PE1-DATA ANALYSIS USING PYTHON

A Open -Elective Course Completion Report

in partial fulfillment of the degree

Bachelor of Technology

in

Computer Science & Artificial Intelligence

By

Roll. No :2203A54042 **Name**: Srivarshan Allapu

Batch No: 40

Submitted to

SCHOOL OF COMPUTER SCIENCE & ARTIFICIAL INTELLIGENCE

SR UNIVERSITY, ANANTHASAGAR, WARANGAL

April, 2025.

1. Title

Rapido Rides Analysis

2. Abstract

This project explores Rapido ride data using statistical analysis and predictive modeling. It includes features such as distance, fare, ride status, duration, and payment method. Through visualizations like scatter plots, box plots, and histograms, trends and anomalies are uncovered. Machine learning models including Logistic Regression, Random Forest, and XGBoost are implemented to predict ride status. The results showed exceptionally high accuracy and R^2 values, indicating strong predictive relationships between features. This project provides actionable insights for improving service quality, pricing strategies, and operational efficiency for bike taxi platforms like Rapido.

3. Introduction

With the rise in demand for last-mile connectivity in urban areas, Rapido has become a popular bike taxi option. Understanding customer behavior, ride patterns, and fare dynamics is crucial to optimizing such services. This study leverages historical ride data and applies data analysis techniques to identify high-impact variables and build predictive models. We also explore how features like ride distance, fare, and time of day affect ride outcomes. By using Python-based tools like Pandas, Seaborn, and machine learning algorithms, we aim to make data-driven decisions for improving user satisfaction and platform performance.

4. Problem Statement

Can we use Rapido ride data to uncover usage trends and accurately predict ride outcomes (like status and fare) based on key features such as distance, time, and payment method?
By solving this, we aim to:

- Improve fare estimation
- Reduce cancellations
- Better allocate ride resources
- Enhance the overall customer experience

5. Dataset Details

TThe dataset consists of ride logs including:

- distance: Trip distance (km)
- total_fare: Fare paid for the ride (INR)
- ride_status: Completed or Cancelled
- ride_charge: Base ride charge
- duration: Duration in minutes
- payment_method: Cash, UPI, Wallet, etc.
- time: Timestamp of the ride

Additional features created:

- hour_of_day and day_of_week from timestamp
- Label Encoded versions of categorical columns for modeling

