Give Me Some Credit

By: Subhankari Mishra, Jatin Malviya, Anisha Bhatla, Abhineet Kalra,Vinayak Raghupathy

Project Goal

- To predict if a bank customer will fall under financial distress in the coming two years by looking at the customer account history and comparing it with the history of other similar customers.

- To explore and experiment with various machine learning algorithms.

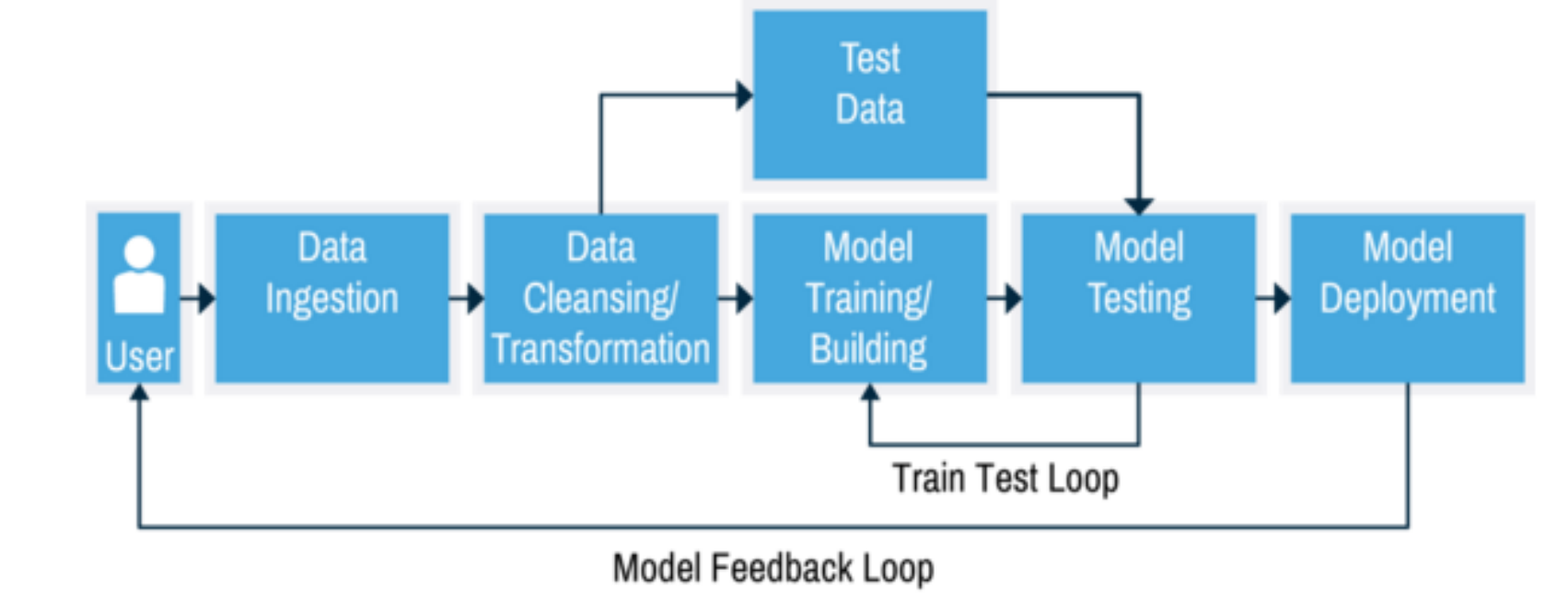
Motivation

- Should I take a 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.

- Also, 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 make a guess at the probability of default, are the method banks use to determine whether or not a loan should be granted.

Project Workflow



Technology

- Framework:
  - Spark 1.6.0
- Programming Language:
  - Scala 1.6.0
- Libraries:
  - Spark MLlib 1.6
- Tools:
  - RapidMiner 7.0
  - Knime 3.1.1
- Deployment Platforms:
  - Windows PC
  - NYU HPC Dumbo cluster
  - Cloudera Quickstart VM 5.5

DataSet: Kaggle

- Dataset downloaded from Kaggle.com
- Training and Test Set: 150,000 entries of public data
- Data Size: Test + Train = 7.21 MB.
- Data Structure: 11 attributes (age, income, debt ratio, no. of. Dependents, default history, no. of loans and their status etc.)

