Project Reports:
Credit Card Transaction Fraud Detection

1. Introduction

1.1. Background

Credit card fraud is a pervasive and growing issue in the digital age, where financial transactions are increasingly conducted online. Fraudulent activities not only result in significant financial losses but also erode consumer trust and strain the resources of financial institutions. According to recent studies, credit card fraud losses worldwide exceeded billions of dollars annually, with fraudsters continuously devising new methods to exploit vulnerabilities in payment systems.

This project aims to leverage the power of machine learning to develop a robust classification model for credit card fraud detection. By utilizing advanced algorithms and thorough data preprocessing techniques, the project seeks to build a model that can accurately distinguish between legitimate and fraudulent transactions, thereby reducing financial losses and enhancing the security of digital payment systems.

1.2. Problem statements

Credit card fraud is a significant issue in the financial industry, resulting in substantial monetary losses for both consumers and financial institutions. As the volume of digital transactions continues to rise, so does the sophistication of fraudulent activities. Detecting fraudulent transactions in real-time is challenging due to the sheer volume of transactions and the need to minimize false positives to avoid inconveniencing customers. This project aims to address this challenge by developing a robust classification model to accurately identify fraudulent credit card transactions, thereby helping to mitigate the impact of fraud on financial systems.

1.3. Objectives and goals

The primary objective of this project is to develop and evaluate a machine learning model capable of accurately classifying credit card transactions as either fraudulent or legitimate. The model should balance the need for high detection rates with the minimization of false positives to ensure practical usability in real-world scenarios.

Goals:

1. **Data Exploration and Preprocessing:** Gain a thorough understanding of the dataset by exploring the distribution of features, handling missing values, correcting data types, and addressing any imbalances in the data.
2. **Feature Engineering:** Identify and create relevant features that can improve the predictive power of the model.
3. **Model Development:** Train multiple machine learning models, including Decision Tree, Random Forest, K-Nearest Neighbors (KNN), and Support Vector Machine (SVM), to classify transactions.
4. **Model Evaluation:** Evaluate the performance of the models using metrics such as accuracy, precision, recall, F1-score, and AUC. Conduct cross-validation to ensure the model's generalizability.
5. **Model Optimization:** Fine-tune the best-performing model to improve its accuracy and robustness in detecting fraud.
6. **Deployment and Monitoring:** Discuss the potential for deploying the model in a real-world environment and outline a strategy for ongoing monitoring and updating of the model.

1.4. Scope and limitations

Scope:

- The project will focus on supervised learning techniques for binary classification of credit card transactions.
- The dataset used will be historical transaction data, which includes both fraudulent and legitimate transactions.
- The project will explore various machine learning algorithms and preprocessing techniques to improve the model's performance.
- The project will also include an analysis of the model's results, offering insights into the factors that contribute to fraudulent transactions.

Limitations:

- The model's performance is dependent on the quality and completeness of the data. Any biases or inaccuracies in the historical data could affect the model's predictions.
- The project will not cover real-time deployment of the model; instead, it will focus on offline evaluation using historical data.
- The scope of the project does not include advanced anomaly detection techniques or unsupervised learning approaches.
- The model may require regular updates and retraining to remain effective against evolving fraud tactics, which is outside the scope of this initial project.

2. Data

2.1. Data source

The data was downloaded from a public dataset on Kaggle, posted by Anurag Verma. The dataset is a flat CSV file, with a size of 11.76 MB.

Link: *https://www.kaggle.com/datasets/anurag629/credit-card-fraud-transaction-data/data*

2.2. Data definition

The dataset has 16 features, including 1 target variable (Fraud field).

Field	Definition	Data Type	Sample
Transaction ID	A unique identifier for each transaction.	String	#3577 209
Date	The date on which the transaction was made.	String	14-Oct-20
Day of Week	The day of the week when the transaction occurred.	String	Monday
Time	The time (hour) at which the transaction occurred.	int64	12, 24
Type of Card	The type of card used in the transaction.	String	Visa, MasterCard
Entry Mode	The method used to input the card details for the transaction.	String	Tap, PIN
Amount	The monetary value of the transaction.	String	£5, £288

