CSCI-GA.3033-011

Big Data Science

Give Me Some Credit

Predicting Financial Stress within 2 years using RapidMiner, Knime and Spark (MlLib Classification)

Bhatla, Anisha
Dept. of Compubleter Science
NYU
NY, USA
ab6544@nyu.edu

Kalra, Abhineet
Dept. of Computer Science
NYU
NY, USA
ak5345@nyu.edu

Malviya, Jatin
Dept. of Computer Science,
NYU
NY, USA
jm6474@nyu.edu

Mishra, Subhankari
Dept. of Computer Science
NYU
NY, USA
sm6202@nyu.edu

Raghupathy, Vinayak
Dept. of Computer Science
NYU
NY, USA
vr840@nyu.edu

Abstract—

The credit card industry has been recently growing rapidly, and thus huge numbers of consumers' credit data are collected by the credit department of the bank. The credit scoring manager often evaluates the consumer's credit with intuitive experience. Our project aims to use Data Science to provide a tool other than Credit Score to the general public to get an idea about their future financial condition. It will be useful to Banks, Money Lenders, Credit Card providers and even to common man in making more concrete financial decisions. However, with the support of the credit classification model, the manager can accurately evaluate the applicant's credit score.

Keywords—Big Data, Data Science, Spark, RapindMiner, Knime, Scala, Credit Scoring, Classification, Machine Learning, Random Forest Classifier, Gradient Boosted Tree Classifier, Decision Tree Classifier, Naïve Bayes Classifier, Support Vector Machine Classifier, Multi-layer Perceptron, Analytics, MlLib, Cross Validation

I. INTRODUCTION

Should I take loan right now? Is it the right time for me to invest somewhere? Is a new Credit Card better idea? Do I need to cut my expenses? Do I need to start saving money? Financial Decisions are always tough. There is a need for a tool which can help us in making these decisions.

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.

We aim to demonstrate how the large records of a person's transactions can be used to predict if the person may or may not face financial distress in the coming years based on patterns in the labelled transactions historical data of other people.

Credit scoring algorithms, which make a guess at the probability of default, are the method banks use to determine whether or not a loan should be granted. In this project we experimented with Big Data

Science concepts and various classification algorithms namely, Ensemble methods, Decision Tree, Naive Bayes Classification, Support Vector Machines and Multilayer Perceptron, to predict if a person will experience financial distress in the next two years.

II. Process Overview

The Cross Industry Standard Process for Data Mining (CRISP-DM) was followed for this project as CRISP-DM is one of the top models for data mining process used to solve problems by experts. It provides a streamlined approach for project development considering the iterative nature of data mining and machine learning projects.

CRISP-DM is based on the process flow shown in Figure 1.

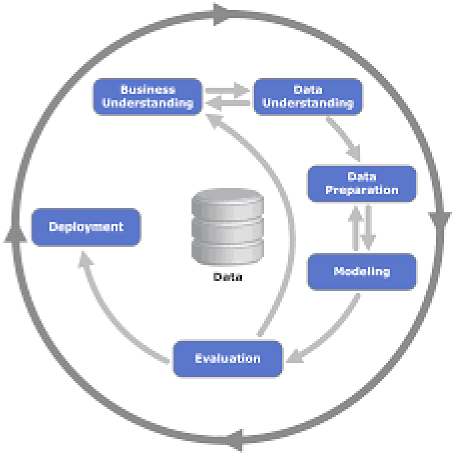


Figure 1. CRISP-DM Model

The model proposes the following steps:

1. Business Understanding – to understand the rules and business objectives of the company.
2. Understanding Data – to collect and describe data.
3. Data Preparation – to prepare data for import into the software.
4. Modelling – to select the modelling technique to be used.
5. Evaluation – to evaluate the process to see if the technique solves the problem of modelling and creation of rules.
6. Deployment – to deploy the system and train its users.

The major steps involved in this process are described in the later sections.

