School of Computing and Information Systems

CS610 Applied Machine Learning

Credit Card Fraud Prediction Project Report (Group 5)

Group Members (G2):

FAKHRUR RAZI
PARK HEE JAE
DEBANJAN DATTA
WOO JIA JIAN
AYUSHI SHAKYA

Project Objective and Solution

Our project on Credit Fraud Prevention focuses on implementing strategies and technologies to proactively identify and mitigate fraudulent activities in credit transactions. By leveraging machine learning algorithms and data analysis, the project aims to develop a robust system that can detect unusual patterns and suspicious behaviors indicative of fraud. The broad objectives can be summarized as follows:

Security Enhancement:

- Rule-based and manual processes are slow with low accuracy.
- ML allows real-time, adaptive, and accurate fraud detection to heighten protection levels

Accuracy Boost:

- Enhance model effectiveness by integrating advanced feature engineering techniques and fine-tuning algorithms, resulting in more reliable identification of fraudulent transactions

Detection Evolution:

- Continuously refine the model through iterative learning processes, enabling it to adapt and evolve alongside emerging fraud tactics and techniques

Given that the business problem is to detect fraudulent credit card transactions, we started off by envisioning a machine learning model that can identify if a particular credit card transaction is fraudulent or not with reasonably high accuracy scores and relevant metrics, given a dataset at transaction level. This was a binary classification problem, where the target variable is yes/no (in relation to fraud or not), while the dependent variables can vary from amount spent, merchant-related data such as frequency of spending at a particular merchant within a predefined period etc.

Data Source

Our data was obtained from Kaggle:https://www.kaggle.com/datasets/kartik2112/fraud-detection

The simulation process was conducted utilizing the Sparkov Data Generation tool developed by Brandon Harris. This tool facilitated the generation of synthetic transactional data by defining a range of merchants, customers, and transaction categories.

Our data contains 22 columns and ~1 million rows pertaining to credit card information of 1,000 customers doing transactions with a pool of 800 merchants.

Data Generation Process

1. **Initialization:** The simulation begins by initializing a set of merchants, customers, and transaction categories within the Sparkov tool.
2. **Parameter Definition**: Using the Python library "faker," along with user-provided parameters such as the number of customers and merchants, an intermediate list is generated. This list serves as the foundation for subsequent transaction creation.
3. **Profile Definition:** Users define specific profiles, such as "adults 25-50 female rural," which encapsulate characteristics like age range, gender, and location. These profiles are represented in JSON files containing parameters such as minimum and maximum transactions per day, distribution of transactions across days of the week, and statistical properties for transaction amounts in various categories.
4. **Transaction Generation:** Leveraging the predefined distributions and parameters, the simulation generates transactions that align with the specified profiles. Transactions are generated using faker to ensure realism and adherence to typical transactional patterns.
5. **Data Integration:** To enhance the authenticity of the simulated data, transactions are generated across multiple profiles and subsequently merged. This amalgamation results in a comprehensive dataset that closely mirrors real-world transactional activity.

Data Preprocessing and Feature Engineering

The data pre-processing steps can be summarized as follows:

- **Data Cleaning:** Since our dataset was simulated we found no missing values and duplicate values. At this stage we also converted Categorical data such as Gender to binary values

- **Train Test Split:**
  - Initial dataset was already pre-split to train and test
  - K stratified resampling was done to ensure a 20% test data,10% validation data and 70% train data

	Train Set	Test Set
Row Count	1,296,675	555,719
Column Count	43	

- **Imbalanced Dataset:** Due to a highly imbalanced dataset, we used SMOTE (Synthetic Minority Over-sampling Technique) to address this issue.

Fraud in Train set After SMOTE	Fraud in Test Set after Train-Test Split
50%	0.39%

Feature Engineering:

Some features such as amount spent were used in their original form while others were transformed. New features were created to emulate the markers of credit card fraud in real life as summarized in table below:

Feature	Description
Amt	Transaction Amount
Gender_Numeric	Encoded gender of customer
City_pop	Population of the customer city
Cc_count	Count of credit cards per customer.
address_multiple_customer_flag	Checks if an address has more than one customer.
fraud_likelihood_merchant	Checks the % of fraud transactions by each merchant out of total transactions by the merchant. Flag if % is more than 1.
fraud_likelihood_category	Checks if an item category belongs to high, medium or low risk. Categories with >20% fraud transactions are high risk. Categories with 5-20% fraud transactions are medium risk
transaction_distance	Calculates distance between merchant and customer for transaction
fraud_likelihood_job	Checks if a job category belongs to high, medium or low risk. Categories with >2% fraud transactions are high risk. Categories with >1% fraud transactions are medium risk
fraud_likelihood_age	Classifies age-groups into highly likely and less likely with respect to fraud likelihood
address_multiple_card_flag	Checks if multiple cards are linked to the same address
multiple_cards_large_orders	Checks if large orders are made through multiple cards by one person
first_time_shopper	Checks if a customer is a first time shopper or not

Models and Performance

We tried out different models and a summary of key performance metrics is summarized in the table below. As per our research and industry experience, Logistic regression is a popular choice among banks for model deployment due to its ease of explainability.

Algorithm	Key Metrics					
	Accuracy	Recall	Precision	AUC	F1 Score	F2 Score
Logistic Regression	0.900	0.744	0.028	0.845	0.054	0.123
Decision Tree	0.982	0.485	0.104	0.735	0.172	0.280
Random Forest	0.983	0.628	0.135	0.935	0.222	0.363
XG Boost	0.948	0.771	0.054	**0.958**	0.102	0.213
Ada Boost	0.918	**0.825**	0.037	0.949	0.072	0.159
Multi-layer Perceptron	**0.996**	0.569	**0.439**	0.948	**0.496**	**0.537**

We tried to explore hyper parameter tuning for our models by using various search methods to converge on the best hyper parameter. The methods we tried were Grid search, Halving random search and Bayes search. However, neither of these enabled for a decent amount of computational time. As such we settled for Randomized search which may not be as optimal and may have resulted in settling for a local optima point for the hyper parameters. This might have been the case for our tuned logistic regression model.

Algorithm	Key Metrics					
	Accuracy	Recall	Precision	AUC	F1 Score	F2 Score
Tuned Random Forest	0.977	0.703	0.111	0.957	0.191	0.340
Tuned Logistic Regression	0.901	0.744	0.028	0.845	0.054	0.123

The choice of model was determined based on precision and recall scores, as well as the explainability of the model. Given that the model users are from a financial organization that values explainability, the following considerations were made:

1. **Logistic Regression:** This model was considered for its simplicity and ease of explainability. It provides straightforward interpretations of coefficients, making it suitable for transparent decision-making processes.
2. **Adaboost:** Adaboost was identified as the preferred choice if the organization prioritizes catching more fraud cases, even at the expense of a higher number of false positives. Adaboost tends to focus on improving recall, which is crucial for detecting fraudulent transactions.
3. **Multilayer Perceptron (MLP):** MLP was suggested for organizations seeking a balance between the count of false positives and false negatives. While offering a more complex structure than logistic regression, MLP can provide a balance between precision and recall, thereby optimizing overall performance.

Considering the seriousness of credit card fraud and the project's goal to mitigate it, ***Adaboost emerges as a potentially suitable solution.*** Its high recall rate increases the likelihood of detecting fraudulent transactions, which aligns with the organization's objective to combat fraud effectively. However, the ultimate decision rests with the organization's priorities and strategies, and careful consideration should be given to the trade-offs between precision and recall.

Discussion on Feature Importance

The table below shows the top 4 important features for each model. "Amount spent" is the most significant feature for all models, which is intuitive as fraud offenders would likely prioritize their risk-reward ratio. Another common feature is city population, which gives insight into cities of certain populations where fraudulent transactions are more likely to occur.

Model/ Feature Rank	Logistic Regression	Tuned Logistic Regression	Decision Tree	Random Forest	Tuned Random Forest	XG Boost	Ada Boost
1	Amount Spent						
2		Fraud Likelihood Job	City Population			Fraud Likelihood Category	Lat-Long Distance
3			Lat-Long Distance	Large Order			Transaction Distance
4			Fraud Likelihood Category			Transaction Distance	Large Order

Conclusion and Future Scope

While our model has been performing well in predicting frauds, it can be further enhanced in the following aspects:

- **Real world Data:** We have currently used a simulated data set for our model building. Original data ensures integrity, enables comprehensive feature engineering, and allows robust model validation for reliable fraud prediction models.

- **Enhancing Fraud Detection with Additional Features:** While we used existing features and derived some new features to enhance our model additional features like failed attempts, transaction type can enhance fraud prediction by providing a more comprehensive view of transaction context.