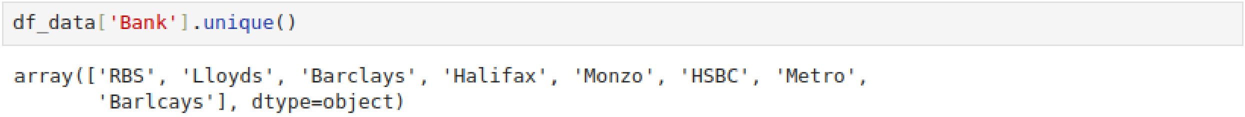
Type of Transaction	The nature of the transaction.	String	POS, ATM, Online
Merchant Group	The category or group to which the merchant belongs.	String	Entertainment, Fashion, Food, etc
Country of Transaction	The country where the transaction took place.	String	UK, USA, China, etc
Shipping Address	The address where goods or services are to be delivered, if applicable.	String	UK, USA, China, etc
Country of Residence	The country where the cardholder resides.	String	UK, USA, China, etc
Gender	The gender of the cardholder.	String	M, F
Age	The age of the cardholder.	float64	25.2, 49.6
Bank	The bank that issued the card.	String	Barclays, HSBC, RBS, etc
Fraud (Target)	Indicator of whether the transaction was fraudulent or not.	int64	0, 1

3. Exploratory Data Analysis (EDA)

3.1. Data information

Data has 100000 rows (data point) with 16 columns (field). Data type is shown in the previous table.

There are false values in the column Bank,

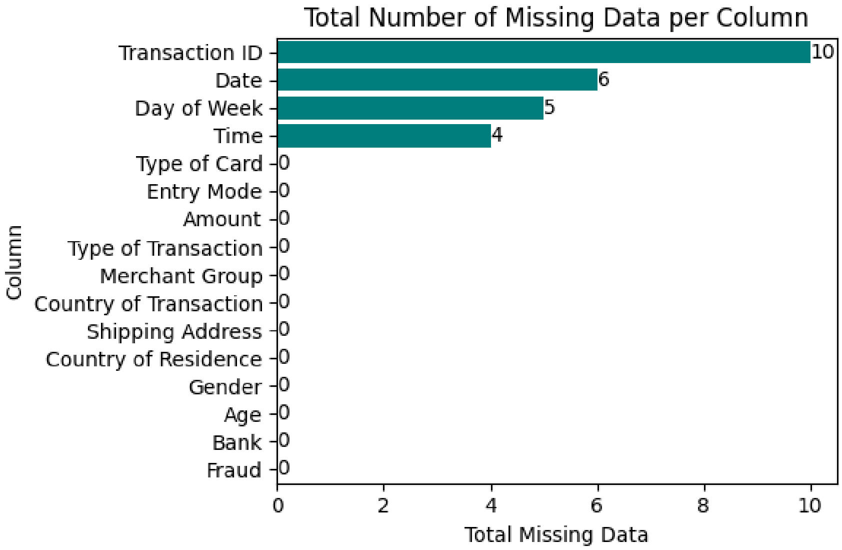


```
df_data['Bank'].unique()
```

```
array(['RBS', 'Lloyds', 'Barclays', 'Halifax', 'Monzo', 'HSBC', 'Metro',
       'Barlcays'], dtype=object)
```

‘Barlcays’ value should be ‘Barclays’. Therefore, the value will be changed.