III. Technical Specifications

Programming Languages:

Scala

Framework:

Spark 1.6.0

Libraries:

Spark MLlib 1.6

Tools:

RapidMiner 7.0

Knime 3.1.1

Platforms Used:

Windows

Dumbo Cluster at NYU

Cloudera Quickstart VM 5.5

Dataset:

Kaggle Give me some Credit Training Data (7.21MB) (https://www.kaggle.com/c/GiveMeSomeCredit/data)

IV. Business Understanding

At first we tried to get a better understanding of the project objectives and requirements from a business perspective, then converted this knowledge into a data mining problem definition and planned a design to achieve the objectives.

Lenders, such as banks and credit card companies, use credit scores to evaluate the potential risk posed by lending money to consumers and to mitigate losses due to bad debt. A Credit Score is a numerical expression based on level analysis of a person's credit files, to represent the creditworthiness of the person. Credit scoring overlaps with data mining, which uses many similar techniques. These techniques combine thousands of factors but are similar or identical.

In USA, FICO score is the most widely known Credit Score developed by Fair Isaac Corporation. Due to the proprietary nature of the FICO score, the exact formula it uses to compute this number is not revealed. However, it is known that the calculation is broken into five major categories with varying levels of importance:

1. Payment History
   The percentage of on-time payments on various account types. Previous problems (size & time) in payment history like bankruptcy, collections, delinquency, etc.

2. Amount Owed
   The current amount of debt that is owed to lenders. It also looks at the number of different debt sources and the debt age. This gives an idea about the present financial situation.

3. Length of Credit History
   Since when the credit score is being maintained for the account. The longer you have a good credit history, the better.

4. New Credit
   How many credit lines the person has. A person with 10 credit cards is riskier than a person having 1 credit card. How many of these credit lines are new. Also, average age of credit accounts and total number of hard credit inquiries contributes to the credit score.

5. Type of Credit Used
   Having various types of credit, both revolving and installment, on the profile can positively contribute to creditworthiness

Studying and researching about the credit score calculation gave us a fair idea about what all factors a money lending company sees while giving out the credit, thus helping us to understand what all features we need to look for while deciding the financial situation of a person.

V. Data Understanding

The initial data should be collected and a description of this data must be produced, as well as a verification of its quality. This is where the default history of the bank is synthesized, with the required attributes such as SeriousDlqin2yrs, age and DebtRatio so on.

The downloaded data was in csv format of 7.21 MB. It consists of 150,000 rows.
Data size: 7.21 Mb
File format: csv
No. of data rows: 150,000
Data Description

Variable Name	Description	Type
SeriousDlqin2yrs	Person experienced 90 days past due delinquency or worse	Y/N
RevolvingUtilizationOfUnsecuredLines	Total balance on credit cards and personal lines of credit except real estate and no installment debt like car loans divided by the sum of credit limits	percentage
Age	Age of Borrower in Years	integer
NumberOfTime30-59DaysPastDueNotWorse	Number of times borrower has been 30-59 days past due but no worse in the last 2 years	integer

DebtRatio	Monthly debt payments, alimony,living costs divided by monthy gross income	percentage
MonthlyIncome	Monthly Income	real
NumberOfOpenCreditLinesAndLoans	Number of Open loans (installment like car loan or mortgage) and Lines of credit (e.g. credit cards)	integer
NumberOfTimes90DaysLate	Number of times borrower has been 90 days or more past due.	integer
NumberRealEstateLoansOrLines	Number of mortgage and real estate loans including home equity lines of credit	Integer
NumberOfTime60-89DaysPastDueNotWorse	Number of times borrower has been 60-89 days past due but no worse in the last 2 years.	Integer
NumberOfDependents	Number of dependents in family excluding themselves (spouse, children etc.)	Integer

Table 1. Feature Description

VI. DATA PREPARATION

We explored the state-of-the-art on dimensionality reduction techniques currently available and accepted in the data analytics landscape:

1. **Replace Missing Values:** Data columns with missing values are replaced by specified replacements such as median, minimum, maximum, most frequent value of the attribute. The replacement option giving the highest accuracy is chosen for further data reduction.

