GraphicEra
(Deemed to be University)
Accredited by NAAC with Grade A

A PROJECT REPORT
ON

PREDICTING THE LIKELIHOOD OF E-SIGNING OF LOAN BASED ON FINANCIAL HISTORY.

Name : Dhruv Saini
University Roll Number : 2015241
Subject : Julia AI Project

Under the guidance of

MR MD FARMANUL HAQUE

ACKNOWLEDGEMENT

We are extremely thankful to our honourable President sir Prof. (Dr.) Kamal Ghanshala of GRAPHIC ERA DEEMED TO BE UNIVERSITY for providing all kind of educational and infrastructural support to work in this project, without which this project would not have been possible.

We would like to thank our head of the department, Dr Devesh Pratap Singh and all other faculties of computer science and engineering department, head of the department and faculties of other departments and all other teaching and nonteaching staff of our collage for the most effective and most valuable guidance.

We hereby like to express our sincere gratitude and respect to our project guide Mr. Md Farmanul Haque for his stimulating guidance and continuous supervision, monitoring and constant encouragement throughout the project completion. The blessing, help and guidance given by him/her time to time shall us a long way in the journey of life on which we are about to embark.

We are obliged to our project team member for the valuable information provided by then in their respective fields. We are grateful for everyone's cooperation during the period of our project assignment.

INDEX

ABSTRACT

The company seeks to leverage this model to identify fewer quality applicants (e.g., those who are not responding to the onboarding process) and experiment with giving them different onboarding screen.

The official application begins with the lead visiting into the website after they opted to acquire it. Here, the applicant starts with the onboarding process to apply for a loan. The user begins to provide more financial information by going over every screen of the onboarding process. The first phase ends with the applicant providing his/her signature indicating all of the given information is correct.

Any of the following screens, in which the applicants are approved/denied and given the terms of the loan, is dependent on the company, not the applicant. Therefore, the effectiveness of the onboarding is measured up to the moment the applicant stops having control of the application process.

We will use various machine learning classification algorithms in this project such as SVM, Decision Tree and Random Forest to achieve our objective optimally. We will also optimize our model according to the data provided.

1 Introduction

1.1 Problem Statement:

To develop a model to predict 'quality' applicants, in this case, study quality applicants are those who reach a minimum threshold part of the loan application process.

1.2 Project Description:

In this project, we are going to use various type of machine learning classification algorithms to predict 'quality' applicants, in this case, we study quality applicants are those who reach a minimum threshold part of the loan application process.

1.3 Objective:

The objective that we are going for and achieve at the completion of this project is:

- Use Case / Business Case

Step one is actually understanding the business or use case with the desired outcome. Only by understanding the final objective we can build a model that is actually of use. In our case the objective is predicting the likelihood of e-signing of loan based on financial history.

- Data collection & cleaning

With understanding the context, it is possible to identify the right data sources, cleansing the data sets and preparing for feature selection or engineering. The predicting model is only as good as the data source.

- Feature selection & engineering

With the third step we decide which features we want to include in our model and prepare the cleansed data to be used for the machine learning algorithm.

- Modelling

With the prepared data we are ready to feed our model. But to make good predictions, we firstly need to find the right model (selection) and secondly need to evaluate that the algorithm actually works.

- Insights and Actions

Last but not least we have to evaluate and interpret the outcomes.

1.4 System Requirements:

Operating System	Windows 10 / Ubuntu 15 or above
Processor	Intel i3 7th Gen or above at 1.8Ghz
RAM	4 GB DDR3 or above
GPU	Nvidia GTX 2GB +
Software Required	Julia 1.5.4 or above

2 Libraries

- **DataFrames .jl** is a popular Julia library. It provides a set of tools for working with tabular data. Its design and functionality are similar to those of pandas (in Python) and data.frame, data.table and dplyr (in R), making it a great general purpose data science tool. DataFrames.jl is a great general-purpose tool for data manipulation and wrangling.