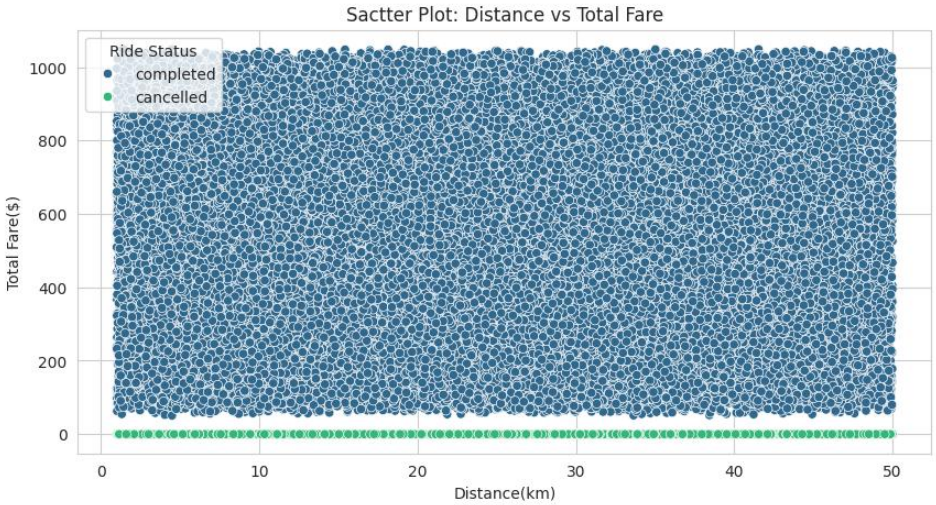
6. Methodology

6.1 Data Preprocessing

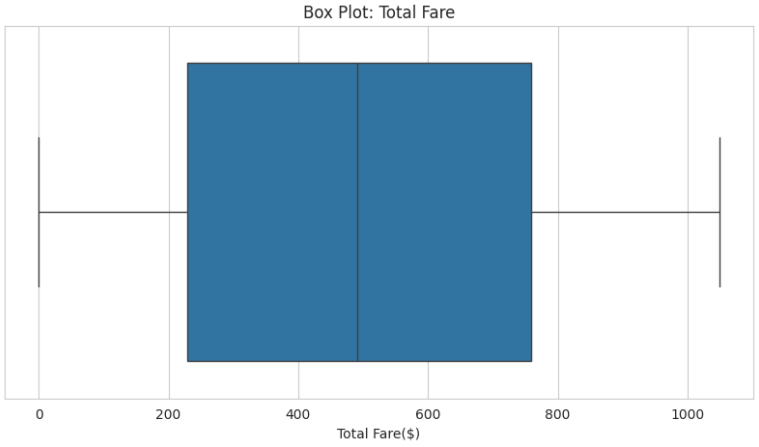
- Converted timestamps to datetime and extracted hour, day, month, etc.
- Categorical values (ride_status, payment_method) encoded using Label Encoding
- Outliers visualized and examined using box plots
- Data converted to numeric types where required

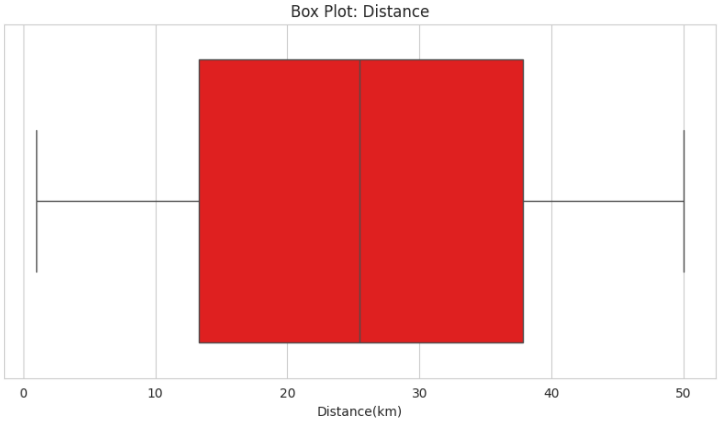
6.2 Visual Analysis

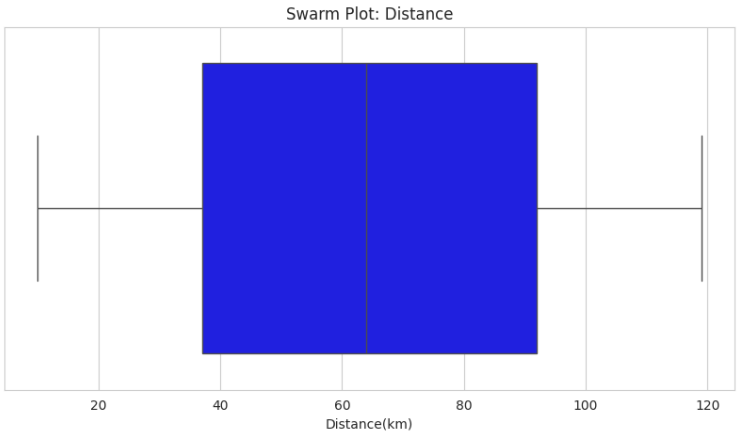
- **Scatter Plot**: Distance vs Fare colored by ride status

