Machine Learning (UG): CSE343/ECE343
Project Progress Report
Automated Financial Distress Predictor

Ashwin Sheoran: 2020288
Harsh Goyal: 2020562
Shivam Jindal: 2020125
Sourav Goyal: 2020341

1. Motivation

Banks play a crucial role in market economies. They decide who can get finance and on what terms and can make or break investment decisions. For markets and society to function, individuals and companies need access to credit.
Credit scoring algorithms, which guess the probability of default, are the methods banks use to determine whether or not a loan should be granted. This model helps to improve state of the art in credit scoring by predicting that someone will experience financial distress in the next two years.

2. Describing the Problem Statement

Our models take various input parameters like age, monthly income and debt ratio and predict whether the person will experience 90 days past due delinquency or worse, i.e. if the person fails to pay back his/her loan even after 90 days of the due date. Our models predict whether this delinquency will occur or not.

3. Literature Survey

We researched using 2 research papers.

1. Automated Credit Scoring System for Financial Services in Developing Countries [1] by Rebeka Sultana, Samira Muntaha, Farhana Sarker, D. M. Anisuzzaman, Khondaker A. Mamun suggests using Linear regression over decision trees and KNN, The only problem with Linear Regression was that the quality of the model decreased when a small dataset was used.
For KNN, predictive accuracy was extremely affected by the measure of distance and the cardinality of the neighborhood.
DT could not predict correctly when a smaller dataset was used.

2.. Financial Distress Prediction based on Multi-Layer Perceptron with Parameter Optimization [2] by Magdi El Bannany, Ahmed M. Khedr, Meenu Sreedharan and Sakeena Kanakkayil, Although this paper suggests we use Neural Networks, apart from it on comparing the bias and variance of different models we can see that random forests are best suited for this.

4. Dataset details

We took our dataset from Kaggle[3]; we have the data of 251503 people, which was divided into training and test dataset.

Feature	Data Type
RevolvingUtilizationOfUnsecuredLines	float
age	int
NumberOfTime30-59DaysPastDueNotWorse	int
DebtRatio	float
MonthlyIncome	int
NumberOfOpenCreditLinesAndLoans	int
NumberOfTimes90DaysLate	int
NumberRealEstateLoansOrLines	int
NumberOfTime60-89DaysPastDueNotWorse	int
NumberOfDependents	int

Data Preprocessing

Handling missing Values using SimpleImputer

We counted the NaN values in the dataset and replaced the NaN Values by the mean along each column using SimpleImputer. SimpleImputer takes the mean of the column and replaces the missing values with the mean.

Data Standardization

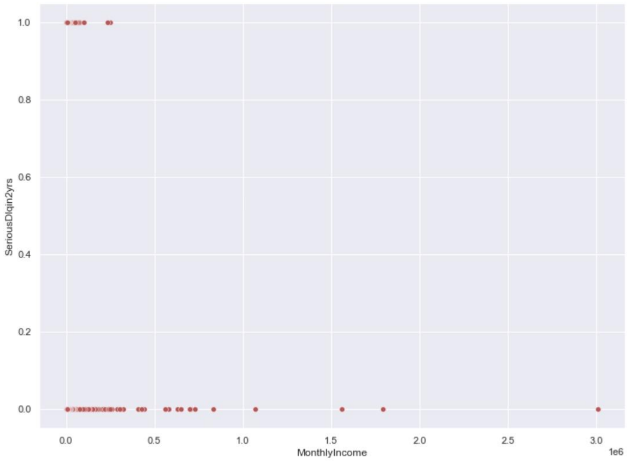
Standardization is a scaling technique where values are centered around the mean with a unity standard deviation. This means that the mean of the attribute becomes zero and the resultant distribution has a unit standard deviation. The formula used for standardization is

$$z_i = (x_i - \bar{x}) / \sigma$$

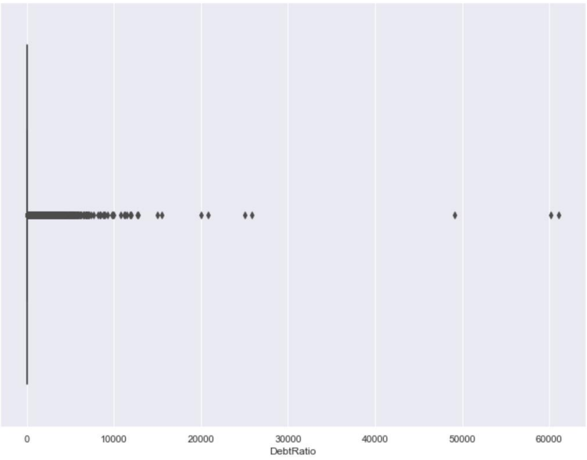
Here $\bar{x}$ is the mean of the values σ is the standard deviation of the values.