- **CSV.jl** is another popular Julia library that provides a number of utilities for working with delimited files. It is build to be fast, flexible and to handle file read, write functions.

- **Plots.jl** is a visualization interface and toolset. It sits above other backends, like GR, PyPlot, PGFPlotsX, or Plotly, connecting commands with implementation. If one backend does not support your desired features or make the right trade-offs, you can just switch to another backend with one command. No need to change your code. No need to learn a new syntax

- **GLM .jl** is a Julia library used to create, fit and test machine learning model. To fit a Generalized Linear Model (GLM), use the function,
  - `glm(formula, data, family, link)`
  - where,
  - formula: uses column symbols for the DataFrame data.
  - data: a DataFrame which may contain NA values, any rows with NA values are ignored.
  - family: chosen from Bernoulli(), Binomial(), Gamma(), Normal(), or Poisson()
  - link: chosen from the list below, for example, LogitLink() is a valid link for the Binomial() family

- **ScikitLearn.jl** is an open-source library that implements a range of machine learning, pre-processing, cross-validation and visualization algorithms using a unified interface. Most of the functions that we are going to use here belongs to the library scikit-learn. We use different sub libraries to call various types of classes to perform all the steps that include, data cleaning, data splitting, model fitting, optimization processes.

Important features of scikit-learn:

- Simple and efficient tools for data mining and data analysis. It features various classification, regression and clustering algorithms including support vector machines, random forests, gradient boosting, k-means, etc.
- Accessible to everybody and reusable in various contexts.
- Built on the top of NumPy, SciPy, and matplotlib.
- Open source, commercially usable – BSD license.

3 Dataset Analysis

Predicting if the cancer diagnosis is benign or malignant based on several observations/features

1. 21 features are used, examples:
    - entry_id
    - age
    - pay_schedule
    - home_owner
    - months_employed
    - years_employed
    - amount_requested
    - has_debt
    - e_signed
2. Number of Instances: 17908
3. Class Distribution
    - e_signed 0 – 8269
    - e_signed 1 – 9639
4. Target class:
    - 0 and 1

3.1 EDA

1. EDA(Exploratory Data Analysis) is essentially a type of storytelling for statisticians . It allows to uncover patterns and insights , often with visual methods , within data . EDA is often the first step of the data modelling process. It is used for gaining a better understanding of data aspects like: - main features of data, -variables and relationships that holds between them, - identifying which variables are important . Various exploratory data analysis methods like:
2. Descriptive Statistics which is a way of a giving a brief overview of the dataset which is dealing with, including some measure and features of the sample DataFrames provide describe() applies basic statistical computations on the dataset like extreme values , count of data , point standard deviation etc. Describe() function gives a group picture of distribution of data.
3. Grouping data an interesting measure which figure out effect of different categorical attributes on other data variables.
4. Anova stands for Analysis of Variance. It is performed to figure out the relation between the different group of categorical data.
5. Under Anova have two measures as result:
6. Correlation and Correlation computation is a simple relationship between two variables in a context such that one variable affects the other. Correlation is different from act of causing. One way to calculate correlation among variables is to find Pearson correlation Here two parameters namely, Pearson coefficient and p-value . There is a strong correlation between two variables when Pearson coefficient is close to either 1 or -1 and the p-value is less than 0.0001.

