Credit Card Risk Classification:
Logistic Regression vs. Ensemble Learning

Aleksandar Dimitrov
October 12, 2023

Table of Contents:

Introduction:

- In the context of our analysis, we have conducted an empirical comparison based on the following GitHub project: Credit Card Risk Classification: Logistic Regression vs. Ensemble Learning

- The empirical comparison is structured around the specific GitHub project, aiming to evaluate its performance and functionality

- The empirical comparison serves as a critical component of our evaluation process

Ensemble Learning: A look at the Landscape:

Multiple Perspectives

- Ensemble Methods utilize multiple models, not just a single one, to make predictions
- They lead to better generalization performance
- By aggregating predictions from diverse models, they can capture underlying patterns in the data more effectively

Increased Security and Reliability

- Achieving greater security through assessment from various angles

Reduced Subjectivity

- Avoiding biases associated with a single model

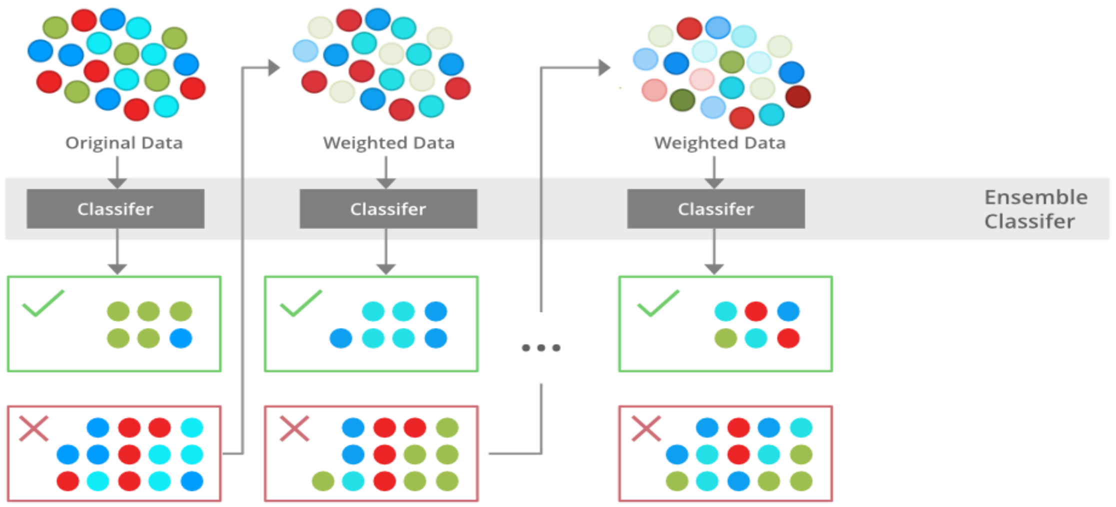
Robustness to Model Variability

- Even if some of the individual models have variations in their performance, the ensemble as a whole can still provide reliable predictions

Error Reduction (Improving Accuracy)

- Ensemble Methods reduce errors in predictions by combining multiple models
- Improvement in the accuracy of predictions
- Diversity of models compensates for common errors in individual models
- Ensemble Methods can reduce the impact of noise or outliers in the data, leading to more stable and reliable predictions

Boosting in Machine Learning

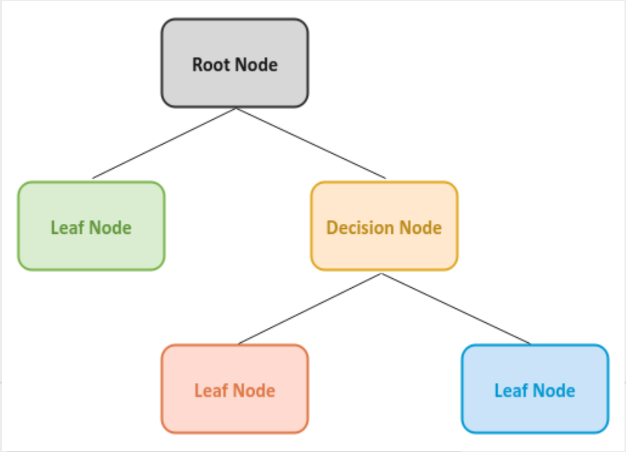


Ensemble Learning:

Definition and Types

Decision Trees

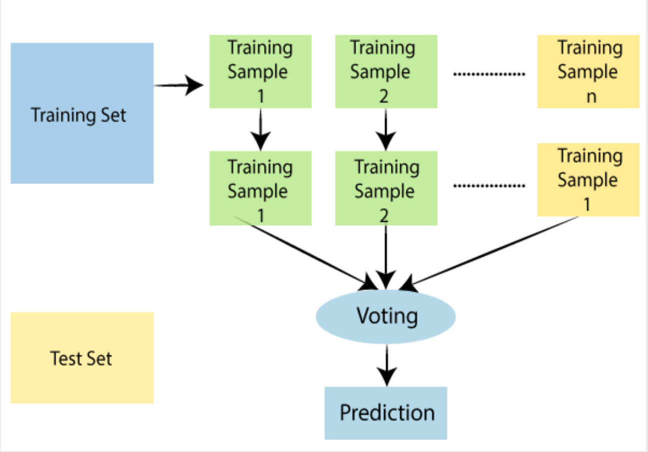
Simple Path to Classification



- **Hierarchical Questioning:** Forms a tree structure with input-related questions
- **Classification Focus:** Primarily for classifying data
- **Visual Clarity:** Easy-to-visualize structure
- **Interactive Decision Path:** Users answer questions for classification

Random Forest

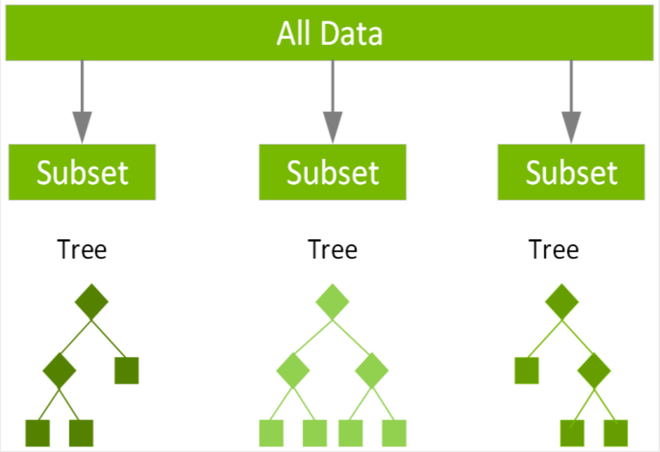
Forested Accuracy



- **Decision Tree Ensemble:** It's a combination of multiple decision trees
- **Result Combining:** The final result is based on averaging or majority voting from the trees
- **Bootstrap Aggregating:** It creates diverse datasets and assigns them to trees
- **Accuracy Enhancement:** Its aim is to improve accuracy and mitigate overfitting

XGBoost

Extreme Gradient Precision

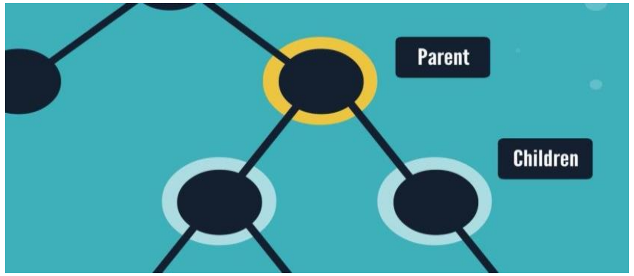


- **Sparse Data Handling:** XGBoost excels with sparse data
- **Weighted Trees:** It employs weighted trees for better predictions
- **Scalable:** Efficient with large datasets
- **Auto Feature Selection:** Automatically selects essential features

Decision Trees:

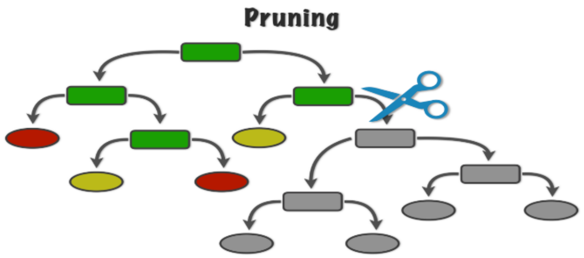
Structure and Operating Principles

Decision Trees Model



$$\mathit{Gini}\ \text{Impurity} = \sum p_i(1 - p_i) = 1 - \sum(p_i)$$

$$\text{Entropy} = \sum p_i \log_2 (p_i)$$



Structure and Operating Principles

Construction:
- Build Node by Node
- Nodes selected for the best data split

Gini Impurity:
- Measures misclassification in a node
- Lower Gini impurity indicates better split