Preprocessing Method	Accuracy	Notes
Replace Missing value	92.96%	Knime used, Median value Used
Replace Missing Values	92.96%	Knime Used, Most Frequent Value Used

Table 2. Replace Missing Values Accuracy

2. **Low Variance Filter**: Data columns with little changes in the data carry little information. Thus all data columns with variance lower than a given threshold are removed.

Preprocessing Method	Accuracy	Notes
Low Variance Filer	92.21%	Rapid Miner Used, Min. Deviation=1
Low Variance Filer	93.33%	Rapid Miner Used, Min. Deviation=2

Table 3. Low Variance Filter Accuracy

3. **High Correlation Filter**: Data columns with very similar trends are also likely to carry very similar information. In this case, only one of them will suffice to feed our machine learning model. Pairs of columns with correlation coefficient higher than a threshold are reduced to only one.

Preprocessing Method	Accuracy	Notes
High Correlation Filter	92.92%	Knime Used, Threshold: 1
High Correlation Filter	92.96%	Knime Used, Threshold: 0.5

Table 4. High Correlation Filter Accuracy

4. **Missing Values Ratio**: Data columns with too many missing values are unlikely to carry much useful information. Thus data columns with number of missing values greater than a given threshold can be removed. The higher the threshold, the more aggressive the reduction.

Preprocessing Method	Accuracy	Notes
Missing Value Ratio	92.96%	Knime Used, Threshold: 5

Missing Value Ratio	92.96%	Knime Used, Threshold: 10

Table 5. Missing Values Ratio Accuracy

5. **Principal Component Analysis (PCA)**: Principal Component Analysis (PCA) is a statistical procedure that orthogonally transforms the original *n* coordinates of a data set into a new set of *n* coordinates called principal components. As a result of the transformation, the first principal component has the largest possible variance; each succeeding component has the highest possible variance under the constraint that it is orthogonal to the preceding components. Keeping only the first *m* < *n* components reduces the data dimensionality while retaining most of the data information, i.e. the variation in the data.

Preprocessing Method	**Accuracy**	**Notes**
Replace Missing Value with Minimum PCA	93.32%	Naive Bayes Cross Validation
Replace Missing Value with Fixed Value PCA	93.32%	Naive Bayes Cross Validation Fixed Value=30

Table 6. PCA Accuracy

6. **Backward Feature Elimination**: In this technique, at a given iteration, the selected classification algorithm is trained on *n* input features. Then we remove one input feature at a time and train the same model on *n-1* input features *n* times. The input feature whose removal has produced the smallest increase in the error rate is removed, leaving us with *n-1* input features. The classification is then repeated using *n-2* features, and so on.

Preprocessing Method	**Accuracy**	**Notes**
Replace Missing value with Backward Elimination	93.35%	Linear Regression, Maximum Number of Eliminations: 5
Replace Missing value with Backward Elimination	93.04%	Linear Regression, Maximum Number of Eliminations: 4

Table 7. Backward Feature Elimination Accuracy

7. **Forward Feature Construction**: This is the inverse process to the Backward Feature Elimination. We start with 1 feature only, progressively adding 1 feature at a time, i.e. the feature that produces the highest increase in performance.

Preprocessing Method	**Accuracy**	**Notes**
Replace Missing value with Forward Selection	93.33%	Linear Regression, Maximum Number of Attributes: 4
Replace Missing value with Forward Selection	93.12%	Linear Regression, Maximum Number of Attributes: 6

Table 8. Forward Feature Construction Accuracy

8. **Singular Value Decomposition:** This is a data mining technique wherein unnecessary data that are linearly dependent on another attribute/field are removed from the dataset. This results in a low dimensional representation of a high dimensional matrix, thus making it simpler to remove less important values/parts, which in turn depends upon the number of dimensions required.