EDA

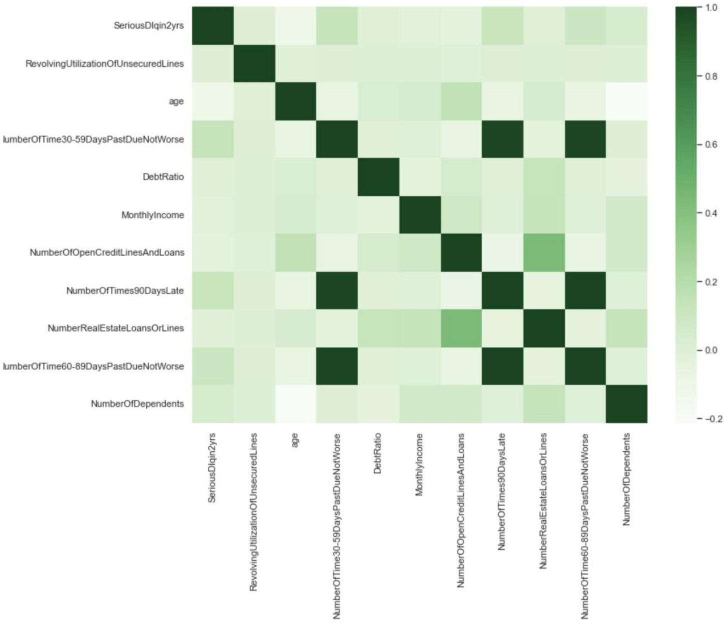
Here, we performed EDA Analysis.



This is a scatter plot graph between Serious Delinquency Rate in 2 years and Monthly Income before we did Standardization. Here we can easily see that People with More Monthly income will face less Delinquency in paying back loans.



Here we can see that before Standardization, the Debt ratios greater than 1 are outliers in the data.



This is the heat map between the features

5. Methodology

Our objective is to classify whether the person fails to pay back his/her loan even after 90 days of the due date. We tried different ML-based classification algorithms for this binary classification problem, like Logistic Regression, Naive Bayes, Decision Trees, and Random Forests. We also tried different hyperparameter values in some of these classification algorithms to visualize the variations in the accuracy of the models with respect to these hyperparameters.

5.1 Decision Trees

We used non-scaled data for decision trees because decision trees are not sensitive to the variance of the data. Decision trees work as an if-else mechanism; hence, they are insensitive to the data points' values. We also visualized Decision Trees for different max depths and used both "entropy" and "gini" as our criterion.

5.2 Random Forests

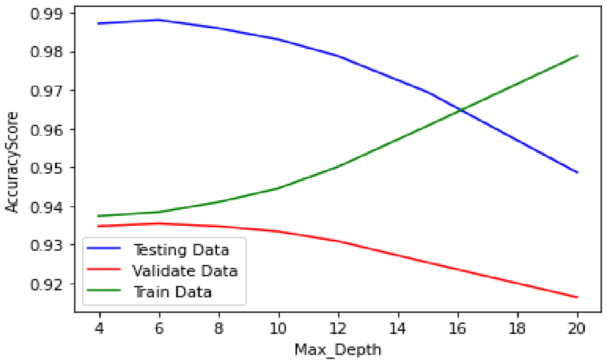
We used non-scaled data for random forests because random forests are combinations of many weak decision trees (weak classifiers). Since decision trees are not sensitive to the data's variance, random forests are also not sensitive to the variance of the data. We also visualized Random Forests for different max depths with "gini" as the default criterion and 100 as the default n_estimators. We use the ensemble method to combine these weak decision trees using the maximum voting technique to make a random forest.

6. Results and Analysis

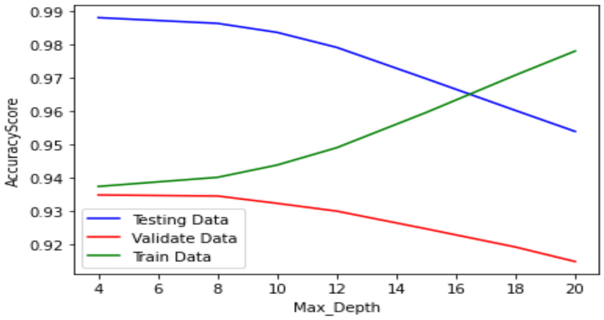
Model Type	Training Accuracy	Validation Accuracy	Testing Accuracy
LR	93.23	93.75	98.39
NB	93.19	93.20	98.12
DT (criterion = gini)	93.83	93.50	98.81
DT (criterion = entropy)	93.63	93.47	98.88
RF	94.12	93.88	99.47

Random Forest algorithm outperforms all the models with a testing accuracy of 99.47% across all the models. Decision Trees with "entropy" as the criterion performed better than "gini" as the criterion. Naive Bayes has the lowest accuracy among all the models. This may be because Naive Bayes assumes all the features to be independent of each other.