- **Implications of High Recall and Low Precision:** This approach prioritizes minimizing missed fraud cases but can lead to increased operational costs and potential customer inconvenience due to the high number of false alarms.

- **Adjusting Fraud Detection Thresholds:** Our current model has a threshold of 50% There is further scope of adjusting fraud prediction models threshold to align with a financial institution's risk appetite, balancing fraud and false positives for optimal risk management

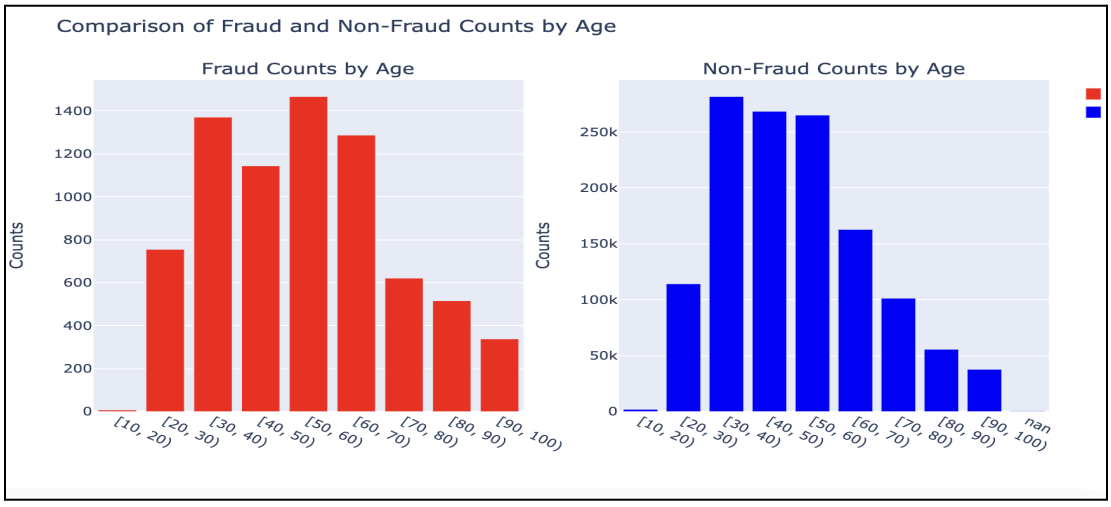
Appendix

Exploratory Data Analysis:

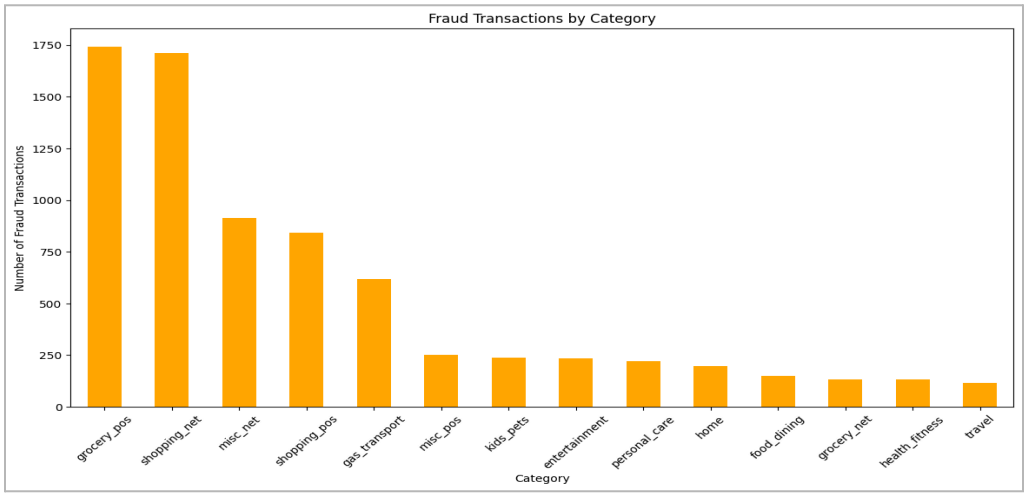
No missing values recorded in dataset,no need for imputation methods

```
 ---  ------                  --------------    -----
 0    Unnamed: 0              1296675 non-null  int64
 1    trans_date_trans_time   1296675 non-null  object
 2    cc_num                  1296675 non-null  int64
 3    merchant                1296675 non-null  object
 4    category                1296675 non-null  object
 5    amt                     1296675 non-null  float64
 6    first                   1296675 non-null  object
 7    last                    1296675 non-null  object
 8    gender                  1296675 non-null  object
 9    street                  1296675 non-null  object
 10   city                    1296675 non-null  object
 11   state                   1296675 non-null  object
 12   zip                     1296675 non-null  int64
 13   lat                     1296675 non-null  float64
 14   long                    1296675 non-null  float64
 15   city_pop                1296675 non-null  int64
 16   job                     1296675 non-null  object
 17   dob                     1296675 non-null  object
 18   trans_num               1296675 non-null  object
 19   unix_time               1296675 non-null  int64
 20   merch_lat               1296675 non-null  float64
 21   merch_long              1296675 non-null  float64
 22   is_fraud                1296675 non-null  int64
```