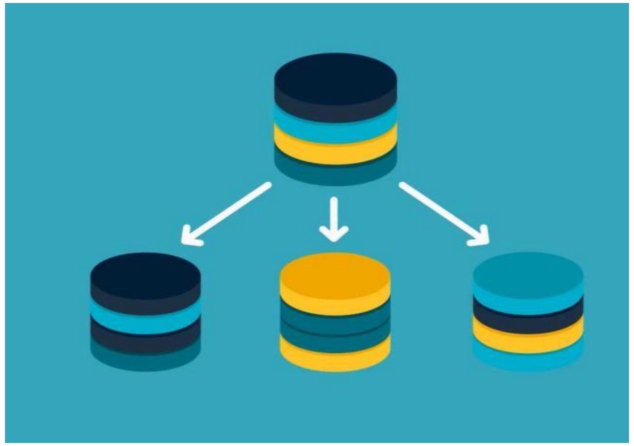
Information Gain (Entropy):
- Measures information in child node
- DT aim to maximize information gain

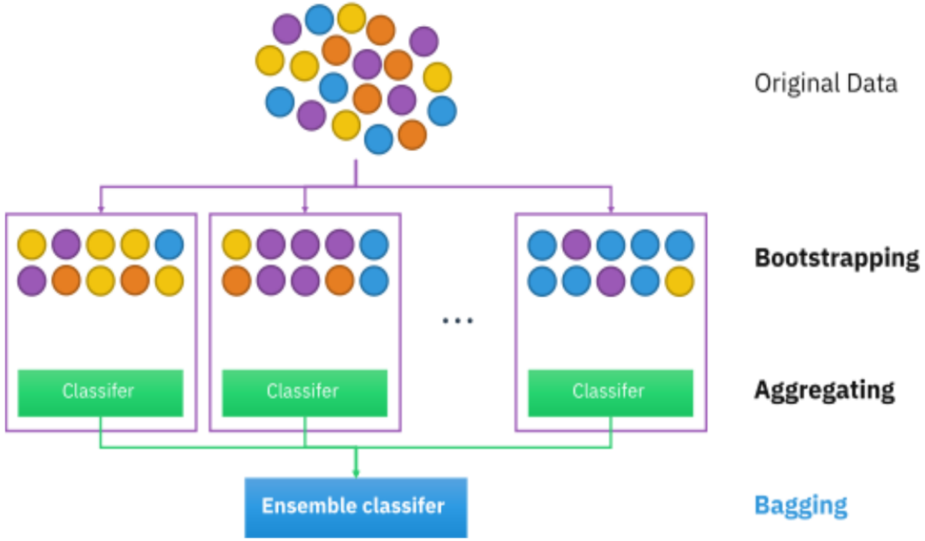
Pruning to prevent Overfitting:
- DT tend to overfit
- Pruning reduces complexity by removing branches

Random Forest:

Structure and Operating Principles

Random Forest Model





Structure and Operating Principles

Collective wisdom:

- Ensemble of multiple decision trees
- Combining different learning algorithms

Majority Voting:

- With multiple DT, we get multiple answers
- Selecting the most common outcome

Bootstrapping:

- Randomly sampling the original dataset with replacement
- Feature subsets are chosen randomly for each tree