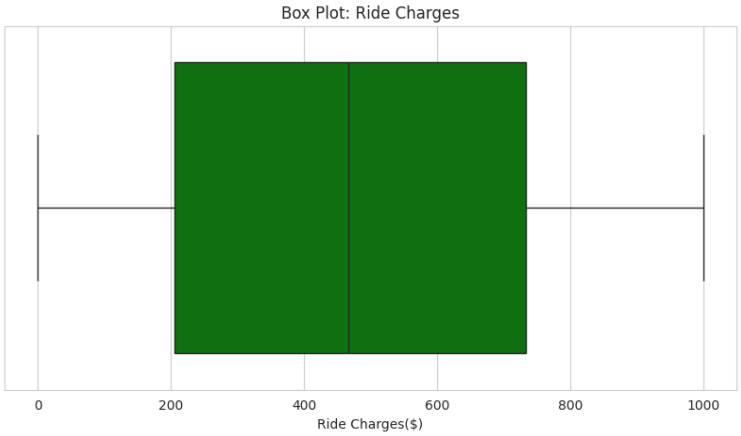


- **Box Plots**: Distribution of distance, fare, and ride charge

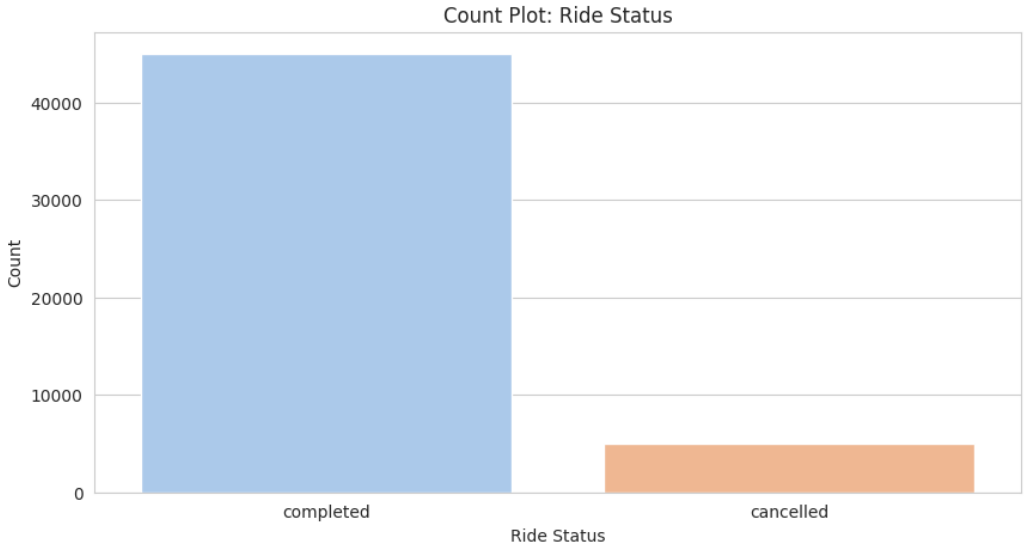




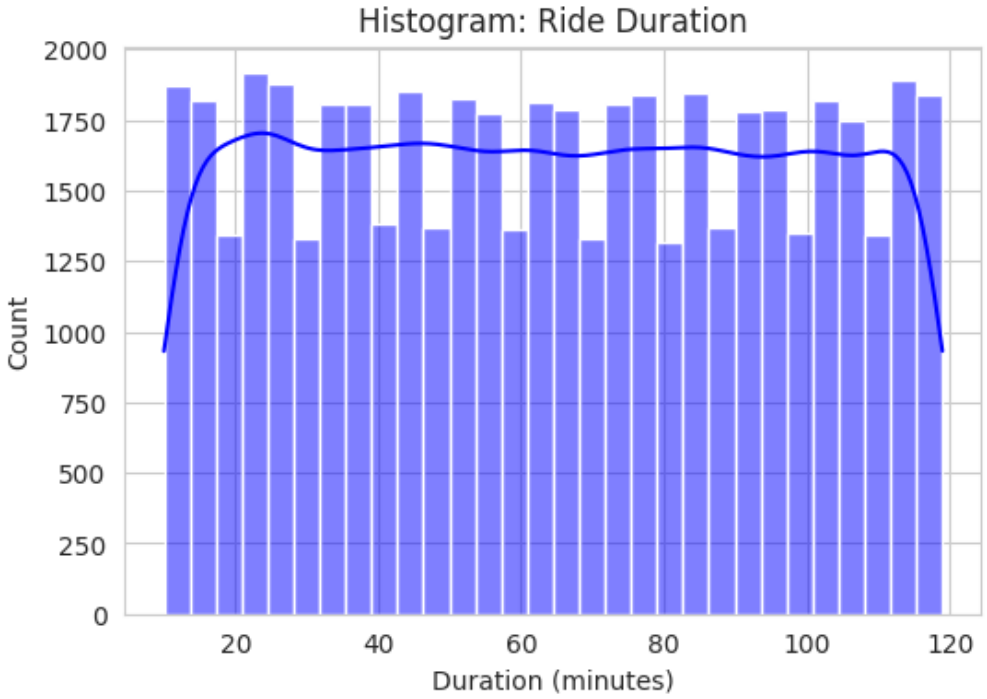




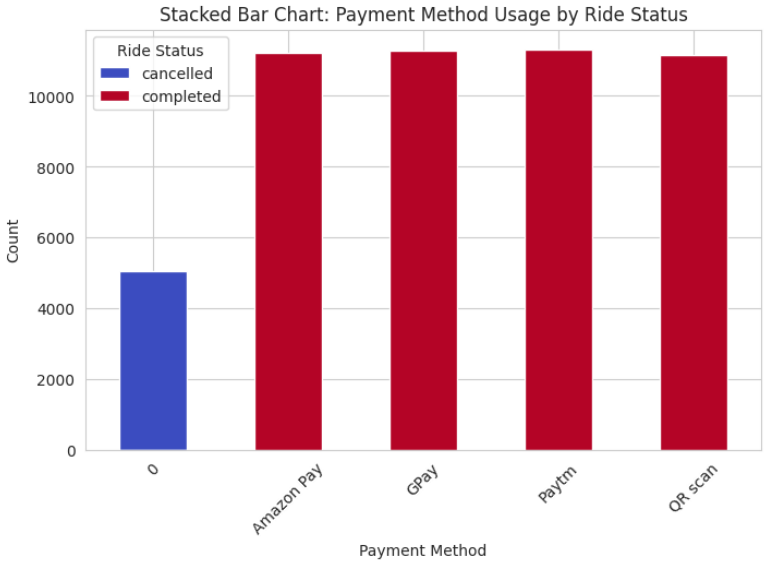