Preprocessing Method	**Accuracy**	**Notes**
Replace Missing Value with Minimum SVD	93.33%	Naive Bayes Cross Validation
Replace Missing Value with Minimum SVD	93.32%	Naive Bayes Cross Validation Backward Elimination

Table 9. SVD Accuracy

The maximum accuracy was achieved with the following combinations:

Preprocessing Method	**Accuracy**	**Notes**
Replace Missing Values + Low covariance Filter + High correlation Filter + Backward Elimination	93.36%	Min. Deviation = 2 Correlation = 1
Replace Missing Values + Low covariance Filter + High correlation Filter + Forward Selection	93.36%	Min. Deviation = 2 Correlation = 1

Table 10. Maximum Accuracy Achieved

VII. Modelling & Evaluation

We are solving a data classification problem with two classes- A person may or may not a face financial distress in the coming years. Classification is the problem of identifying to which of a set of categories (sub-populations) a new observation belongs, on the basis of a training set of data containing observations (or instances) whose category membership is known. We implemented and experimented with several Classification Models. Also, we evaluated these models to be certain that the model properly achieves the business objectives by using the method of Cross-Validation.

1. Naïve Bayes Classifier Model

Naive Bayes is a simple multiclass classification algorithm which assumes that every pair of features is independent. It can be trained very efficiently. Within a single pass on the training data, it computes the conditional probability distribution of each feature given the label, and then it applies Bayes' theorem to compute the conditional probability distribution of the label given an observation and use it for prediction for the new data.

$$p(C_k|\mathbf{x}) = \frac{p(C_k)\ p(\mathbf{x}|C_k)}{p(\mathbf{x})}$$

$$\mathbf{x} = (x_1, \dots, x_n)$$

Here x represents the n features (independent) and C_k represents the K outcomes or classes.

We implemented the Multinomial naive Bayes algorithm for our data, and it is one of the two classic naive Bayes variants. In Multinomial variant we can think each observation as a document and each feature in it is a term whose value is it frequency. We trained our Multinomial Naïve Bayes Classifier model with additive smoothing parameter lambda=1.0 and feature values must be non-negative. Accuracy obtained is:

Training Data %	**Test Data %**	**Accuracy %**
60%	40%	91.21 %
70%	30%	92.12%
80%	20%	91.04%
90%	10%	83.71%

Table 11. Multinomial Naïve Bayes Classifier Accuracy

2. Decision Trees Classifier Model

Decision tree builds classification or regression models in the form of a tree structure. It breaks down a dataset into smaller and smaller subsets while at the same time an associated decision tree is incrementally developed.

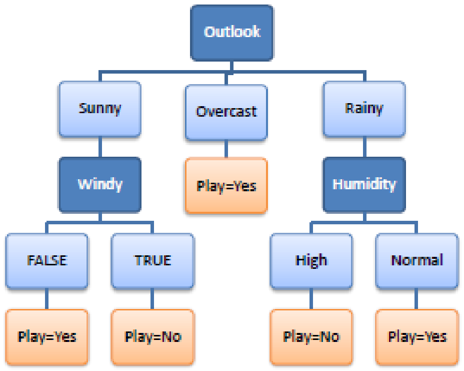


Figure 2. Decision Tree Example

The core algorithm for building decision trees called ID3 by J. R. Quinlan which employs a top-down, greedy search through the space of possible branches with no backtracking. ID3 uses Entropy and Information Gain to construct a decision tree. We trained our decision tree with maximum depth of 5 and obtained the following accuracy:

Training Data %	Test Data %	Accuracy %
60%	40%	93.72%
70%	30%	93.52%
80%	20%	93.69%
90%	10%	93.57%

Table 12. Decision Tree Classifier Model Accuracy

ENSEMBLE METHODS

Ensemble methods use multiple learning algorithms to obtain better predictive performance than that obtained from any of the constituent learning algorithms. Figure 2 shows how this works - Given a data set, generate multiple models and combine the results.