3.2 Feature Engineering

1. Feature Engineering is the process of using domain knowledge of the data to create features that make machine learning algorithms work . Feature engineering is fundamental to the machine learning and is both difficult and expensive . The feature engineering process is
    a. Brainstorming or testing feature ;
    b. Deciding what feature to create;
    c. Creating features;
    d. Checking how the features work with the model;
    e. Improving features if needed;
    f. Go back to brainstorming / creating more features until the work is done .
    g. Preparing the proper input dataset , compatible with machine learning algorithm requirements;
    h. Improving the performance of machine learning .
2. This metric is very impressive to show the importance of feature engineering in data science .
3. Techniques are listed as: IMPUTATION , HANDLING OUTLIERS , BINNING, LOG TRANSFORM, ONE HOT ENCODER, GROUPING OPERATIONS, FEATURE SPLIT, SCALING, EXTRACTING DATE.
4. In machine learning and pattern recognition , a feature is an individual measurable property or characteristic of a phenomenon being observed . Choosing informative , discriminating and independent features is a crucial step for effective algorithms in pattern recognition , classification and regression. While feature engineering requires label times, in our general-purpose framework, it is *not hard-coded* for specific labels corresponding to only one prediction problem. If we wrote our feature engineering code for a single problem — as feature engineering is traditionally approached — then we would have to redo this laborious step every time the parameters change.
5. Instead, we use APIs like Featuretools that can build features for *any set of labels without requiring changes to the code.* This means for the customer churn dataset, we can solve multiple prediction problems — predicting churn every month, every other week, or with a lead time of two rather than one month — using the exact same feature engineering code.

3.2 Data Cleaning

- In the context of data science and machine learning, data cleaning means filtering and modifying your data such that it is easier to explore, understand, and model. Filtering out the parts you don't want or need so that you don't need to look at or process them. Modifying the parts you do need but aren't in the format you need them to be in so that you can properly use them.

3.3 One Hot Encoding

- One hot encoding is a process by which categorical variables are converted into a form that could be provided to ML algorithms to do a better job in prediction.
- We use One Hot Encoding in “pay_schedule” as they contain categorical values. We use the function : get_dummies() from the pandas library.
- We combine 'personal_account_m' and 'personal_account_y' under a single column named 'personal_account_months' and drop 'personal_account_m' and 'personal_account_y' as they are no longer needed.

3.4 Splitting the data

- The basic idea is to divide the dataset T into two subsets – one subset is used for training while the other subset is left out and the performance of the final model is evaluated on it. The main purpose of cross- validation is to achieve a stable and confident estimate of the model performance.
- The optimal ratio of splitting the data is 8:2 so we are keeping the 80% of data into the training set and the rest 20% into the testing set at random state = 0, so that we don’t have any randomization in our data
- We save the training set into X_train and y_train. We save the test set into X_test and y_test.

3.5 Feature Scaling

- Feature Scaling is a technique to standardize the independent features present in the data in a fixed range. It is performed during the data pre-processing to handle highly varying magnitudes or values or units.
- We use the class StandardScalar from the library scikit learn to implement feature scaling.

- The function fit_transform() is then used to transform all the data in a fixed range to fit it into our model. We also use the function : Dataframe() to preserve the column

4 Data Visualization

4.1 **Plot.jl** for Julia Data Visualization

It is a high-level data visualization library of Julia that enable a user to create various statistical graphs. If one backend does not support your desired features or make the right trade-offs, you can just switch to another backend with one command.

Plot Type – Histogram Plot

A histogram is a graphical display of data using bars of different heights. In a histogram, each bar groups numbers into ranges. Taller bars show that more data falls in that range. A histogram displays the shape and spread of continuous sample data.