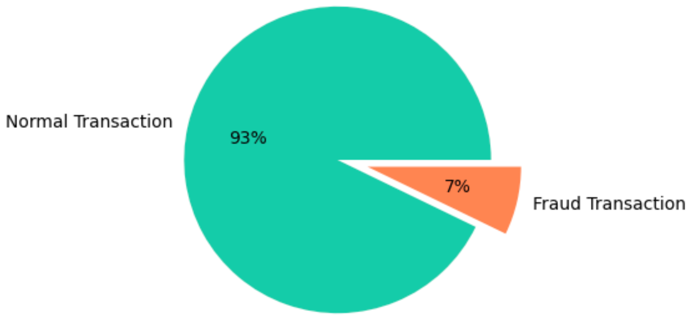
3.2. Missing data



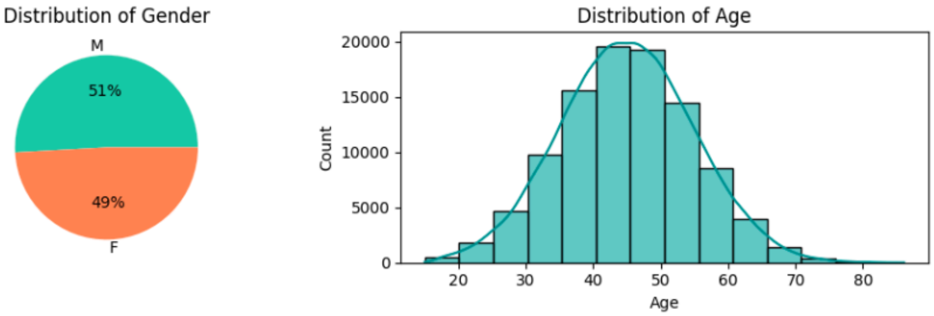
The amount of missing data is insignificant; therefore, these missing data points will be excluded from the analysis. After removing the missing data, the total number of rows is 99,977.

3.3. Fraud data

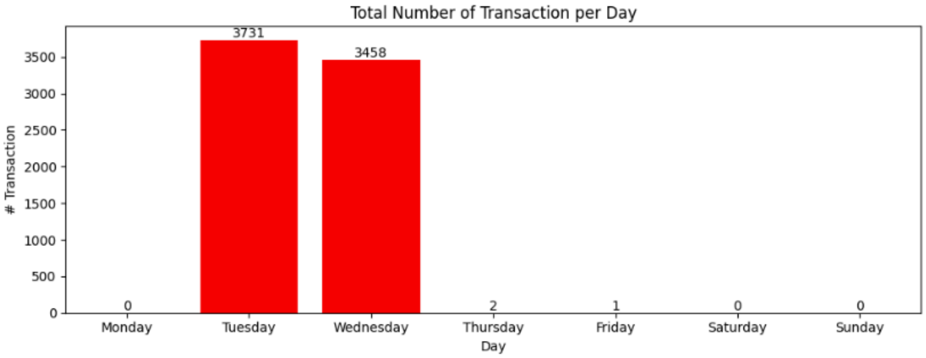
Out of the entire population, the total number of fraudulent transactions is 7,192, while the total number of normal transactions is 92,785.

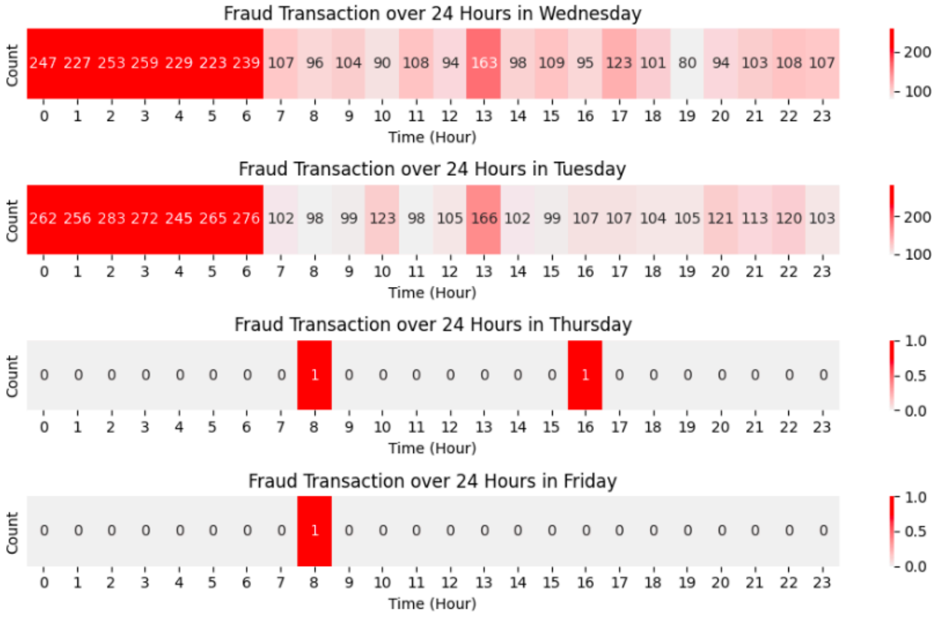


Based on gender, fraudulent transactions are almost equally balanced between females and males. As for age, the data is normally distributed.