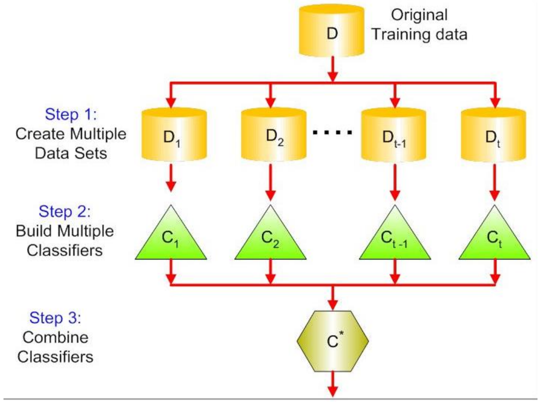


Figure 3. Ensemble Method

3. Random Forest Classifier Model

Random Forests Model is an ensemble learning method for classification, regression and other tasks, that operate 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.

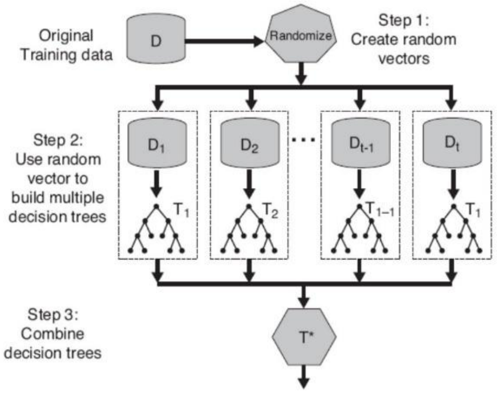


Figure 4. Random Forests

Combining the predictions from each tree reduces the variance of the predictions, improving the performance on test data. Each tree's prediction is counted as a vote for one class. The label is predicted to be the class which receives the most votes.

We trained our Random Forest model with the maximum number of trees set to 10. The Accuracy we:

Training Data %	Test Data %	Accuracy %
60%	40%	93.51%
70%	30%	93.50%
80%	20%	93.64%
90%	10%	93.51%

Table 13. Random Forest Model

4. Gradient-Boosted Classifier Model

Gradient-Boosted Trees (GBTs) are ensembles of decision trees. GBTs iteratively train decision trees in order to minimize a loss function. Like decision trees, GBTs handle categorical features, extend to the multiclass classification setting, do not require feature scaling, and are able to capture non-linearities and feature interactions.

Gradient boosting iteratively trains a sequence of decision trees. On each iteration, the algorithm uses the current ensemble to predict the label of each training instance and then compares the prediction with the true label. The dataset is re-labeled to put more emphasis on training instances with poor predictions. Thus, in the next iteration, the decision tree will help correct for previous mistakes.

We trained our Gradient Boosting Classifier model with the number of trees set to 10.

Training Data %	Test Data %	Accuracy %
60%	40%	93.64%
70%	30%	93.58%
80%	20%	93.43%
90%	10%	93.73%

Table 14. Gradient Boosted Classifier Model

5. Support Vector Machine Classifier Model

Support Vector Machine (SVM) is a supervised machine-learning method that can be used for binary classification. It also extends to patterns that are not linearly separable by transformations of original data to map into new space, referred to as the kernel-trick. The goal is to select one hyper plane that maximizes the margin of separation. So we choose the hyper plane so that the distance from it to the nearest data point on each side is maximized. Below is the objective function for SVM:

$$\min P(\boldsymbol{w}, b) = \underbrace{\frac{1}{2}\|\boldsymbol{w}\|^2}_{\text{maximize margin}} + \underbrace{C\sum_i H_1[y_i f(\boldsymbol{x}_i)]}_{\text{minimize training error}}$$

Where H_1 is the hinge loss at the data points.

Hinge Loss is given by $H_1 = \max(0, 1-yf(x))$.

This hinge loss function is not differentiable so the huberized hinge loss is used which is smoothed version of hinge loss.