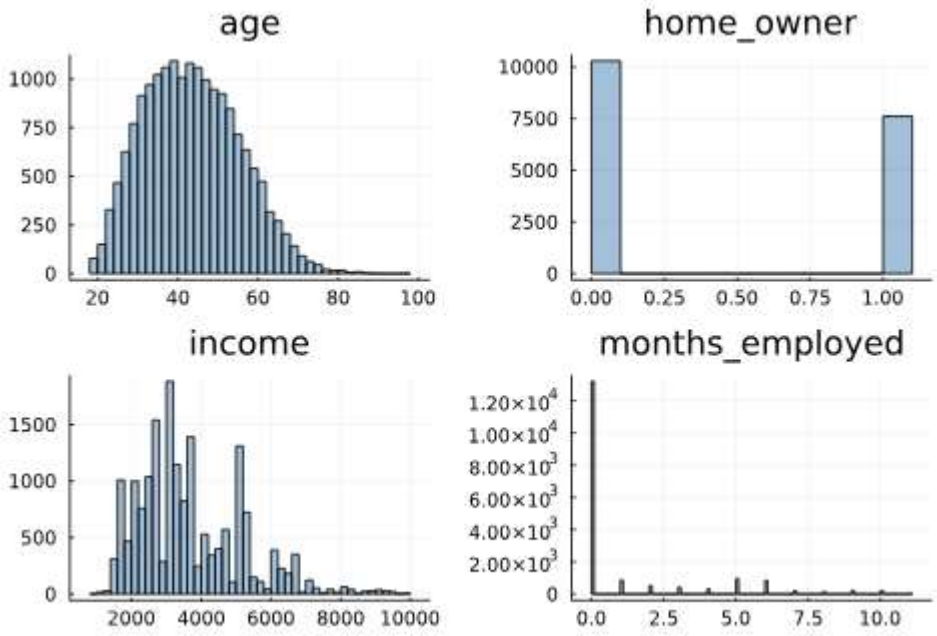


Fig: histogram of age, home_owner, income and months_employed

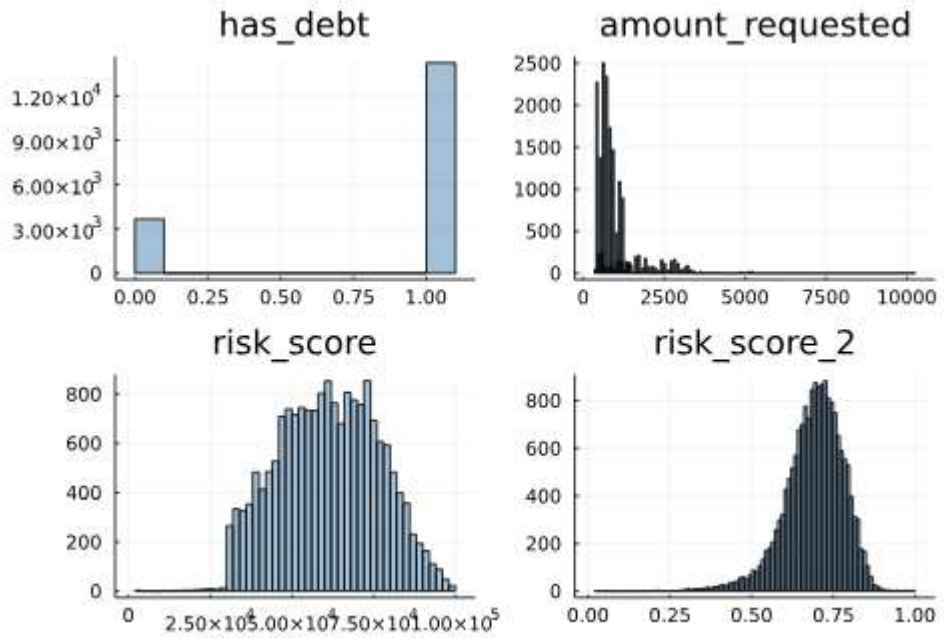


Fig: histogram of has_debt, amount_requested and risk_score and risk_score_2

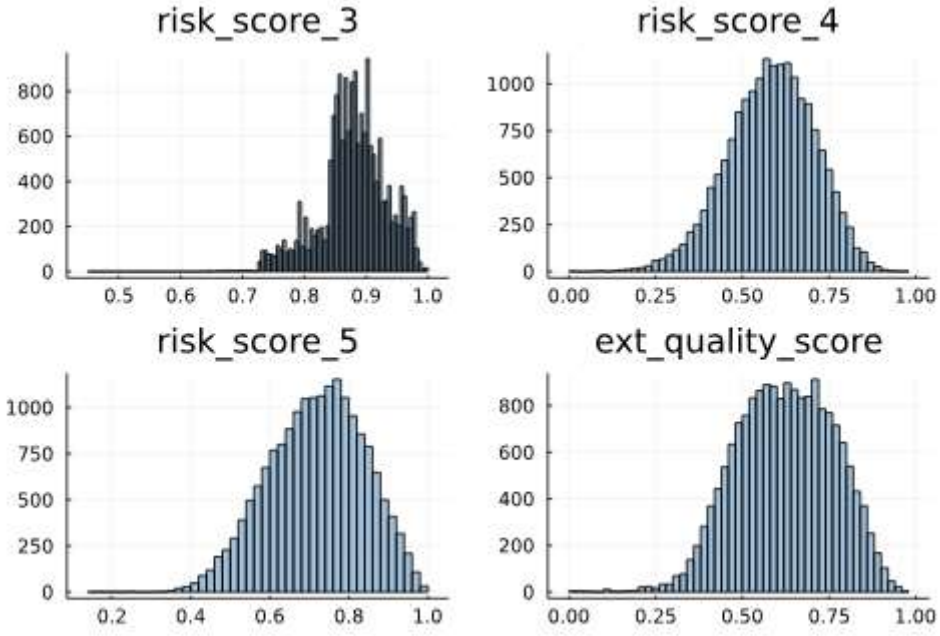


Fig: histogram of risk_score3, risk_score_4, risk_score_5 and ext_quality_score

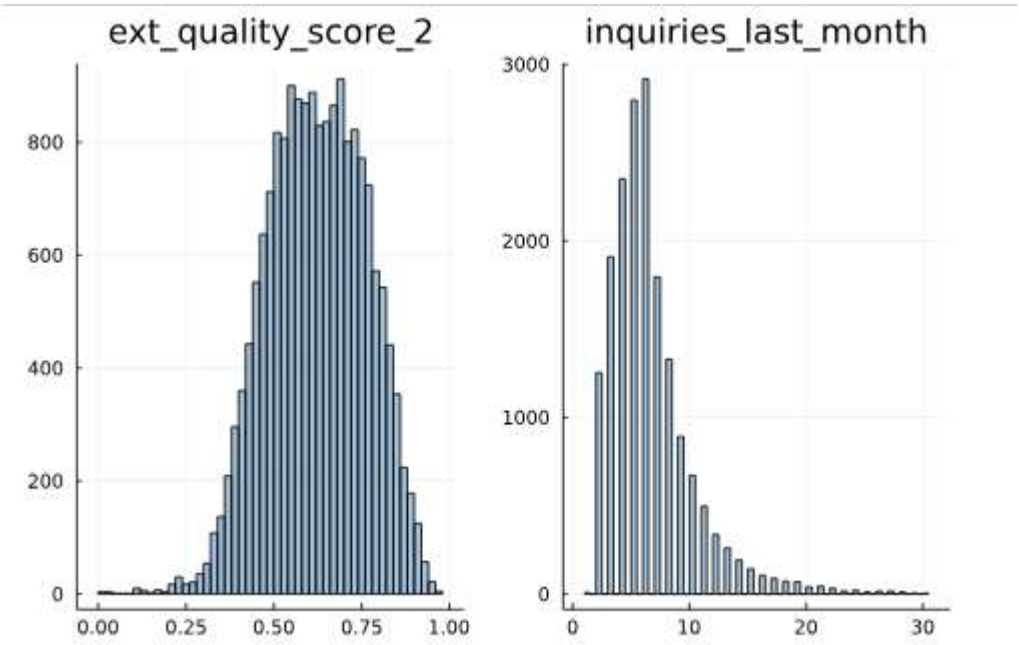


Fig: histogram of ext_quality_score_2 and inquiries_last_month

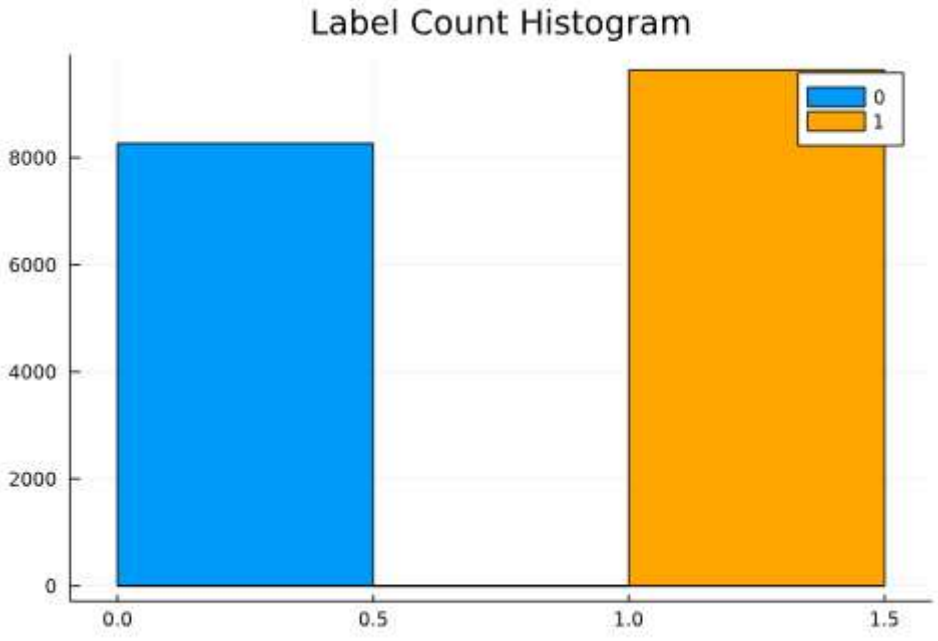


Fig: Label count histogram

5 Model Selection

5.1 Machine Learning

- Machine learning is an application of artificial intelligence (AI) that provides systems the ability to automatically learn and improve from experience without being explicitly programmed. Machine learning focuses on the development of computer programs that can access data and use it learn for themselves.