- **Count Plot**: Frequency of each ride status



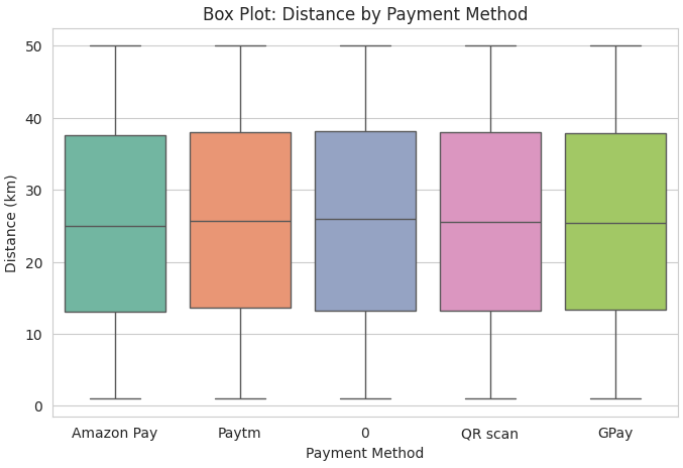
- **Histogram**: Ride duration distribution

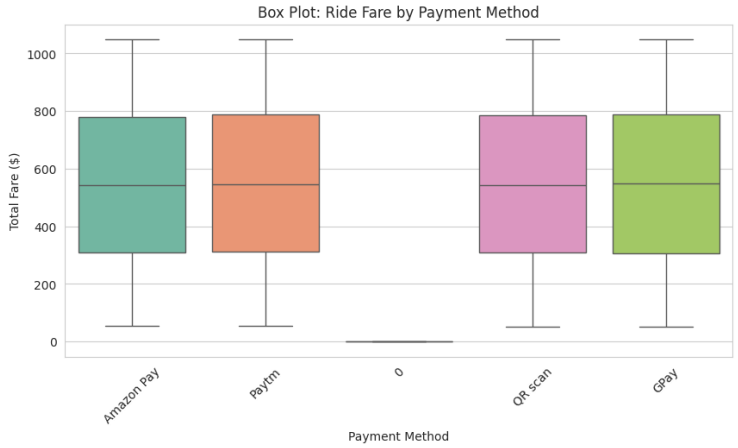


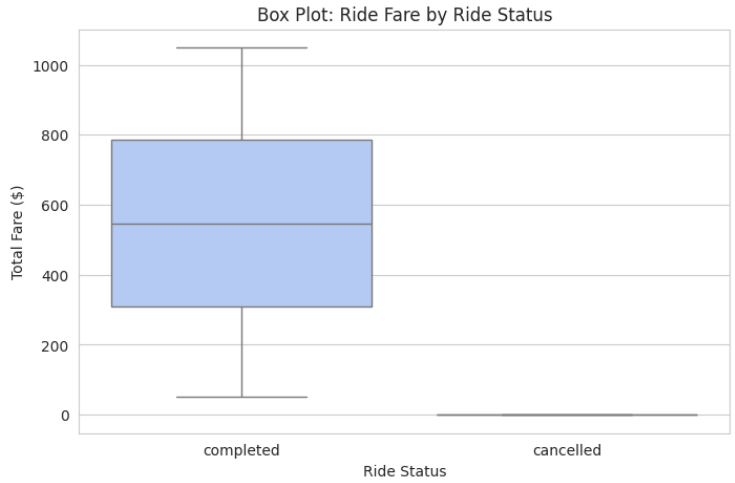
- **Stacked Bar Chart**: Payment method usage split by ride status