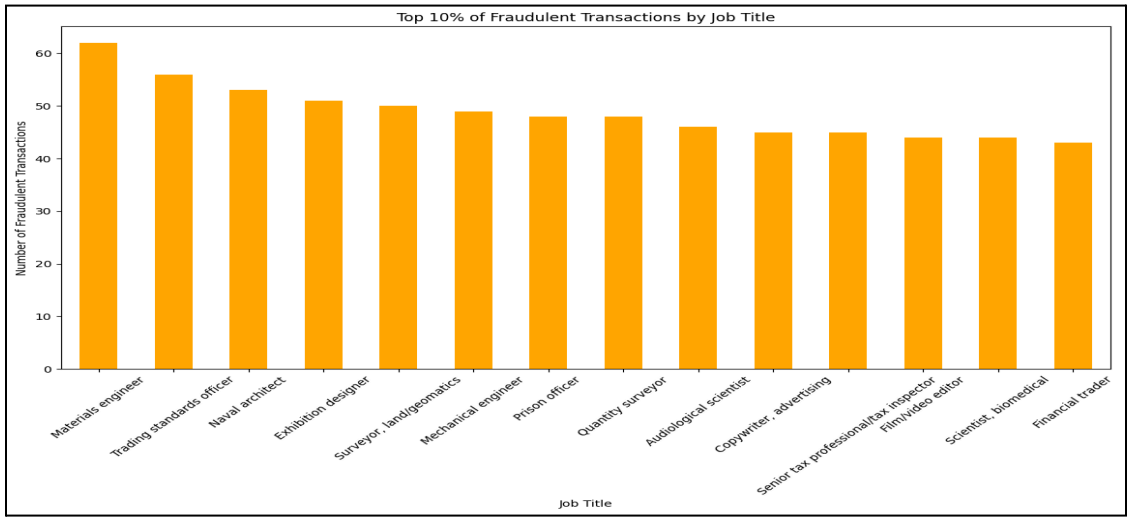
Plot of fraud/Non-fraud counts divided into age category (Binning)