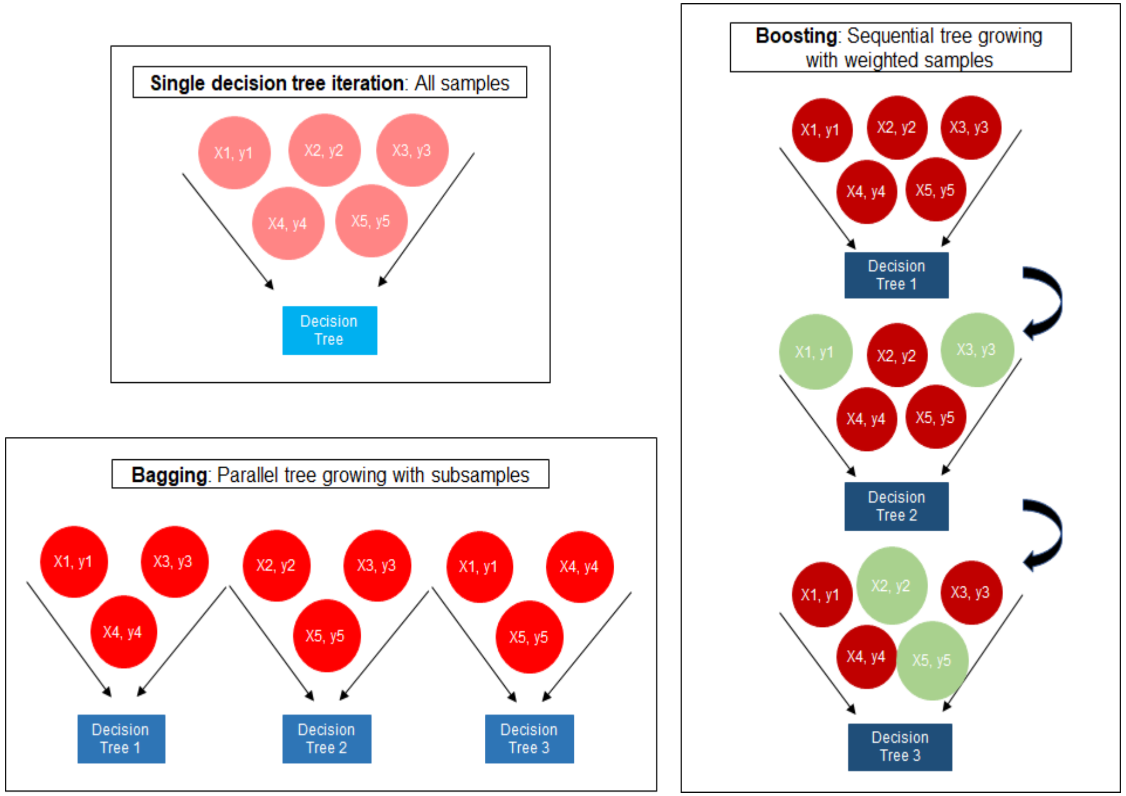
Random Forest:

- Bootstrap Aggregated decision trees
- Bootstrapping is type of regularization

XGBoost:

Structure and Operating Principles

XGBoost Model



Structure and Operating Principles

Boosting:
- Predictive accuracy by combining different models
- Model optimizing by trial and error

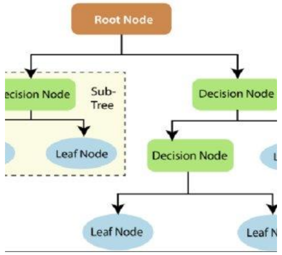
Gradient Boosting:
- Gradient boosting algorithm implementation
- Employing DT with limited depth as weak learners

Iterative Model Construction:
- XGBoost is build iteratively
- Each iteration improving upon the errors of the previous ones

Objective Function and Regularization (for Regression only):
- Objective function defining
- L1(Lasso) and L2(Ridge) incorporating

Exploring Ensemble Methods:

Decision Trees



Applications and Use Cases

Most recent characteristics

Used For
- Regression
- Classification

Input
- Numerical
- Categorical

Handles
- Small Dataset
- Large Dataset
- Sparse data
- Hight Dimensions

Preprocess
- No preprocessing required

Algorithm Speed
- Training: Fast
- Testing: Really Fast

Avoid Overfitting ?
- Perform pruning (during or after the process)

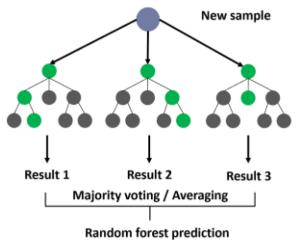
Pros
- Simple to understand and interpret
- In build feature selection
- Fast testing on new data

Cons
- Sensitivity to Small Data Variations
- Prone to Overfitting
- Limited Optimization Options

Application
- Credit card fraud detection
- Credit risk application and behave models
- Medical diagnoses