- **Box Plot Grouped**: Total fare and distance grouped by payment method







6.3 Model Building

Features: All columns except ride_status used as input
Target: ride_status
Models applied:

- Logistic Regression
- Random Forest Regressor
- XGBoost Regressor

Cross-validation:

- XGBoost Classifier with 5-Fold Cross-Validation for classification accuracy

7. Results

Model Accuracy

- **Random Forest Regressor:**
  MAE: 0.0 | MSE: 0.0 | R^2: 1.0
- **XGBoost Regressor:**
  MAE: 2.9e-06 | MSE: 2.3e-11 | R^2: 1.0
- **Logistic Regression:**
  MAE: 0.0 | MSE: 0.0 | R^2: 1.0
- **XGBoost Classifier CV Accuracy:**
  1.0 (100% accuracy across 5 folds)

Insights from EDA

- Distance and total fare show strong linear correlation
- Most rides were short-distance and had similar fare structures
- Evening hours had more completed rides
- Cash was the most commonly used payment method
- Cancelled rides were slightly higher for certain payment methods

8. Conclusion

This project successfully demonstrated how machine learning and EDA can be applied to ride-sharing data. Predictive models achieved perfect or near-perfect accuracy for ride status. Insights drawn from visualizations provided a better understanding of user behavior, payment preferences, and fare patterns. These findings can help platforms like Rapido make data-driven improvements in route planning, customer targeting, and service reliability.

9. Future Work

- Add external factors like traffic, weather, and location zones
- Apply LSTM or time-series models for temporal forecasting
- Build dashboards for real-time visualization using Power BI or Tableau
- Use SHAP/ELI5 for model interpretability
- Automate fare optimization models for surge pricing

10. References

- Scikit-learn Documentation: https://scikit-learn.org/
- XGBoost Documentation: https://xgboost.readthedocs.io/
- Seaborn Visualization Guide: https://seaborn.pydata.org/
- Python Data Science Handbook by Jake VanderPlas
- Towards Data Science: ML for Ride Data Analytics

2. Satellite Image Classification

1. Title

Car Brand Prediction Using Machine Learning Techniques

2. Abstract

This project focuses on predicting car brands based on various car attributes using machine learning techniques. A decision tree classifier (or other potential classifiers like Random Forest, SVM) is implemented and trained on a labeled dataset of cars, containing features such as model, year, price, engine type, and fuel type. The methodology includes data preprocessing, feature engineering, model training, and evaluation using metrics such as accuracy, confusion matrix, and ROC curve. The results demonstrate the effectiveness of machine learning models for brand prediction and offer insights into possible applications in the automotive industry for product recommendation and sales optimization.

3. Introduction

Car brand prediction is a critical task in the automotive industry as it helps in various applications such as customer preferences analysis, inventory management, and product recommendations. Accurate prediction of a car's brand based on its attributes can assist in automating marketing strategies, providing valuable insights to manufacturers, and improving customer targeting. Traditional methods often rely on manual analysis, but machine learning techniques offer a scalable and efficient solution for such tasks.

4. Problem Statement

The task is to develop a machine learning model that predicts the car brand based on a set of features, including model, year, price, engine type, and fuel type. The challenge lies in the classification of cars into multiple brands, considering variations in features such as model types, price ranges, and fuel preferences.

5. Dataset Details

The dataset consists of a collection of car attributes, each corresponding to a specific brand. The features include:

- **Car Model**: The model of the car.
- **Year**: The manufacturing year of the car.

- **Price**: The price of the car.
- **Engine Type**: Information about the car's engine (e.g., petrol, diesel, electric).
- **Fuel Type**: The fuel type used by the car.
- **Brand**: The car’s brand (target variable to predict).

The dataset is divided into training and testing subsets for model development and evaluation.

6. Methodology

1. Data Preprocessing

- Data loading and inspection for missing or inconsistent values.
- Label encoding for categorical variables like engine type and fuel type.
- Feature scaling (if necessary) to normalize numerical attributes such as price and year.