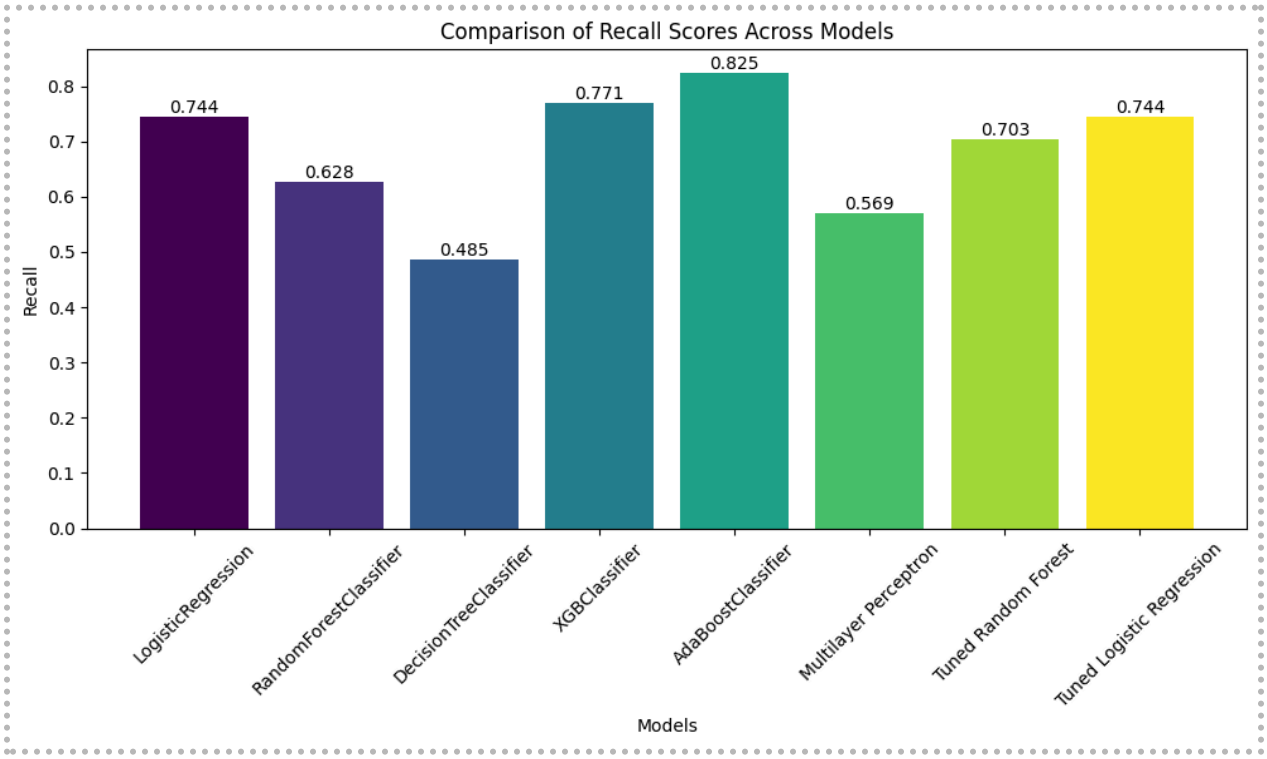
Top 25% of Frauds by category of merchant sorted in decreasing order