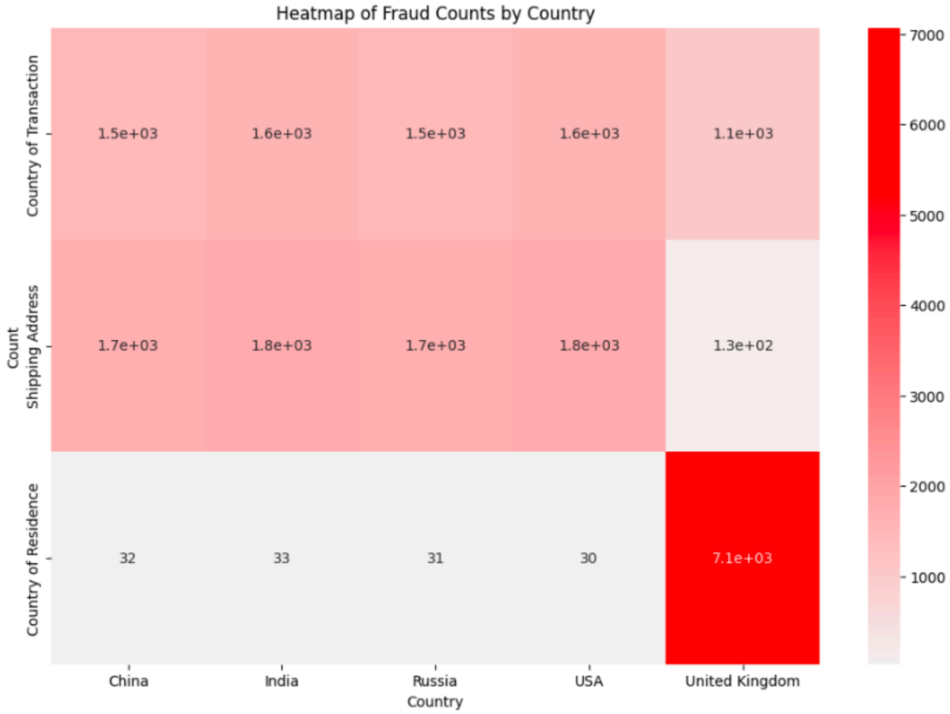


Based on the time dimension, fraudulent transactions are predominantly concentrated on Tuesday and Wednesday. Additionally, most of these transactions occur after midnight.

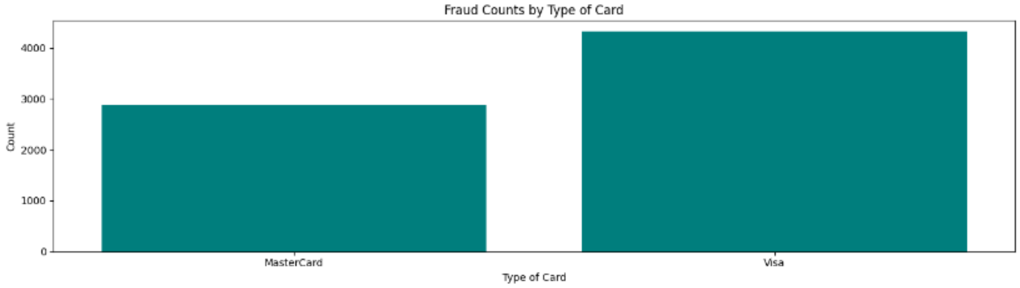




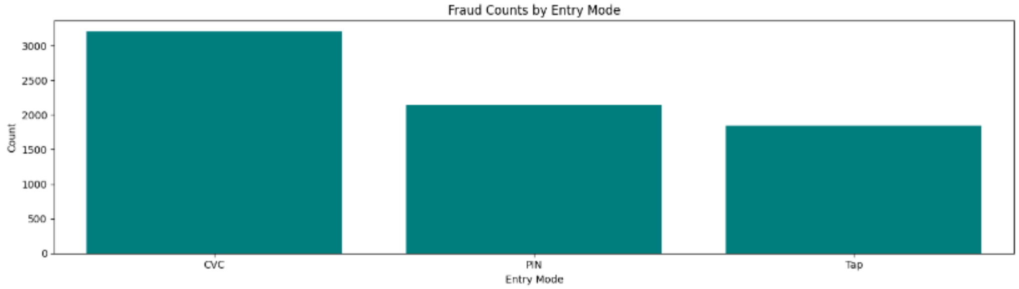
Based on location, most fraudulent transactions involved cards held by individuals residing in the UK. However, the transactions occurred evenly across China, India, Russia, the USA, and the UK. Additionally, the shipping addresses were mostly outside the UK and evenly distributed across other countries.