- Machine learning is closely related to computational statistics, which focuses on making predictions using computers. The study of mathematical optimization delivers methods, theory and application domains to the field of machine learning. Data mining is a field of study within machine learning, and focuses on exploratory data analysis through unsupervised learning. In its application across business problems, machine learning is also referred to as predictive analytics.

5.2 Testing the model

- We use various functions to test our model on how much time it is taking to train, how accurately it can predict the outcomes on new data. For this we use various functions that are :

  - **accuracy_score( )** - In multilabel classification, this function computes subset
  - **f1_score( )** - Compute the F1 score, also known as balanced F-score or F-measure. The F1 score can be interpreted as a weighted average of the precision and recall, where an F1 score reaches its best value at 1 and worst score at 0. The relative contribution of precision and recall to the F1 score are equal. The formula for the F1 score is: F1 = 2 * (precision * recall) / (precision + recall)

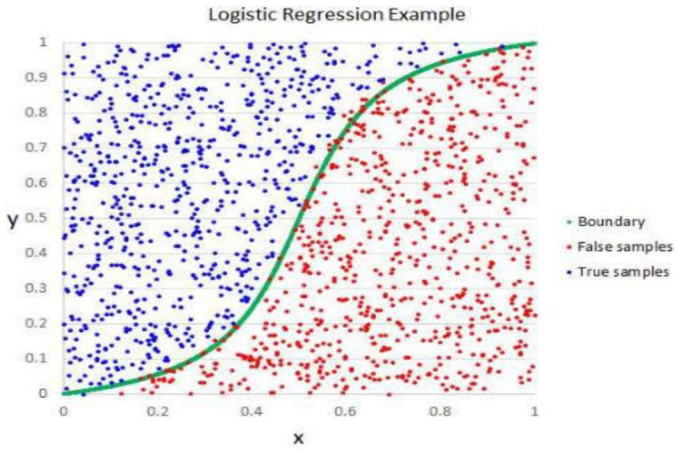
Model Selection

5.3 Feature Engineering

- Logistic regression is a statistical model that in its basic form uses a logistic function to model a binary dependent variable, although many more complex extensions exist. In regression analysis, logistic regression (or logit regression) is estimating the parameters of a logistic model (a form of binary regression).

- We use the Logistic Regression algorithm and fit the data into this model and check the results.