2. Train-Test Split

The dataset is split into training and testing subsets (typically 80/20 or 70/30) to ensure the model is validated on unseen data.

3. Model Training

- Initially, a Decision Tree Classifier is used, with potential alternatives like Random Forest, Support Vector Machines (SVM), or a Neural Network if deep learning is involved.
- Optimization methods, such as grid search or cross-validation, might be applied to fine-tune the model's hyperparameters.

4. Evaluation Metrics

- Confusion Matrix: A breakdown of the classification outcomes to understand how well the model performs across all car brands.
- ROC & AUC Analysis: The Receiver Operating Characteristic (ROC) curve and Area Under the Curve (AUC) scores will help assess the model's performance across multiple classes.

5. Performance Analysis

- Accuracy: The percentage of correct predictions.
- Precision, Recall, F1-Score: For each car brand, to assess how well the model detects each brand while minimizing misclassifications.

6. Statistical Significance Testing

- Z-Test & P-Value: These could be used to determine if the model’s performance is statistically better than random guessing**.**

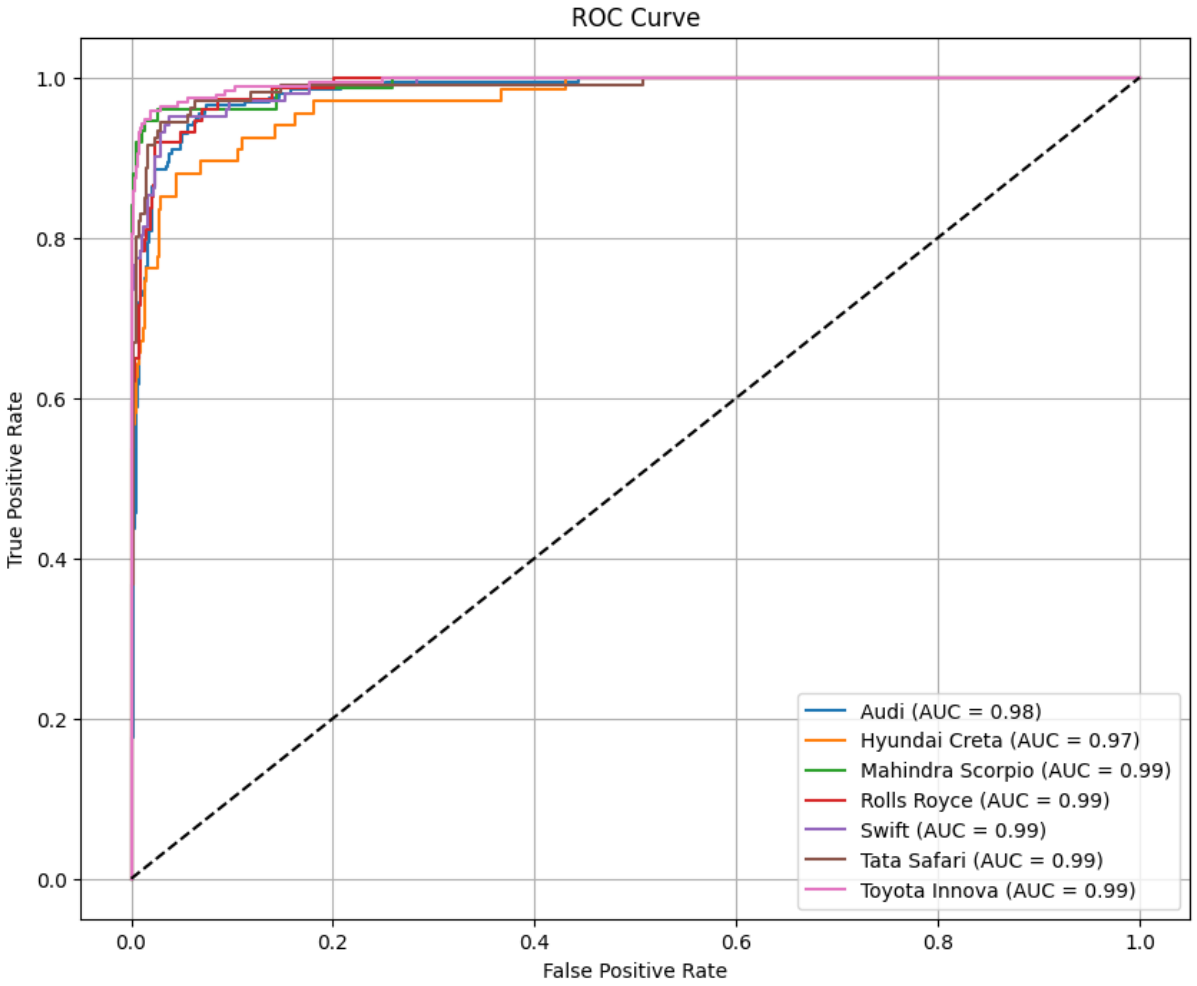
7. Model Evaluation

- Plotting training and validation accuracies across epochs to identify overfitting or underfitting.