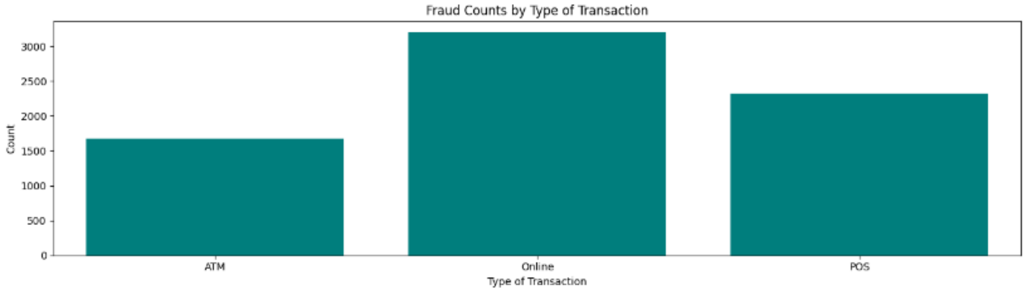
For others category,



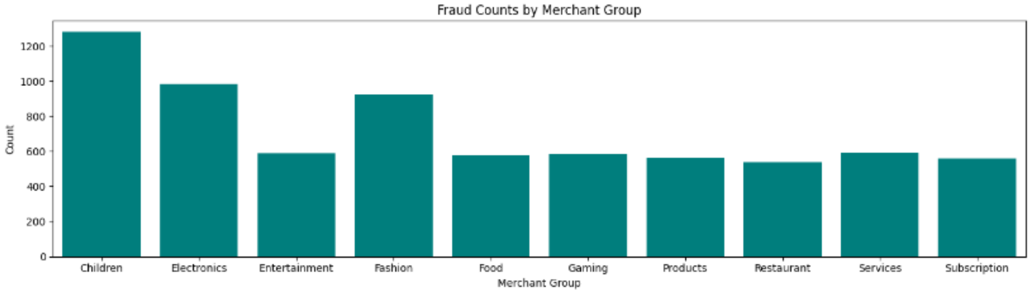
Fraud Counts by Type of Card



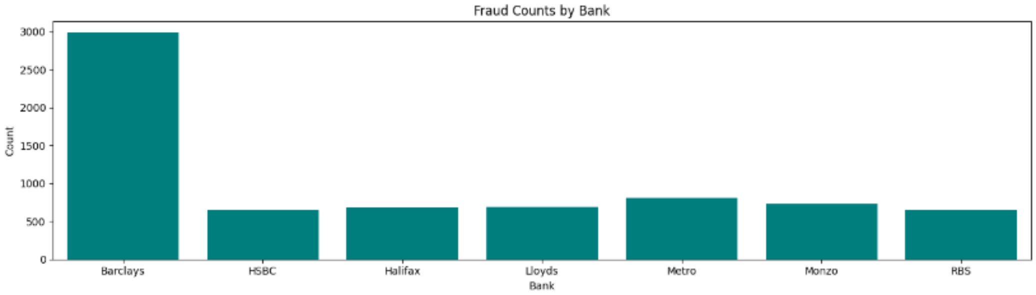
Fraud Counts by Entry Mode



Fraud Counts by Type of Transaction

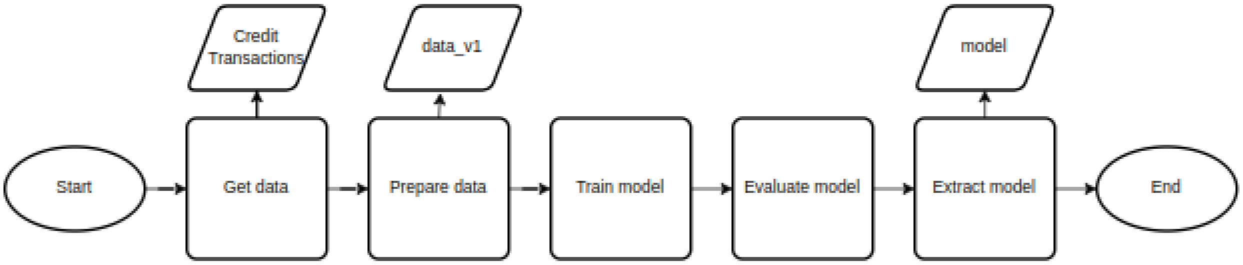


Fraud Counts by Merchant Group



Fraud Counts by Bank

4. Methodology



Data Preparation:

1. **Handling Missing Data**: Identify and address any missing values through imputation techniques or removal strategies to ensure data integrity and completeness.
2. **Correcting Data Types and Values**: Verify that each variable has the correct data type and rectify any inconsistencies or errors in data values to maintain accuracy and reliability.
3. **Addressing Data Imbalance**: Apply resampling techniques (undersampling) to correct class imbalances in the dataset, thereby improving model performance and generalization.
4. **Splitting Data**: Divide the dataset into training and test subsets, with 20% of the data allocated to the test set, ensuring a robust evaluation of the model's performance.
5. **Standardization**: Normalize numerical features using Standard Scaler to ensure uniformity across the dataset, which enhances model performance and convergence.
6. **Encoding Categorical Data**: Transform categorical variables into numerical formats through techniques such as one-hot encoding, facilitating their use in machine learning algorithms.

Model Training:

- Utilize various machine learning models to build predictive models. These are the model,
    - Decision Tree
    - K-Nearest Neighbors (KNN)
    - Random Forest
    - Support Vector Machine (SVM)
    - XGBoost
    - AdaBoost

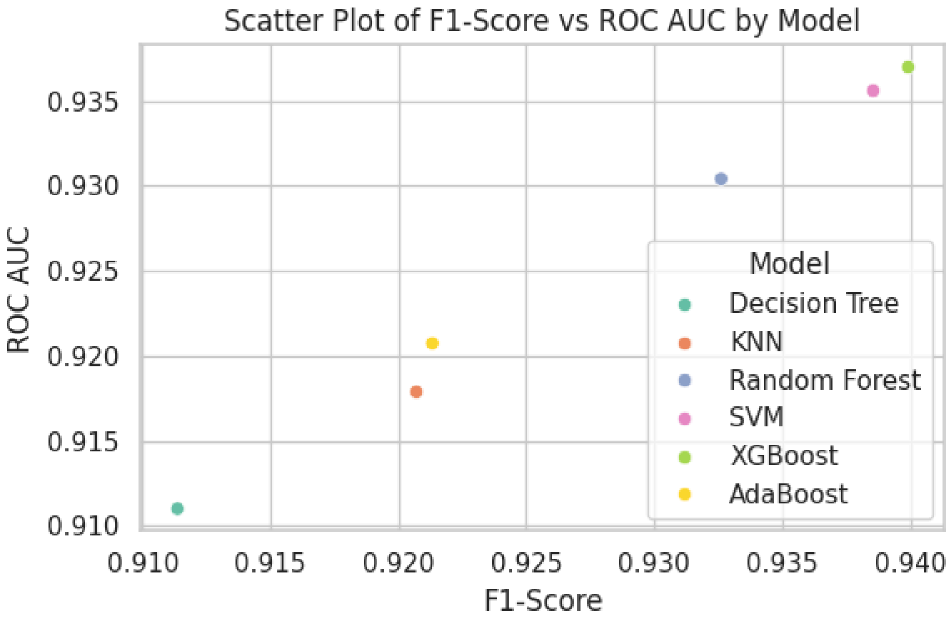
Model Evaluation:

- Assess model performance using appropriate metrics (e.g., accuracy, precision, recall, F1-score, AUC) and validate results through cross-validation or a holdout test set.

5. Result & Discussion

Model	Accuracy	Precision	Recall	F1-Score	ROC AUC
Decision Tree	0.911	0.910	0.913	0.911	0.911
KNN	0.918	0.893	0.950	0.921	0.918
Random Forest	0.930	0.907	0.960	0.933	0.930
SVM	0.936	0.901	0.980	0.939	0.936
XGBoost	**0.937**	0.901	**0.982**	**0.940**	**0.937**
AdaBoost	0.921	**0.917**	0.926	0.921	0.921

Based on the evaluation metrics, the **XGBoost** model emerges as the best performing model, excelling in both F1-Score and ROC AUC. Its superior ability to balance precision and recall, combined with the highest ROC AUC, makes it the optimal choice for this dataset.


Scatter Plot of F1-Score vs ROC AUC by Model
ROC AUC
F1-Score
Model
Decision Tree
KNN
Random Forest
SVM
XGBoost
AdaBoost