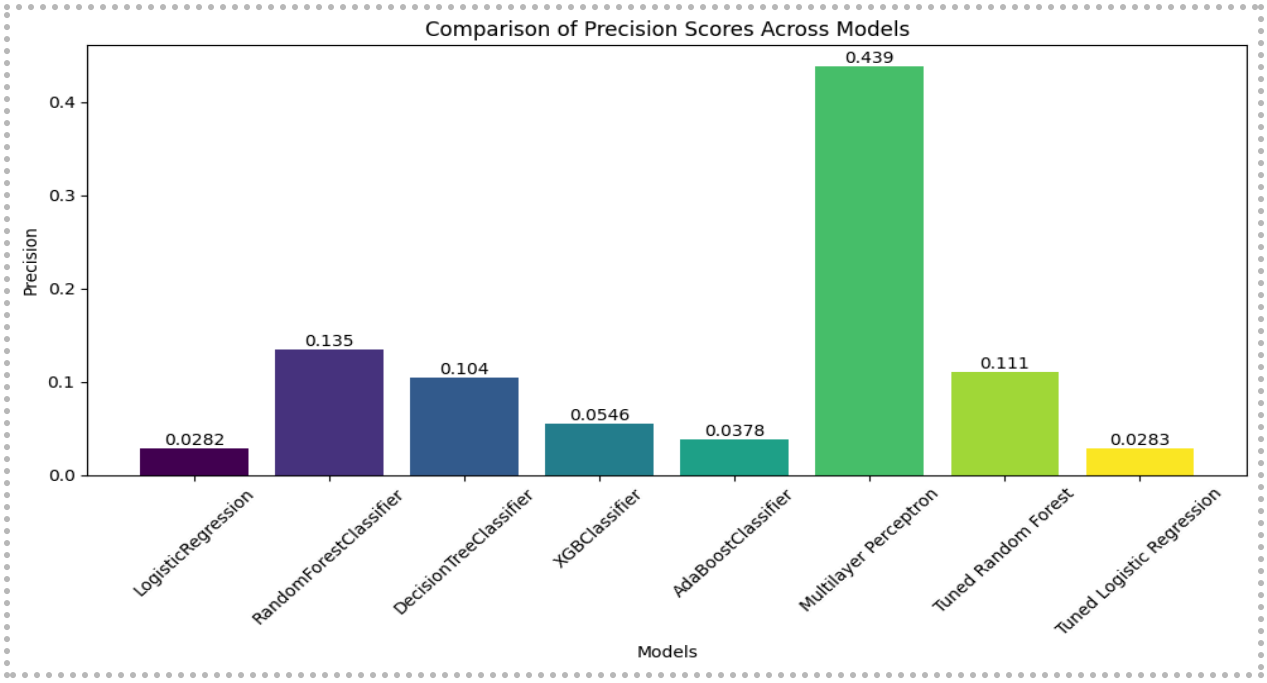


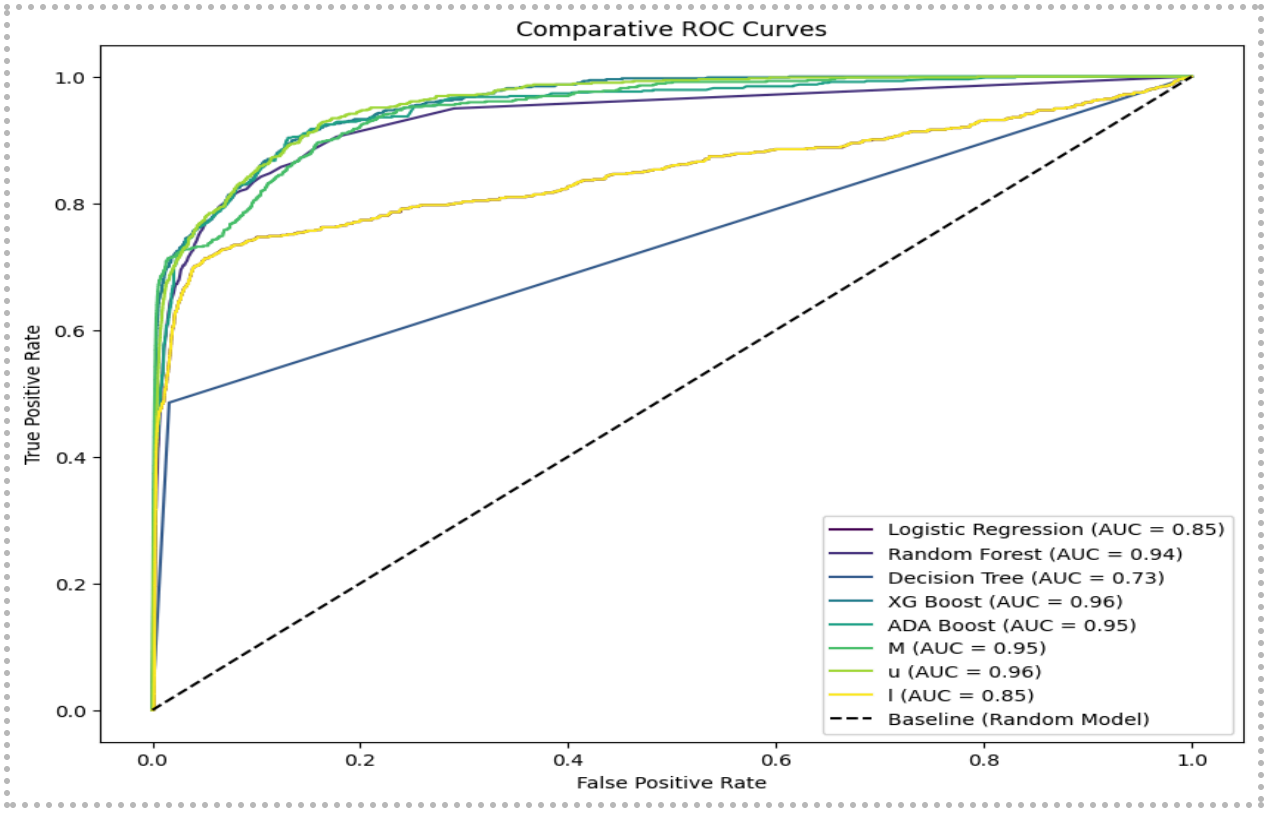
Top 10% of Frauds by jobs of consumer sorted in decreasing order



Performance Metrics:






Comparative ROC Curves
True Positive Rate
False Positive Rate
0.0
0.2
0.4
0.6
0.8
1.0
Logistic Regression (AUC = 0.85)
Random Forest (AUC = 0.94)
Decision Tree (AUC = 0.73)
XG Boost (AUC = 0.96)
ADA Boost (AUC = 0.95)
M (AUC = 0.95)
u (AUC = 0.96)
l (AUC = 0.85)
Baseline (Random Model)