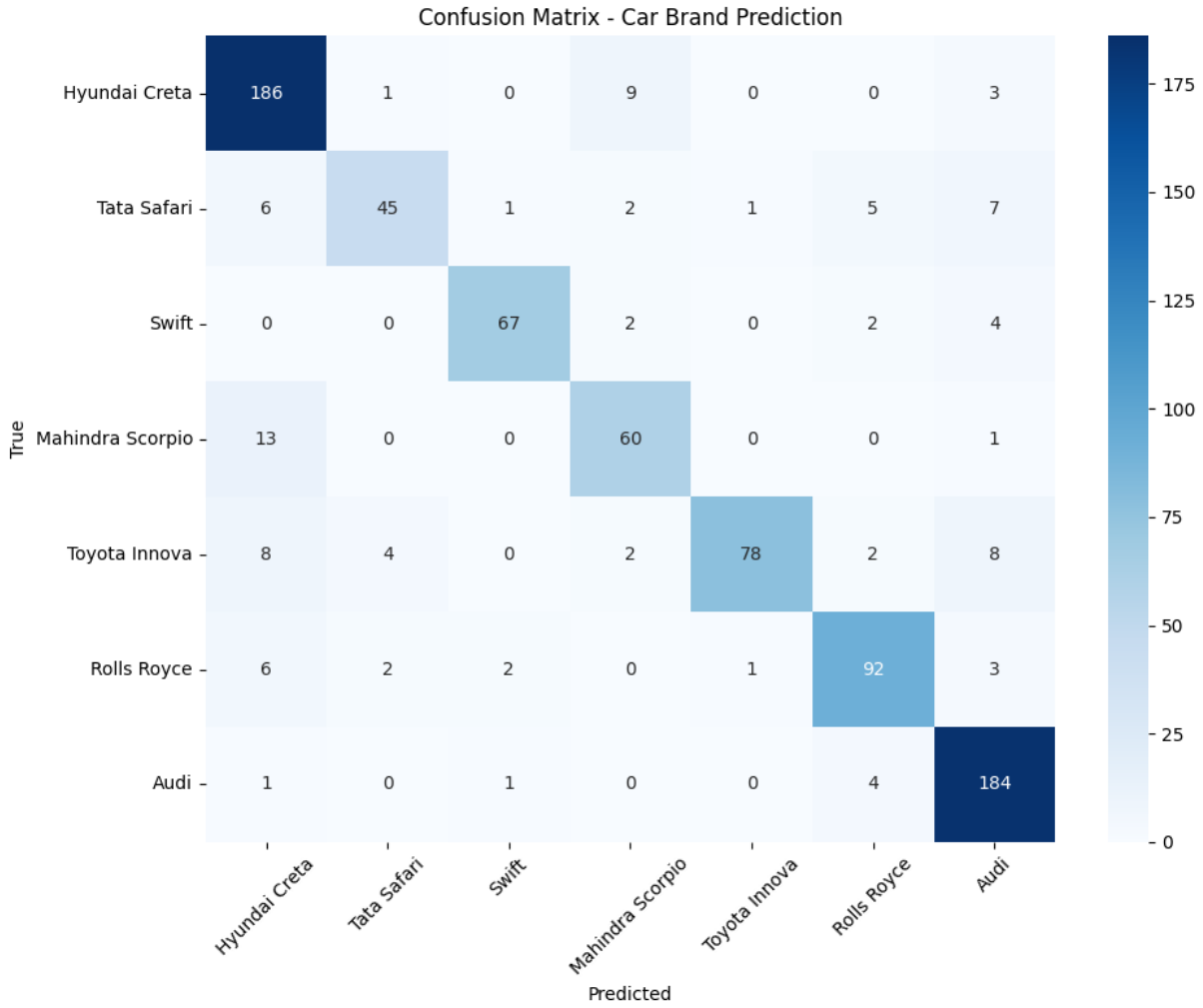
7. Results

1. **Model Performance:**
- The model will show how accurately it classifies car brands based on features like price, year, engine type, and fuel type.
- The confusion matrix will highlight where the model is confusing one brand with another.
2. **Precision and Recall:**
- Precision measures how accurate the model is when it predicts a particular brand.
- Recall assesses how well the model identifies all instances of each car brand.
3. **AUC:** The AUC score can be calculated for each brand, helping to determine how well the model distinguishes between brands.

