID, Converting all numeric attributes to nominal, Discretize ApplicantIncom, CoapplicantIncom, LoanAmount***

***3. Post Processing: Replacing Nulls with mean & mode, removing Loan\_ID, Converting all numeric attributes to nominal, Discretize ApplicantIncom, CoapplicantIncom, LoanAmount***

**Regression**

Another purpose of this project is to determine which variables have the most impact on the loan approval variable. Moreover, we want to produce probability estimates for loan approval rather than classification. We would like to find the probability of each class; the likelihood that a loan will be approved or not. To do so, we decided to use the logistic regression model. Since the provided dataset is small, the logistic regression model is best for prediction as it has better performance. To run logistic regression in Weka we had to remove LoanID and Dependents. We used ‘logistic’ and cross-validation with 10 folds:

Another reason that we chose this technique is that logistic regression is a robust model to predict the binary variable. In the dataset, the dependent variable is loan approval and it is a binary (approve: yes =1, approve: No=0). Another assumption of the logistic regression which is made by the provided dataset is that independent variables should not be correlated. As mentioned in the Naive Bayes technique, we proved that the attributes have no correlation by conducting a heatmap plot.

In order to have a more robust model and increase the overall accuracy, we will remove the variables that are not relevant. (mention which variables we removed)

***1.Raw data- remove Loan-ID, Dependents***

***2. Post Processing: Removing Nulls, Loan\_ID, Converting all numeric attributes to nominal, Discretize ApplicantIncom, CoapplicantIncom, LoanAmount***

***3. Post Processing: Replacing Nulls with mean & mode, removing Loan\_ID, Converting all numeric attributes to nominal, Discretize ApplicantIncom, CoapplicantIncom, LoanAmount***

***4. Post Processing: Replacing Nulls with mean & mode, removing Loan\_ID, Dependents***

***Converting all numeric attributes to nominal, Discretize ApplicantIncom, CoapplicantIncom, LoanAmount***

Process of Analysis

Initially, we gathered our data from kaggle. As part of the data exploration phase, we looked at the shape (rows and columns) and the descriptive summary of the unprocessed data set. We found that there our overall dataset was split into a train and test set. The train set had 614 rows with 13 columns. Other than Loan\_ID, ApplicantIncome/CopplicantIncome, Property\_area, and Loan\_status, every other column contains null values. The test set had 367 rows and 12 columns (missing the Loan\_status column). The test set was more complete than the train set, but still had a few null values in most columns. Looking at the descriptive statistics of the numeric columns, we found the values to be within reasonable range for its column.

Once we assessed our data set, we decided to focus on classification. The project goal was to use the attributes to classify whether applicants are credible or not. For this goal, we planned to use naive bayes, logistic regression, and decision tree/random forest.

Before we could run those techniques, we needed to clean and preprocess our data sets. To clean, we checked for duplicates and nulls. Fortunately, there were no duplicates, however there were many nulls so we used python to drop the null values. We also dropped two columns that were irrelevant: loan\_id and dependents. While dependents may have been a usable factor in the credibility verification process, it was preventing us from performing logistics regression so we recursively decided to eliminate that column. For null values, we tried dropping them but found that it lost too much data (171 rows) so instead we replaced categorical null data with mode and numeric null values we replaced with mean. We also found several obvious outliers. As these skewed the data of Loan Amount, ApplicantIncome, CoapplicantIncome to the right, we chose to discretize them and replace the outliers with averages in order to normalize the data.

Number of Nulls in each column:

Furthermore, given that we were running different machine learning techniques that required different data types, we had to convert the datasets into the appropriate data types. In Weka, we changed all numeric and string values into nominal to create a fully nominal dataset in order to run Naive Bayes. In Python, we converted nominal attributes into numeric by creating dummy variables. We used the fully numeric dataset in order to run logistic regression. Lastly, we kept the unprocessed (aside from cleaning duplicates and nulls) dataset to run our decision trees and random tree algorithms.

Once the datasets were cleaned and ready, we used two programs to run our selected techniques. In Weka, we used the nominal dataset to run Naive Bayes. We also used Weka to run Random Forest. In Python, we used the cleaned, original dataset to run decision trees and logistic regressions. For all these methods, we first ran the unprocessed dataset in order to have a baseline to compare our results.

In the end, we chose the model that had the best combination of accuracy and fit in order to run our test data set. Our Naive Bayes and Decision Tree models ended up having a slightly higher accuracy than our logistic regression model. Interestingly, we found that our Naive Bayes model and Decision Tree model had the same accuracy although they had different confusion matrices. However, we ultimately decided to select Naive Bayes because we wanted to caution against the tendency of Decision Trees to overfit.

## Results of Analysis

**Benchmarks before preprocessing data**

**Accuracy**

Decision Tree: 80.9446%

Naiive Bayes: 79.6417%

Logistic Regression: 80.9446%

Random Forest: 77.8502%

Since Logistic Regression and Decision Tree are the same classifier and we used nominal data before we did preprocessing for logistic regression, the accuracy for those two models are the same.