For our project, we applied SVM with Stochastic Gradient Descent using MLlib SVMWithSGD under classification models. Gradient Descent algorithm aims at finding a weight vector i.e. the parameter "θ" that minimizes the objective function J(θ). The algorithm starts with an initial weight vector usually initialized to zero and then repeatedly changes "θ" to make J(θ) smaller until the it converges to a value of "θ" that minimizes J(θ). The weight vector is updated using the below formula:

$$\theta_j := \theta_j - \alpha \frac{\partial}{\partial \theta_j} J(\theta)$$

Here, α is called the learning rate. This is a very natural algorithm that repeatedly takes a step in the direction of steepest decrease of J.

This algorithm updates the weight vector after processing the entire data in the training set which is not feasible when the dataset is large and cannot fit in memory. For large datasets, Stochastic Gradient Descent is used which looks at one data sample at a time and updates the weight vector. To get unbiased results the data sample can be picked randomly. The model was evaluated by calculating the accuracy for the experiments run for 100 iterations as described in the below table.

Training Data %	Test Data %	Accuracy %
60%	40%	92.98%
70%	30%	92.82%
80%	20%	92.91%
90%	10%	92.99%

Table 15.SVM Model

6. Multilayer Perceptron Classifier Model

Perceptron is linear binary classifier, assumes that data is linearly separable and will converge to any available hyperplane. Similar to gradient descent in perceptron also we use a weight vector which is initialized to some random value or zero and is updated every time the function finds an error i.e. the predicted label does not match with the actual label while iterating over the training set using the below equation:

$$w = w + \eta y x$$

Where, η is the learning rate used by the algorithm and can be initialized to any value between 0 and 1.

Multilayer perceptron (MLP) used in this project is a feedforward artificial neural network model that maps sets of input data onto a set of appropriate outputs. It is a modification of standard linear perceptron and can distinguish data that are not linearly separable. MLP uses a supervised learning technique called backpropagation for training the network. An MLP consists of multiple layers of nodes in a directed graph, with each layer fully connected to the next one. Below is the high level overview of MLP network:

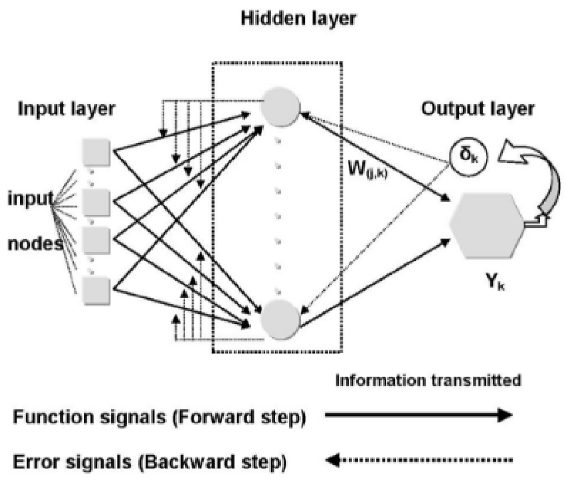


Figure 5. Multilayer Perceptron Classifier

Below are the experiments and results for MLP.

Training Data %	Test Data %	Accuracy %
60%	40%	93.27%
70%	30%	93.24%
80%	20%	93.22%
90%	10%	93.09%

Table 16. Multilayer Perceptron Model

VIII. Deployment

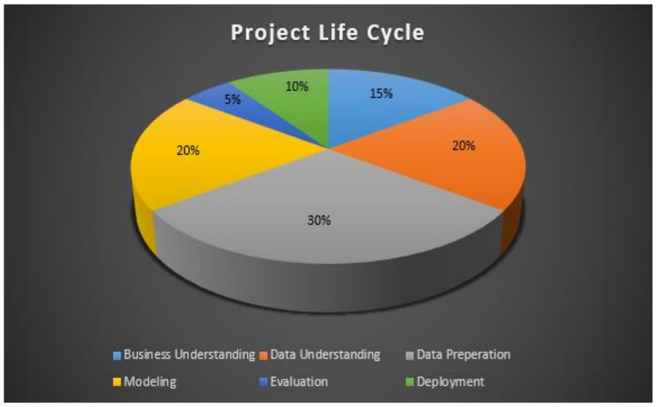
As we know the strength of different models in our data, we select the model which can most benefit the users and different money lenders and organization in the deployment phase. Also, it is required that we create artifacts in each preceding phase of the process so as to conduct training sessions with the users of the system.