  We use the class LogisticRegression from the library scikit learn.



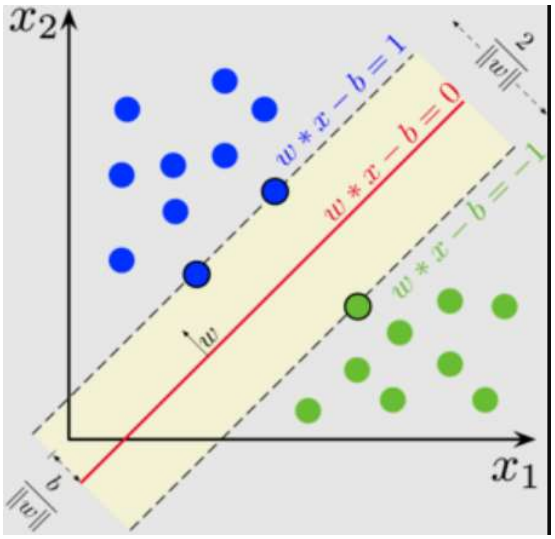
- The results we achieve are:

```
In [22]:  1  # Finding the Accuracy Score
          2  accuracy = mean(prediction_df.correctly_classified)

Out[22]: 0.5699977743156021

In [ ]:   1
```

- We achieve accuracy of 56.99% which is not really good. Our next objective is to use other algorithms and compare the result.

5.4 SVM(Linear)

- A Support Vector Machine (SVM) is a discriminative classifier formally defined by a separating hyperplane. In other words, given labelled training data (supervised learning), the algorithm outputs an optimal hyperplane which categorizes new examples.
- We use the class SVC from the library scikit learn with kernel = “linear”.



- The results we achieve are:

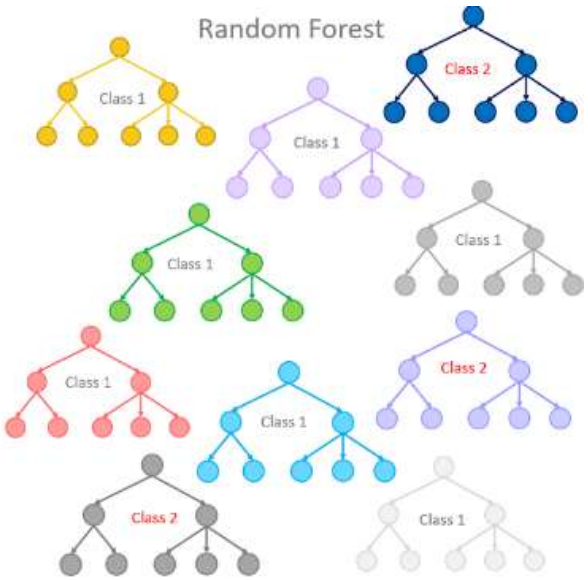
```
In [53]:  1  # SVM (Linear)
          2  println("Accuracy : ", accuracy_svm)

          Accuracy : 0.5423992877809927

In [ ]:   1
```

- We can observe that SVM (linear) is giving better results than Logistic Regression, we can change the parameters of SVM and try to fir our data into it once again.

5.5 Random Forest

- Random forests or random decision forests are an ensemble learning method for classification, regression and other tasks that operates by constructing a multitude of decision trees at training time and outputting the class that is the mode of the classes (classification) or mean prediction (regression) of the individual trees. Random decision forests correct for decision trees' habit of overfitting to their training set.

- Ensemble Learning : Random forests or random decision forests are an ensemble learning method for classification, regression and other tasks that operates by constructing a multitude of decision trees at training time and outputting the class that is the mode of the classes (classification) or mean prediction (regression) of the individual trees. Random decision forests correct for decision trees' habit of overfitting to their training set.