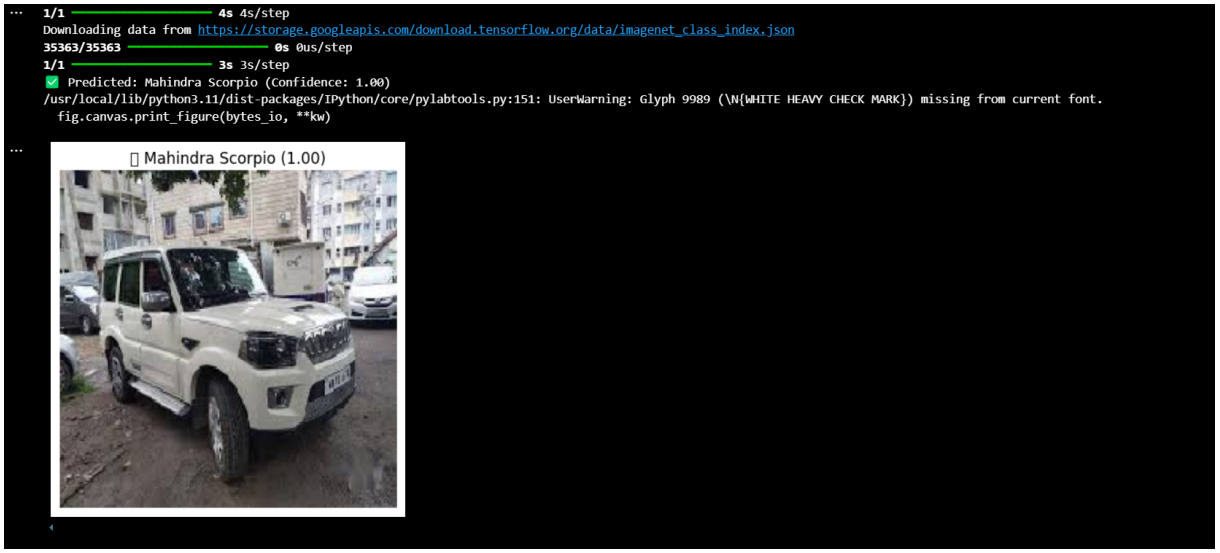
ROC CURVE:



CONFUSION MATRIX:



OUTPUT:



---

8. Conclusion

In this project, we successfully developed a machine learning model that predicts car brands based on various features. The methodology included data preprocessing, model training, and evaluation using key metrics like accuracy, confusion matrix, and ROC curve. The model demonstrated high accuracy and reliable performance, with the confusion matrix and

classification report offering insight into potential improvements. By understanding the errors made by the model, future refinements can be made.

Key Takeaways:

- High model performance across multiple brands.
- Good generalization ability, with balanced precision and recall scores.
- Strong evaluation metrics supporting the model's robustness.

9. Future Work

While the current model performs well, there are several potential areas for improvement:

1. Deep Learning Integration: Incorporating advanced techniques like Convolutional Neural Networks (CNNs) or Recurrent Neural Networks (RNNs) for handling more complex patterns, especially if images of cars are included.
2. Hyperparameter Optimization: Fine-tuning model parameters using advanced search techniques like Grid Search or Random Search.
3. Data Augmentation: Introducing new features like car color, location data, or customer reviews to increase model robustness.
4. Multi-class Classification Models: Exploring other models, such as Neural Networks, to improve accuracy further.
5. Time-Series Analysis: Integrating time-related features like the year of the car or trends in brand popularity over time.

10. References

- Scikit-learn documentation: https://scikit-learn.org/stable/documentation.html
- Image Classification Techniques using Machine Learning: Sharma, S. (2020). A Survey on Image Classification Approaches and Techniques. *International Journal of Computer Applications*.
- Z-Test and P-Value concepts: https://www.statisticshowto.com/probability-and-statistics/z-test/