Exploring Ensemble Methods:

Random Forest



Applications and Use Cases

Most recent characteristics

Used For
- Regression
- Classification

Input
- Numerical
- Categorical

Handles
- Small Dataset
- Large Dataset
- Sparse data
- Hight Dimensions

Preprocess
- No preprocessing required

Algorithm Speed
- Training: Fast
- Testing: Fast

Avoid Overfitting ?
- Prune the individual trees

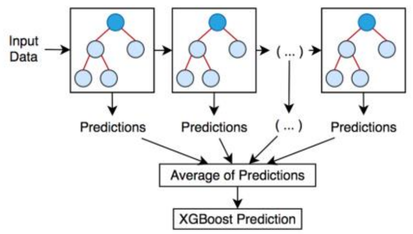
Pros
- Performs well with large dataset
- Lots of hyperparameters to control
- Relatively better results than decision trees

Cons
- Less interpretable than decision trees
- Doesn’t solve regression problems well
- Outperformed by gradient-boosted trees

Application
- Credit card fraud detection
- Credit card application and behave models
- Medical diagnoses

Exploring Ensemble Methods:

XGBoost



Applications and Use Cases

Most recent characteristics

Used For
- Regression
- Classification

Input
- Numerical
- Categorical

Handles
- Small Dataset
- Large Dataset
- Sparse data
- Hight Dimensions

Preprocess
- No preprocessing required

Algorithm Speed
- Training: Fast
- Testing: Fast

Avoid Overfitting ?
- Resistant to overfitting

Pros
- Easy to parallelize
- Smart trees penalize
- Easily scalable

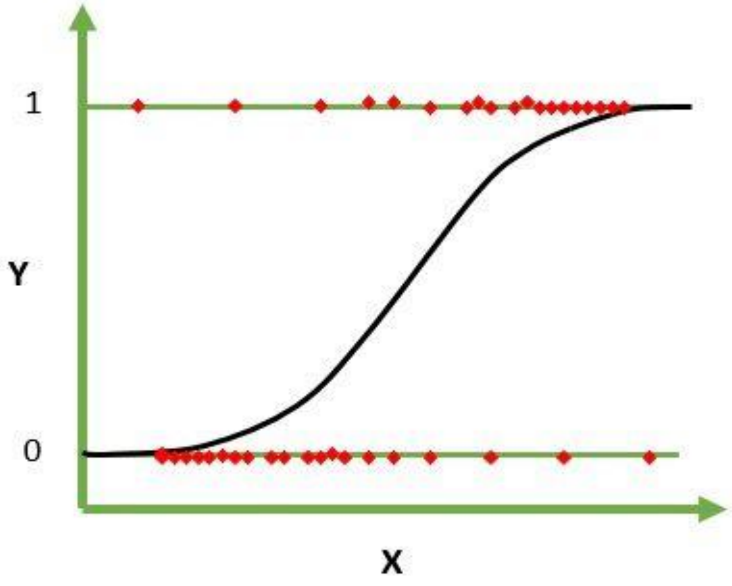
Cons
- Lack of interpretability
- Difficult to optimize the many different parameters

Application
- Credit card fraud detection
- Credit card application and behave models
- Store sales predictions

Logistic Regression:

Definition and Limitation

Logistic Model



The logistic regression predicts the probability of an event occurring

Logistic regression model

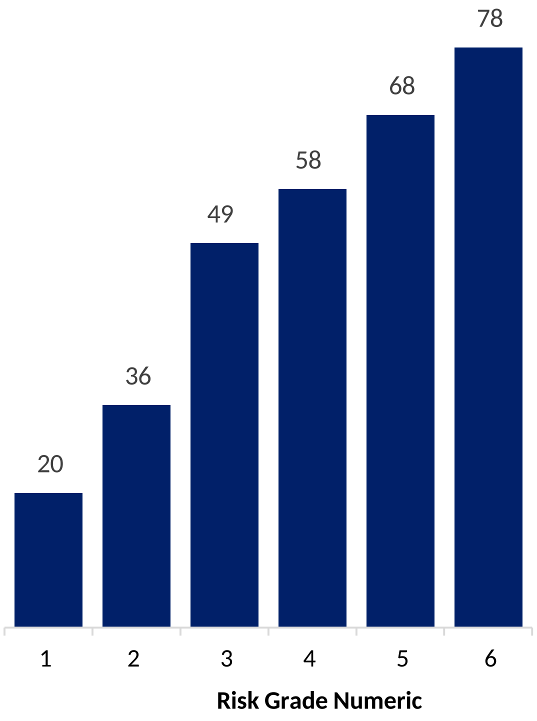
$$\log(\text{odds}) = \boldsymbol{\beta}_0 + \boldsymbol{\beta}_1 \boldsymbol{x} + \cdots \boldsymbol{\beta}_k \boldsymbol{x}_k$$