- We use the class RandomForestClassifier from sklearn.ensemble. We are running the algorithm at

- n_estimators = 100 , this means that it will take average of 100 predictions from the model.
- The results we achieve by using Random Forest Classifier are :

In [88]:
```
# Predict
@sk_import metrics: accuracy_score
@sk_import metrics: f1_score

y_pred = model.predict(X_test)
println("Accuracy : ", accuracy_score(y_test, y_pred))
println("F1 Score : ", f1_score(y_test, y_pred))
```

```
Accuracy : 0.6274111675126903
F1 Score : 0.6801278326554329
```

- Random Forest being one the best models in classification algorithms gives immensely better results than all the previous models that is 62.17%.
- Apart from accuracy Random Forest is giving better results in precision, recall and f1 score too.

6 Optimizing Model

Optimization is the most essential ingredient in the recipe of machine learning algorithms. It starts with defining some kind of loss function/cost function and ends with minimizing them using one or the other optimization routine. The choice of optimization algorithm can make a difference between getting a good accuracy in hours or days. The applications of optimization are limitless and is widely researched topic in industry as well as academia.

6.1 K-Fold Validation

K-Fold CV is where a given data set is split into a K number of sections/folds where each fold is used as a testing set at some point. Let's take the scenario of 10-Fold cross validation(K=10). Here, the data set is split into 10 folds. In the first iteration, the first fold is used to test the model and the rest are used to train the model. In the second iteration, 2nd fold is used as the testing set while the rest serve as the training set. This process is repeated until each fold of the 10 folds have been used as the testing set.

```
In [43]:  1 using ScikitLearn.CrossValidation: cross_val_score
          2 accuracy = cross_val_score(model, Xtest, ytest, cv=10)

Out[43]: 10-element Array{Float64,1}:
          0.5933333333333334
          0.6266666666666667
          0.5688888888888889
          0.6
          0.58
          0.5777777777777777
          0.579064587973274
          0.6138392857142857
          0.6026785714285714
          0.6183035714285714

In [ ]:  1
```

7 Conclusion

Our model has given us an accuracy of 62.74%. With this, we have an algorithm that can help predict whether or not a user will complete the E-signing step of the loan application. One way to leverage this model is to target those predicted to not reach the e-sign phase with customized onboarding. This means that when the lead arrives from the p2p, they may receive a different onboarding experience based on how likely they are to finish the general onboarding process. This can help the company minimize how many people drop off from the funnel. This funnel of screens is as effective as we, as a company, build it. Therefore, user drop-off in this funnel falls entirely on our shoulders. So, with new onboarding screens built intentionally to lead users to finalize the loans application, we can attempt to get more than 40% of those predicted to not finish the process to complete the e-sign step. If we can do this, then we can drastically increase profit. Many lending companies provide hundreds of loan everyday, gaining money for each one. As a result, if we can increase the number of loan takers, we can increase the profits. All with a simple model!

The algorithms used for this modes was Logistic Regression, SVM, Decision Tree Classifier, Random Forest.

The method of preparation and selection of features is one of the biggest aspect of the success of this model. The Random Forest which yielded the best results was optimized even further using k-fold validation and grid search method and best parameters was selected.

8 Future Scope of Project

Our model gave a final prediction accuracy of 62.74% on the dataset of 300,000 entries. Which might be satisfactory for some p2p companies but we could further try to increase the accuracy of our model using a few more optimization techniques. We could further try to better the execution time of the model. At the moment it takes around 2 to 2.5 hours to execute which could be bettered by somehow manipulating our given data set.

Not only it can be used to study whether a customer will end up e-signing for loan or not we could even use this model in future to predict which person applying for the loan is likely to pay back loan on time based on the financial history of the person. It could even be used to decide what is the maximum amount which could be given to a person which he or she is likely to pay back on time in the future.

9 Bibliography

Reference Used :

- Julia
- https://julialang.org/community/
- Julia Libraries :
- https://dataframes.juliadata.org/stable/
- https://csv.juliadata.org/v0.1.1/
- https://docs.juliaplots.org/latest/tutorial/
- Sci-Kit Learn :
- https://scikitlearnjl.readthedocs.io/en/latest/