Decision Tree Classification using "Gini" as the criterion:

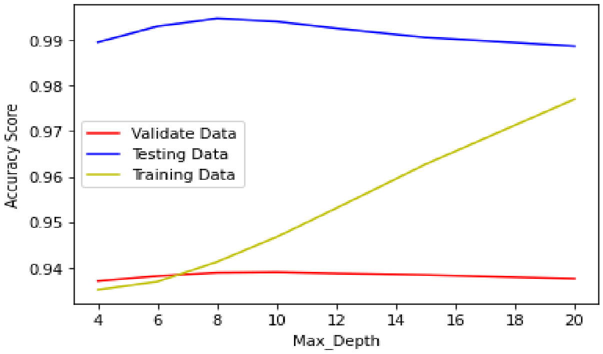


Decision Tree Classification using "Entropy" as the criterion:



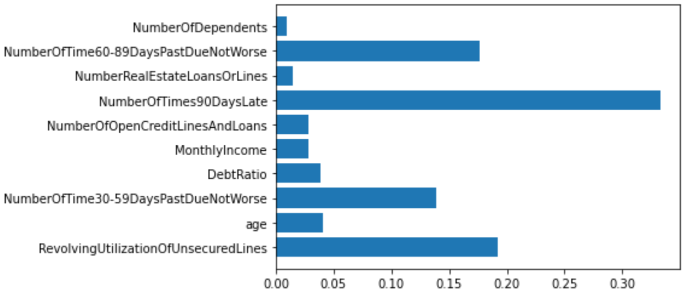
From the above graphs, we can conclude that as the model complexity increases, that is, the max depth for the decision tree is increasing, the training accuracy is also increasing, and hence bias is decreasing. But as the model complexity increases, testing accuracy initially increases. After a certain point, testing accuracy starts decreasing, and further increasing the model complexity, it overfits the model and makes it less generalized.

Random Forest Classifier:



From the above graph of Random Forest (criterion = "gini" and n_estimators = 100)) with different values of max depth, we can conclude that bias decreases as model complexity increases, but the testing accuracy first increases with an increase in model complexity but later on decrease with further increase in model complexity which shows that model overfits as max depth increases.

Random Forest Feature Importance:



From the above graph based on Random Forest Classification, the "NumberofTimes90DaysLate" feature is the most important feature while "NumberofDependents" is the least important feature.

7. Conclusion

From the above results and analysis, we can conclude several observations:-

7.1 Learning from the Project

- Learning about SimpleImputer class for replacing missing values by strategies like mean, mode, etc of the column values.
- By combining many weak classifiers, we can make a strong classifier.
- If there is a linear relationship between the input and output, then standardization can be used to improve the model's performance.
- As we change the hyperparameters in our models like in Random Forest, model complexity increases, and after an optimal point of hyperparameter, it decreases.
- There is always a tradeoff between bias and variance. As model complexity increases, bias decreases but variance increases.
- Evaluating models to be good or bad based on the model performance/metrics.
- Understanding how to debug an ML model.

7.2 Work Left

Most of the part has been done with our project. Now we are left with the below task, which we will perform on our models:-

- Implementing feature extraction
- Hyperparameter tuning
- Dimensionality reduction with PCA, TSNE
- Checking for overfitting and underfitting
- Some ML algorithms are left:- SVM, MLP, Boosting and other algorithms.

7.3 Contribution of each Member

Tasks	Team Members
Data Collection	Ashwin and Harsh
Pre-Processing and Data Visualization	Harsh and Shivam
Feature Extraction	Shivam and Sourav
Analysis of Features (EDA etc.)	Ashwin and Sourav
Logistic Regression	Ashwin and Shivam
Naive Bayes	Ashwin and Sourav
Decision Tree	Harsh and Shivam
Random Forest	Harsh and Sourav
Analysis and Performance of Models	Ashwin, Harsh, Shivam, and Sourav
Presentation and Report creation	Ashwin, Harsh, Shivam, and Sourav

References

[1] https://www.researchgate.net/publication/328559774_Automated_Credit_Scoring_System_for_Financial_Services_in_Developing_Countries

[2] http://www.iaeng.org/IJCS/issues_v48/issue_3/IJCS_48_3_41.pdf

[3] https://www.kaggle.com/competitions/GiveMeSomeCredit/data