Process Overview

- Data mining model used: CRISP-DM (Cross Industry Standard Process for Data Mining)
  - Business Understanding
  - Data Understanding
  - Data Preparation
  - Modelling
  - Evaluation
  - Deployment

Business Understanding

- Banking Sector thrives on loans.
- Critical decisions regarding loan requests is needed.
- Decisions are made using the popular metric Credit Score.
- Most widely known Credit Scoring formula is FICO.
- Consideration of other factors is required to provide a more accurate prediction.

Data Understanding

- Explored the Dataset which was structured.
- Understood the data types, their meaning and importance w.r.t business.
- The process looped back to Business Understanding to get a in-depth view of data.
- Primarily the data had the below fields:
  - SeriousDlqin2yrs -> Person experienced 90 days past due delinquency or worse
  - 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
  - Age -> Age of Borrower in Years
  - NumberOfTime30-59DaysPastDueNotWorse -> Number of times borrower has been 30-59 days past due but no worse in the last 2 years
  - DebtRatio -> Monthly debt payments, alimony,living costs divided by monthy gross income
  - Some of other fields: MonthlyIncome, NumberOfOpenCreditLinesAndLoans, NumberOfTimes90DaysLate, NumberRealEstateLoansOrLines, NumberOfTime60-89DaysPastDueNotWorse, NumberOfDependents.

Data Preparation

Implemented various data cleaning, data reduction and feature selection techniques and their combinations:

- Replace Missing Values
- Low Variance Filter
- High Correlation Filter
- Missing Values Ratio
- Backward Feature Elimination
- Forward Feature Selection
- Singular Value Decomposition

Data Preparation

- Low Variance Filter

Preprocessing Method0	Accuracy	Notes
Low Variance Filer	92.21%	Rapid Miner Used, Min. Deviation=1
Low Variance Filer	93.33%	Rapid Miner Used, Min. Deviation=2

Data Preparation

- High Correlation Filter

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

Data Preparation

- Principal Component Analysis (PCA)

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

Data Preparation

- Singular Value Decomposition

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

Data Preparation

- 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

Modeling

- A person may or may not face financial distress in the coming years, classification is an approach of identifying whether the customer will default or not.

- Below were algorithms used to train Kaggle data set.
  - Naïve Bayes Classification
  - Decision Tree
  - Ensemble Methods
    - Random Forest
    - Gradient Boosted Trees
  - Support Vector Machines
  - Multilayer Perceptron

Multinomial Naïve Bayes

- Multinomial Naïve Bayes Independence Assumptions $P(x_1, x_2, \ldots, x_n | c)$
  - Bag of Words assumption: Assume position doesn't matter
  - Conditional Independence: Assume the feature probabilties $P(x_i | c_j)$ are independent given the class c.

    $P(x_1, \ldots, x_n | c) = P(x_1|c)*P(x_2|c)*P(x_3|c)*\ldots*P(x_n|c)$
  - $C_{MAP} = \underset{c \in C}{\operatorname{argmax}} P(x_1, x_2, \ldots, x_n|c)P(c)$
  - $C_{NB} = \underset{c \in C}{\operatorname{argmax}} P(c_j) \prod P(x|c)$

$$\hat{P}(c) = \frac{N_c}{N}$$

$$\hat{P}(w|c) = \frac{count(w,c)+1}{count(c)+|V|}$$

	Doc	Words	Class
Training	1	Chinese Beijing Chinese	c
	2	Chinese Chinese Shanghai	c
	3	Chinese Macao	c
	4	Tokyo Japan Chinese	j
Test	5	Chinese Chinese Chinese Tokyo Japan	?

Priors:

$P(c) = \frac{3}{4}$

$P(j) = \frac{1}{4}$

Conditional Probabilities:

P(Chinese|*c*) = (5+1) / (8+6) = 6/14 = 3/7

P(Tokyo|*c*) = (0+1) / (8+6) = 1/14

P(Japan|*c*) = (0+1) / (8+6) = 1/14

P(Chinese|*j*) = (1+1) / (3+6) = 2/9

P(Tokyo|*j*) = (1+1) / (3+6) = 2/9

P(Japan|*j*) = (1+1) / (3+6) = 2/9

Choosing a class:

$P(c|d5) \propto 3/4 * (3/7)^3 * 1/14 * 1/14 \approx 0.0003$

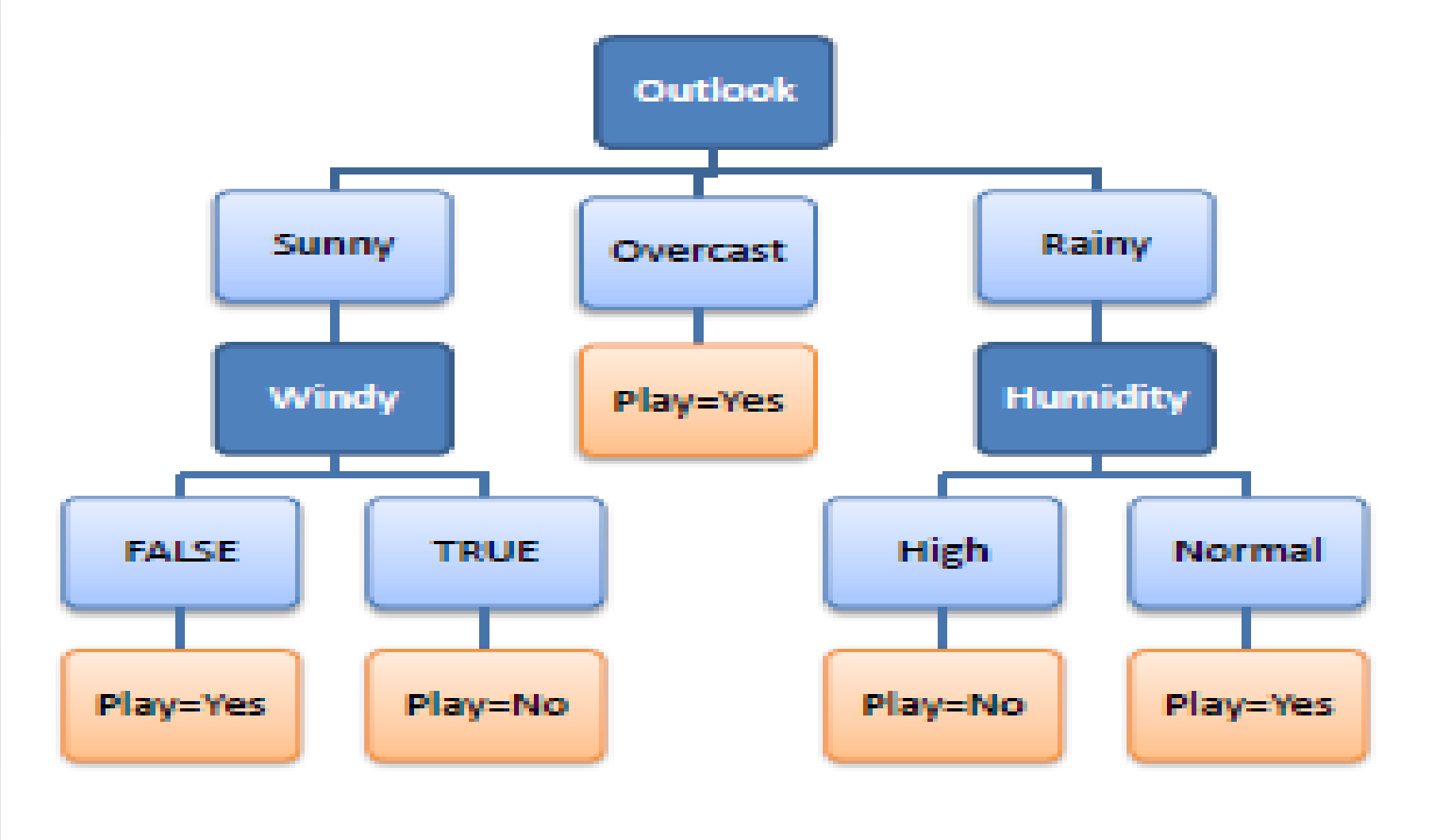
$P(j|d5) \propto 1/4 * (2/9)^3 * 2/9 * 2/9 \approx 0.0001$

Decision Tree

- Decision trees are popular methods for the machine learning tasks of classification.
- Decision trees are widely used since they are easy to interpret, handle categorical features, extend to the multiclass classification setting, do not require feature scaling, and are able to capture non-linearities and feature interactions. Tree ensemble algorithms such as ID3, random forests and boosting are among the top performers for classification and regression tasks.

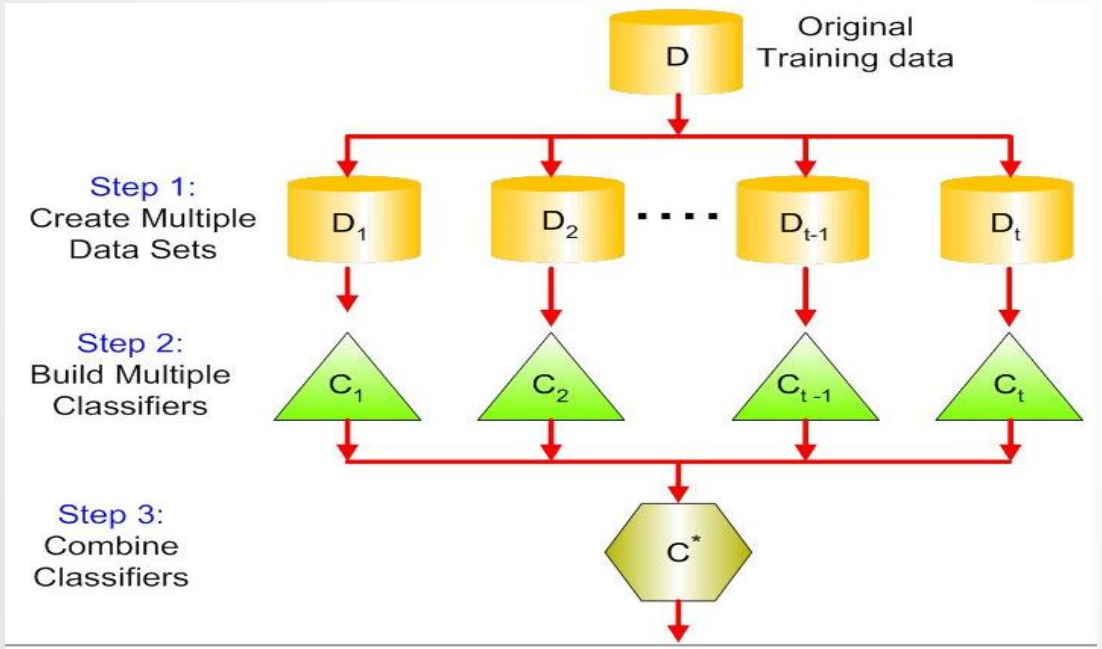
Decision Tree

- The decision tree is a greedy algorithm that performs a recursive binary partitioning of the feature space. The tree predicts the same label for each bottommost (leaf) partition. Each partition is chosen greedily by selecting the *best split* from a set of possible splits, in order to maximize the information gain at a tree node.
- The results of decision tree generated by ID3 can be used to predict the future values.
- **Information Gain:** Information Gain measures the difference in entropy from before to after the set S is split on attribute A. Since we want to reduce the uncertainty the most, we take the one with the biggest information gain at every iteration.


Outlook
Sunny
Overcast
Rainy
Windy
Play=Yes
Humidity
FALSE
TRUE
High
Normal
Play=Yes
Play=No
Play=No
Play=Yes

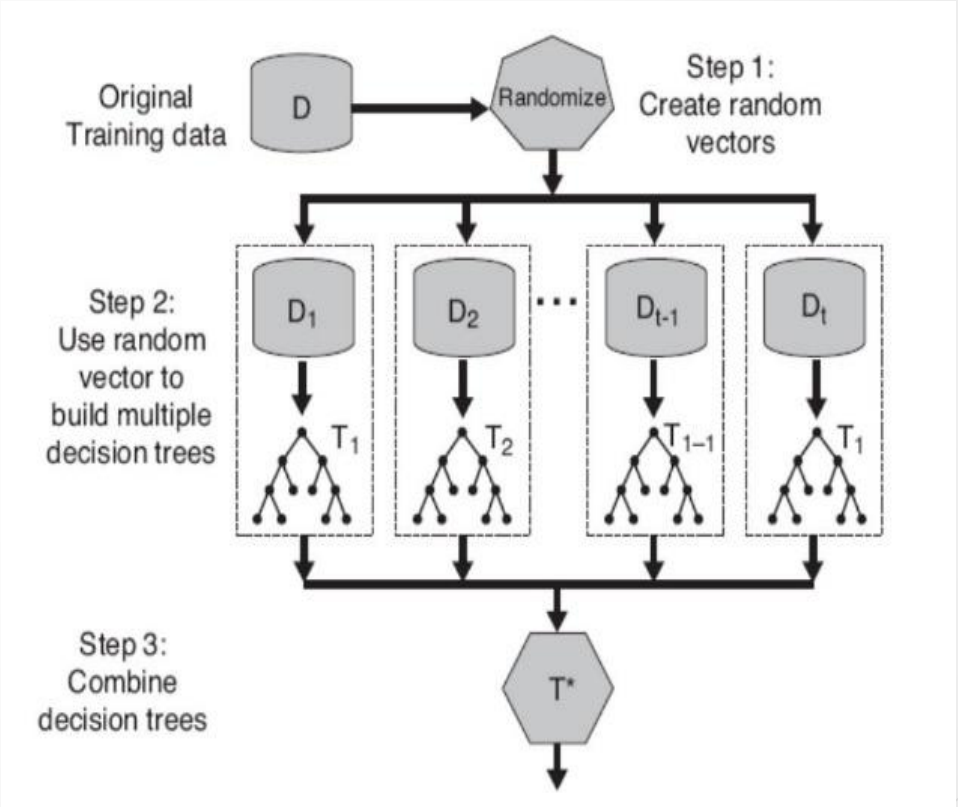
Ensemble Methods

- Ensemble methods use multiple learning algorithms to obtain better predictive performance than those which could be obtained from any of the constituent learning algorithms.



Random Forest Classifier Model

- 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.
- 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.



Gradient Boosted Trees

- Ensemble method that uses weighted sums of regressors to predict an overall better regressor
- Begins with a simple regression model
- Subsequent models are trained sequentially to predict the errors made by the model so far

Gradient Boosted Trees

- Gradient boosted 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.
- The specific mechanism for re-labeling instances is defined by the loss function.

Support Vector Machines (SVM)

- Supervised learning method primarily for linearly separable data can be extended for higher dimension using the kernel trick.
- Goal:
  - Select one hyper plane that maximizes the margin of separation
- Objective Function:

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

  Where H_1 is the Hinge Loss.
- Used SVM with SGD for the project.
- SGD:
  - Initializes weight vector with some random value
  - Sees one input sample at a time
  - Update the weight vector in the direction of steepest decrease of the objective function.

Multilayer Perceptron(MLP)

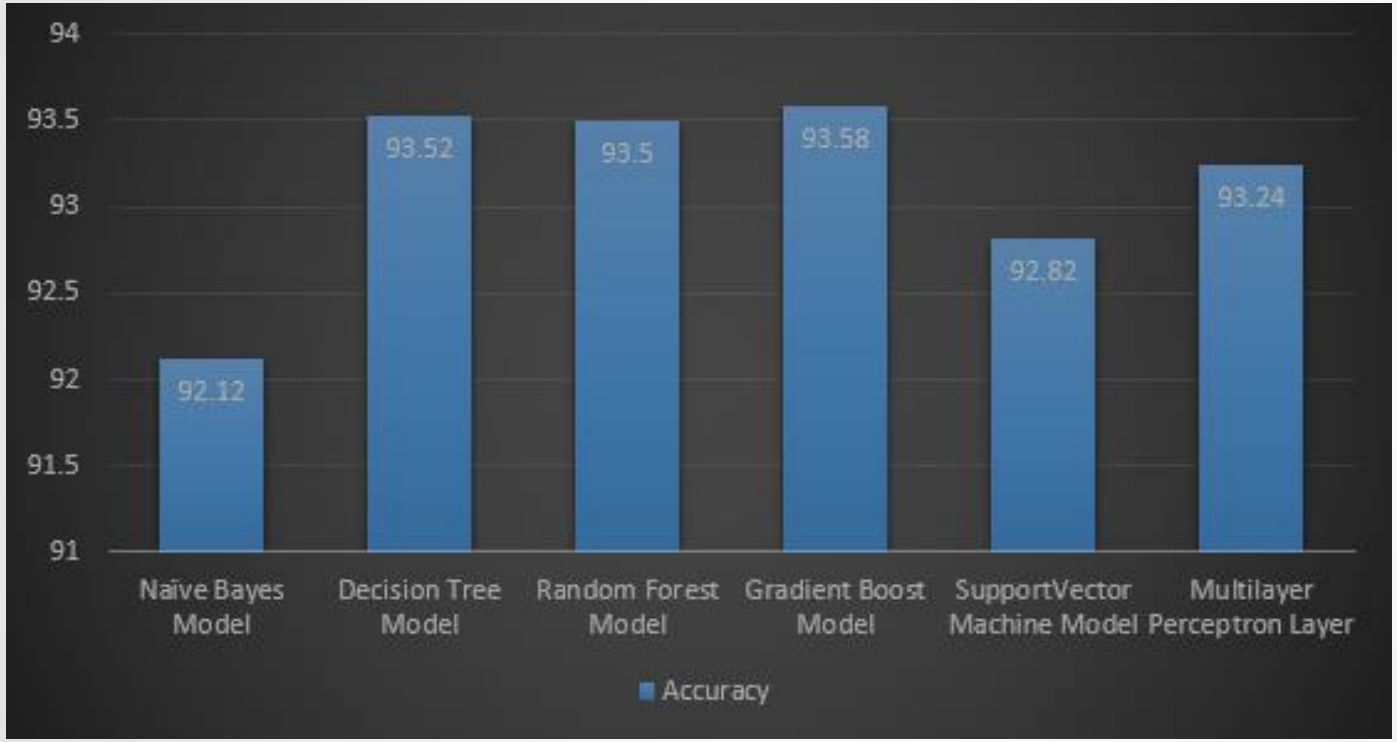
- Perceptron:
  - Similar to Gradient descent uses a initial weight vector.
  - Updates the weight vector for misclassification errors.

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

Where η is the learning rate

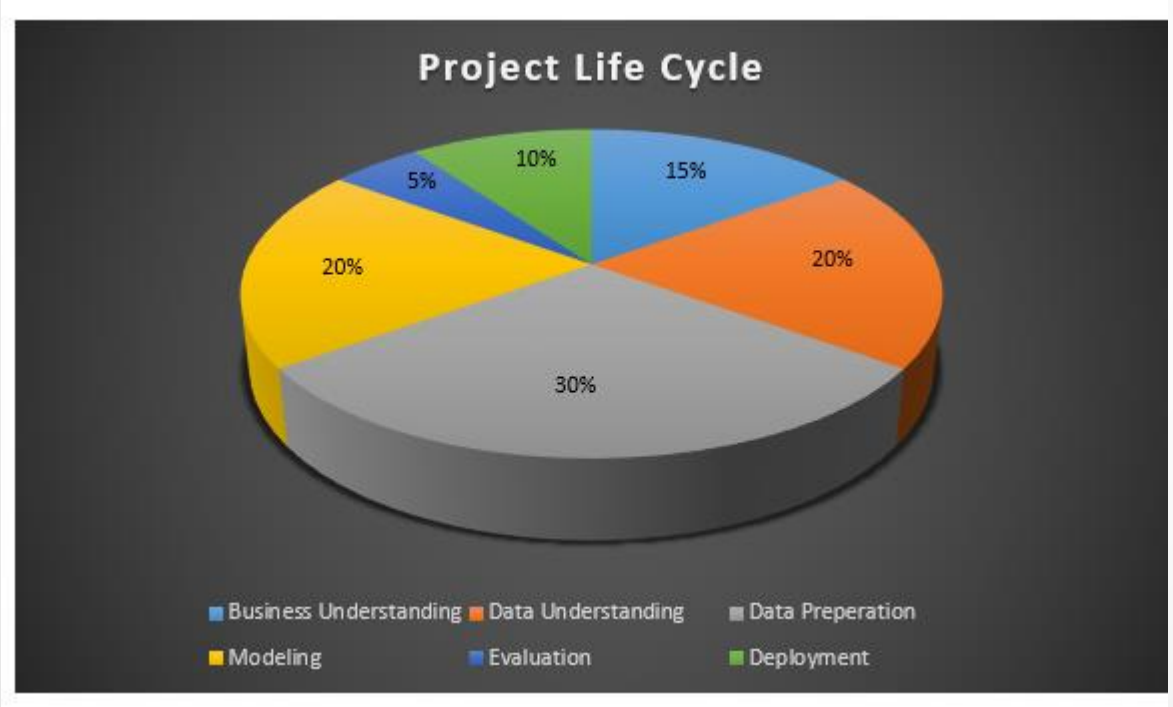
- MLP:
  - Feedforward artificial neural network model that maps sets of input data onto a set of appropriate outputs.
  - Modification of standard linear perceptron
  - Able to distinguish data that are non-linearly separable.
  - Uses Backpropagation for training the network.

Evaluation and Deployment



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

Overall Project Life Cycle



Learning Outcomes

- Current technologies like Spark MLlib.
- Language Scala.
- Classification algorithms work and their pros and cons.
- Applying CRISP DM for efficient development of project.
- Using different data preprocessing techniques.
- Implement tools like Rapidminer, Knime for data preprocessing

Obstacles

- Setting up the HPC account and system in our machines for the first time.
- Working on Quickstart VM which was slow due to high system memory requirement.
- Learning on a new language like Scala took a while.

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

Thank you !