Logistic Regression Constraints

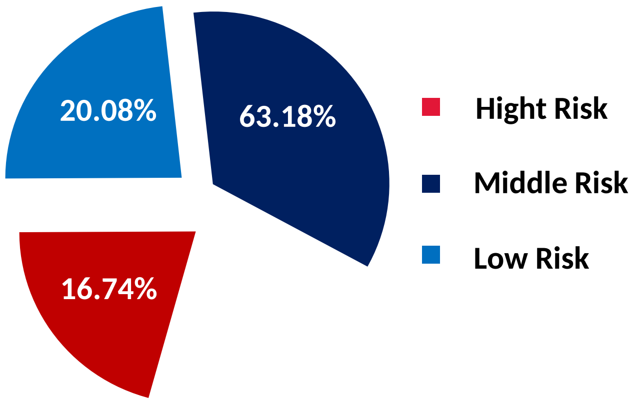
- **Assumes Linearity:** Logistic regression assumes a linear relationship, which can't capture complex non-linear patterns
- **Binary Classification Bias:** May need modifications for multi-class problems
- **Overfitting Risk:** Susceptible to overfitting
- **Independence Assumption:** Assumes observations are independent, which may not hold in time series or spatial data
- **Outlier Sensitivity:** Outliers can strongly influence results
- **Handling Missing Data:** Logistic regression struggles with missing data and imputation can introduce bias

Model Outcome and Risk Segmentation:

Bad Rate after activation



Risk Class Segmentation



Analyzing Model Output

- **Model Results:** An overview of model outcomes and output probabilities
- **Risk Segmentation:** How the model divides data into segments based on default probability
- **Customer Classification :** Categorizing customers from low-risk (marked as 1) to high-risk (marked as 6-15 classes)
- **Segment Assessment:** Evaluating risk segments(Low, Middle and Hight Risk), including calculating the percentage of bad customers
- **Model Application:** How to apply the model's output for new data assessment and decision-making

Application Models vs. Behavioral Models for Credit Risk:

ML models leverage data and automation to improve decision-making and efficiency in various contexts

Machine Learning Assessment Models

Application Models

- Assessing New Customers
- Using External Sources: Credit Bureaus, Open Banking
- Application Form

Customer Risk Profile

Behave Models

- Assessing Existing Customers
- Behavioral Analysis
- Payment History Evaluation
- Portfolio History

Comprehensive Credit Assessment with Application and Behavioral Models:

For Individuals

Models Performance Evaluation:

KEY Measures

What numbers tell us...

Confusion Matrix and Classification Report

01

CM is **critical tool** in model evaluation. CR includes metrics like precision, recall, F1-score, and support for each class

AUC (Area under the Curve)

02

Intuitive Interpretation: A higher AUC means better client classification by the model

Gini Coefficient

03

Gini is calculated as
Gini = 2*AUC - 1

It's Working...

For Credit Risk models, higher Gini values are preferred

75%+

For Application models
Gini > 50% implies effective model, correctly identifies ~75% clients

50%

For Application models
Gini > 50% is desirable

30%

For Behave models
Gini > 60% is desirable, for even higher accuracy

Application:

Performance Measure	Description
Accuracy = (TN + TP) / TN + FN + FP + TP	The ratio between the number of all correctly predicted samples and the number of all samples
Precision = TP / TP + FP	The ratio between the number of true positives and the number of all samples classified as positive
Recall = TP / TP + FN	The ratio between the number of true positives and the number of all samples whose true class is the positive one
F_1Score = 2 / (1/precision + 1/recall)	The harmonic mean of precision and recall

THANK YOU !