The models developed in this work will not last long with bank data being produced continuously. Also, customer defaults occur frequently so it is necessary that the model is flexible enough to incorporate these changes. Information that is true today may not be true tomorrow, since the data is very volatile and different types of fault are always expected.

We can implement our project in such a way that it can handle more data for our training model which adds the possibility of flexibility in our project with new features and testing those features.

IX. Experiment & Results

Finally, all phases of CRISP-DM Model were completed. Our efforts can be summed as below:

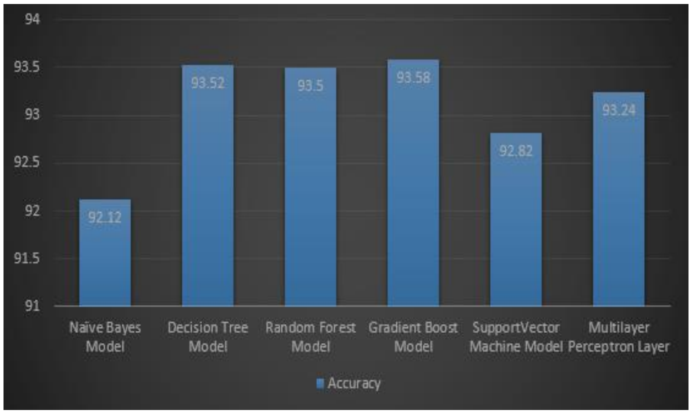


Graph 1. Crisp-DM Model Effort

Hence, we implemented 6 models on our data and their accuracies were (70% Training Data & 30% Test Data):

Model	Accuracy
Naïve Bayes Model	92.12%
Decision Tree Model	93.52%
Random Forest Model	93.50%
Gradient Boosted Model	93.58%
Support Vector Machine Model	92.82%
Multilayer Perceptron Layer	93.24%

Table 17. Final Accuracies



Graph 2. Model Comparison

Thus, Ensemble Methods gave us the best accuracy particularly Gradient Boost Model. Naïve Bayes Model gave the worst performance on our data.

X. Conclusion

We implemented the project on predicting whether a person will face a financial stress within 2 years using RapidMiner, Knime and Spark (MlLib Classification).

For data preparation we tried different preprocessing methods on our data and chose the best performing method. Next, we implemented different classification models namely Naïve Bayes, Decision Tree, Random Forest, Gradient Boosted, SVM and Multilayer Perceptron Classification Model. Lastly, we compared the accuracies of the models and analyzed the results.

Thus, this project gave us a better understanding of Data Science concepts particularly the Crisp-DM model, Data Preprocessing methods, Data Classification and Validation.

XI. Future Scope

This project shows how the power of Data Science can be used in a real life scenario. We have shown that this tool can help in making better financial decisions. This approach can also be used by the Money Lenders, Banks, Credit Card companies, general public with some modifications.

Also, we tried to implement and experiment with several Classification Algorithms in this project. But still, there are several other AI Algorithms like Fuzzy Algorithms & Genetic Algorithms, which are not supported by Spark's MlLib Library which might result in better accuracy.

References

[1]http://spark.apache.org/

[2]http://spark.apache.org/docs/latest/mllib-guide.html

[3]Credit scoring with a data mining approach based on support vector machines

[4]Credit Scoring Models Using Soft Computing Methods

[5]Using Genetic Algorithms for Credit Scoring System Maintenance Functions

[6]Building credit scoring models using genetic programming

[7]Neural network credit scoring models

[8]Inferring descriptive and approximate fuzzy rules for credit scoring using evolutionary algorithms

[9] https://en.wikipedia.org/wiki/Credit_score

[10] http://www.investopedia.com/ask/answers/05/creditscorecalculation.asp

[11] https://en.wikipedia.org/wiki/Random_forest