Since most of the data before preprocessing is numeric it works better with Decision Tree and Logistic Regression than with Naiive Bayes.

Random Forest has the lowest accuracy, but it can be due to the model preventing overfitting, unlike Decision Tree.

**Proportion of class variable**

Decision Tree: Y: 422, N: 192

Naive Bayes: Y: 422, N: 192

Logistic Regression: Y: 422, N: 192

Random Forest: Y: 422, N: 192

We have not done any unique preprocessing to specific models yet and, therefore, class variable proportion is the same for all models.

**Preprocessing steps**

Since we were using three different models for this project, we decided to make pre-processing steps slightly different for each model depending on the type of data that is most suited for the specific model.

**Decision Tree Preprocessing**

Firstly, we removed all duplicate rows and Load ID using Python.

Created two versions of the dataset with different preprocessing steps for null values to test the accuracy of each model:

1. Remove all rows with null values
2. Replace null values with column averages

Since WEKA can run decision tree models without normalizing data, we kept preprocessing steps for this model simple.

**Decision Tree Accuracy After Preprocessing:**

Nulls removed: 80.8333%

Since we removed about 20% of the rows, the accuracy of the model decreased due to a smaller number of cases to base the model on.

Nulls replaced: 80.9446%

Since WEKA’s decision tree model can run on data that has not been normalized, the results remained the same as before preprocessing.

**Naiive Bayes Preprocessing**

For this model we also did some basic cleaning/preprocessing steps of removing duplicates and Load ID and creating two different versions of models approaching null values the same way as in the decision tree model.

To make the data most optimal for Naive Bayes model we converted all numeric attributes to nominal using WEKA’s preprocessing filter “NumericToNominal”. Since some of the rows in the “Dependents” attribute contain strings, we converted it to nominal values as well using the “StringToNominal” filter. The last step was to Discretize ApplicantIncom, CoaaplicantIncom and LoanAmount. The reason for doing this is to normalize the distribution of values in those attributes, since the distribution was skewed.

**Naiive Bayes Accuracy After Preprocessing:**

Nulls removed: 80.625% - accuracy has increased compared to accuracy of 79.6417% before preprocessing.

Nulls replaced: 80.9446% - accuracy is the same as decision tree before preprocessing and decision tree after preprocessing and replacing nulls with averages.

**Logistic Regression Preprocessing**

The first steps were similar to the previous models where we removed duplicates and Load ID and created two different versions of models approaching null values the same way as in the decision tree model. We also created another version of the model where we converted all attributes to numeric.

Since Logistic Regression predicts binary outcome (our class attribute is a binary attribute with Y/N outcome) it works well with numeric data. We have also converted nominal attributes values into dummy variables for the logistic regression to be possible. We had to drop “Dependents” attribute, as it was preventing us from running the regression in WEKA. The preprocessing was done using Python.

**Logistic Regression Accuracy After Preprocessing:**

Nulls removed: 79.17%

Nulls replaced with nominal: 79.4788%

Nulls replaced with numeric: 80.7818%

The accuracy showed to be the best when we did in-depth preprocessing and converting values into all numeric and dropping “dependents” attribute.

**Random Forest Preprocessing**

We again did some basic cleaning/preprocessing steps of removing duplicates and Load ID and creating two different versions of models approaching null values the same way as in the previous models.

**Random Forest Accuracy After Preprocessing:**

Nulls removed: 77.7083%

Nulls replaced: 75.0814%

Even though the accuracy for this model ended up to be the lowest, it might possibly be more reliable than the decision tree model due to it selecting features randomly unlike decision trees.

**Testing different preprocessing steps for all models:**

**Missing Values**

The two different pre-processing that we used to compare the results and benchmarks of the models were focused on the nan values in the dataset. Since the rows with missing values compiled around 20% of the dataset, which already had not a large amount of data, we decided to use mean/mode of columns for missing values. To make sure that the accuracy of the models does not decrease drastically, we created models for both: the dataset with removed nulls and the dataset with averages instead of nulls to compare model accuracy.

The models that used dataset with null values replaced with column averages all performed better than the ones with removed null rows. Naive Bayes model showed an increase in accuracy compared to the model before preprocessing and Decision Tree results in the same accuracy as before preprocessing. Logistic Regression accuracy increased after doing preprocessing, and, especially after we converted all values to numeric.

Random Forest accuracy decreased after doing preprocessing. The reason for it might be that the model does not have enough attributes and rows.

**Binning**

We have also tried discretizing some of the attributes into different amounts of bins anywhere from 5 to 10. Since the dataset was not large enough and data did not vary too much, we decided to go with 6 bins and it gave us the most optimal results.

Results: ? show increase/decrease in accuracy - Need to run a model with 10 bins to show that 6 bins work better

**Choosing Best Fitting Model and Conclusion:**

Which model has the highest accuracy and works the best with the dataset that we have.

Logistic Regression or Naive Bayes?

Random forest has too low accuracy; decision tree tend to overfit

Run the chosen model on the test datatset and classify people as reliable/unreliable (the value for the class variable).

### Appendix

Visualization of Initial (Un-preprocessed)Datasets: