Artificial Intelligence and Machine Learning

Project Report

Semester-IV (Batch-2022)

NYC Taxi Data Analysis

Submitted To: - Dr Monica Dutta

Supervised By: - Shubham Singhal

Hanish Mittal	2210991604	G11
Himanshu Malhotra	2210991663	G11
Harsh	2210991612	G11
Harsh	2210991614	G11

Department of Computer Science and Engineering
Chitkara University Institute of Engineering & Technology,
Chitkara University, Punjab

Table of Contents:

1 Introduction:-

1.1 Background:

Analyzing NYC taxi data holds significant relevance in several domains, particularly in transportation, urban planning, and data science. Here's a brief overview of the context and significance:

Transportation Optimization: NYC taxis serve as a vital mode of transportation in one of the busiest cities globally, generating vast amounts of data regarding travel patterns, routes, and demand fluctuations. Analyzing this data can help optimize taxi operations, reduce congestion, and improve overall transportation efficiency.

Urban Planning: Understanding taxi usage patterns provides valuable insights into urban mobility trends, helping city planners make informed decisions regarding infrastructure development, traffic management, and public transportation systems.

Predictive Analytics: By leveraging AI and ML techniques on NYC taxi data, predictive models can be developed to forecast future demand, identify high-demand areas, and optimize taxi dispatching strategies. This enhances service reliability and customer satisfaction while minimizing waiting times and empty trips for drivers.

Data Science Applications: NYC taxi data presents an excellent opportunity for data scientists to apply various machine learning algorithms and statistical techniques for tasks such as clustering analysis, anomaly detection, and predictive modeling. These analyses can uncover hidden patterns, derive actionable insights, and contribute to advancements in AI-driven decision-making.

In summary, analyzing NYC taxi data offers a rich source of information that can be leveraged to address real-world challenges in transportation, urban planning, and data science. By applying AI and ML techniques to this data, we can enhance transportation efficiency, optimize urban infrastructure, and pave the way for smarter, more sustainable cities.

1.2Objectives:

The primary objective of this project is to develop an AI-driven predictive model for car price estimation. Specifically, our goals include:

Demand Prediction: Develop models to accurately predict taxi demand in different regions of NYC at various times of the day, week, and year.

Route Optimization: Identify optimal taxi routes based on historical traffic patterns, time of day, and customer demand, aiming to minimize travel time and maximize driver earnings.

Customer Segmentation: Segment taxi customers based on their travel preferences, frequency, and destinations, enabling personalized service offerings and targeted marketing strategies.

Anomaly Detection: Detect unusual or anomalous taxi behavior, such as sudden spikes or drops in demand, irregular route deviations, or fraudulent activities, to enhance operational efficiency and security.

Fare Estimation: Develop algorithms to estimate taxi fares accurately, considering factors like distance, time, traffic conditions, and surge pricing, to provide transparency and fairness to customers.

Environmental Impact Assessment: Analyze the environmental impact of taxi operations, including carbon emissions, fuel consumption, and congestion, to inform sustainability initiatives and policy decisions.

Accessibility Improvement: Evaluate the accessibility of taxi services for different demographic groups and geographic areas, identifying underserved communities and proposing solutions for equitable access.

Operational Efficiency: Optimize taxi fleet management, driver scheduling, and dispatching algorithms to improve overall operational efficiency, reduce idle time, and increase revenue generation.

Predictive Maintenance: Develop models to predict maintenance needs for taxi vehicles based on usage patterns, mileage, and wear and tear, aiming to minimize downtime and repair costs.

Market Insights: Extract insights from taxi data regarding market trends, competitor analysis, and customer preferences to support strategic business decisions and enhance competitiveness in the transportation industry.

These objectives can serve as guiding principles for your analysis, helping you focus your efforts and derive meaningful insights from the NYC taxi data.

1.3Significance:

The significance of this project are:-

Transportation Efficiency: By analyzing NYC taxi data, we can optimize transportation efficiency by identifying high-demand areas, peak travel times, and optimal routes. This not only improves the overall taxi service but also contributes to reducing traffic congestion and enhancing urban mobility.

Customer Experience: Understanding customer travel patterns and preferences enables personalized service offerings, improved route planning, and more accurate fare estimates. This enhances the overall customer experience and satisfaction, leading to increased loyalty and retention.

Resource Allocation: Analyzing taxi data helps in better allocating resources such as vehicles and drivers, leading to reduced idle time, improved utilization rates, and increased revenue generation for taxi companies.

Environmental Impact: By optimizing routes and reducing unnecessary trips, we can minimize fuel consumption, carbon emissions, and environmental pollution associated with taxi operations. This contributes to sustainability efforts and mitigates the environmental impact of urban transportation.

Public Policy and Planning: Insights derived from taxi data analysis inform public policy decisions and urban planning initiatives. For example, identifying underserved areas can guide the allocation of transportation resources, while predicting demand patterns helps in designing efficient public transportation systems.

Safety and Security: Analyzing taxi data enables the detection of irregularities and anomalies, such as route deviations or suspicious behavior, contributing to improved safety and security for both passengers and drivers.

Business Insights: For taxi companies and stakeholders, analyzing NYC taxi data provides valuable business insights, including market trends, competitor analysis, and customer behavior. This helps in strategic decision-making, marketing campaigns, and revenue optimization.

Overall, this project on analyzing NYC taxi data is important as it addresses critical challenges in urban transportation, enhances customer experience, contributes to sustainability goals, informs public policy decisions, and provides valuable business insights. It has the potential to bring about tangible benefits for both transportation stakeholders and the broader community.

2 Problem Definition and Requirements:-

2.1 Problem Statement:

The aim of this project is to leverage machine learning and data analysis techniques to address the challenges of optimizing taxi operations in New York City. Specifically, we seek to develop predictive models to accurately forecast taxi demand across different regions and time periods, enabling more efficient allocation of resources and route planning. Additionally, we aim to identify opportunities for improving overall service quality, reducing congestion, and enhancing the customer experience through data-driven insights and recommendations. By tackling these challenges, we aim to contribute to the advancement of transportation efficiency and urban mobility in NYC.

2.2 Software Requirements:

Pandas: Data manipulation and analysis.

NumPy: Numerical computing and matrix operations.

Matplotlib: Basic plotting and visualization.

Scikit-learn: Machine learning algorithms and tools.

2.3 Hardware Requirements:

The hardware requirements for this project are minimal and include a standard computer with sufficient processing power and memory to run the chosen algorithms and process large datasets.

2.4 Data Sets:

The project will utilize publicly available NYC Taxi datasets, such as the Kaggle Used NYC Data Analysis by 15 models dataset, comprising a mix of new and old models for training and testing the ML models.

3. Proposed Design / Methodology:-

3.1 Schematic Diagram:

The schematic diagram illustrates the flow of the project, starting with data collection from various sources such as historical sales records, market trends, and economic indicators. Preprocessing techniques are applied to clean and normalize the data. The processed data is then fed into the AI-driven predictive model, which utilizes advanced machine learning algorithms to generate accurate car price predictions. Finally, the model's outputs are evaluated and validated against real-world data, ensuring its effectiveness and reliability.

3.2 File Structure:

The project adopts a modular and organized file structure to facilitate code development, experimentation, and collaboration. The file structure is structured as follows:

- Data: This directory contains the raw car price datasets obtained from reputable sources, such as Kaggle or other institutions. It also includes any additional datasets or resources used for feature engineering or model training.

- Notebooks: This directory contains Jupyter Notebooks, serving as the primary development environment for exploratory data analysis (EDA), data preprocessing, model training, and evaluation. Each notebook is dedicated to a specific task or stage of the project, ensuring modularity and reproducibility.

- Scripts: This directory contains Python scripts encapsulating reusable functions, classes, and utilities utilized throughout the project. These scripts facilitate code organization, modularity, and maintainability, enabling seamless integration into the overall system architecture.

- Models: This directory stores serialized machine learning models trained on the car price data. Each model is saved in a standardized format (e.g., pickle, HDF5) for easy retrieval and deployment in production environments.

- Visualizations: This directory contains visualizations generated during exploratory data analysis, model evaluation, and performance monitoring. These visualizations include histograms, scatter plots, confusion matrices, ROC curves, and precision-recall curves, providing insights into the data distribution, model performance, and decision boundaries.

- Configuration: This directory contains configuration files specifying hyperparameters, settings, and environmental variables required for model training, evaluation, and deployment. These configuration files enable reproducible experiments and facilitate parameter tuning and optimization.

- Tests: This directory comprises unit tests, integration tests, and end-to-end tests validating the correctness, robustness, and performance of the implemented functionalities. Test suites are executed automatically using continuous integration (CI) tools like Travis CI or Jenkins, ensuring code quality and reliability throughout the development lifecycle.

- Results: This directory stores intermediate and final results generated during data preprocessing, model training, and evaluation. It includes metrics, logs, and output files summarizing the performance of trained models, facilitating comprehensive analysis and comparison.

- Dependencies: This directory contains dependency files specifying project dependencies, libraries, and versions required for reproducible environment setup and execution. Dependency management tools like pipenv or conda environment files are used to manage dependencies efficiently and avoid conflicts.

By adopting this structured file organization, the project ensures clarity, maintainability, and scalability, enabling seamless development, experimentation, and deployment of the NYC Taxi Data .

3.3 Algorithms Used:

Linear Regression:

Purpose: Predicting continuous variables like fare amount or trip duration.

How It Works: Establishes a linear relationship between the dependent variable and one or more independent variables.

Decision Trees:

Purpose: Classification and regression tasks.

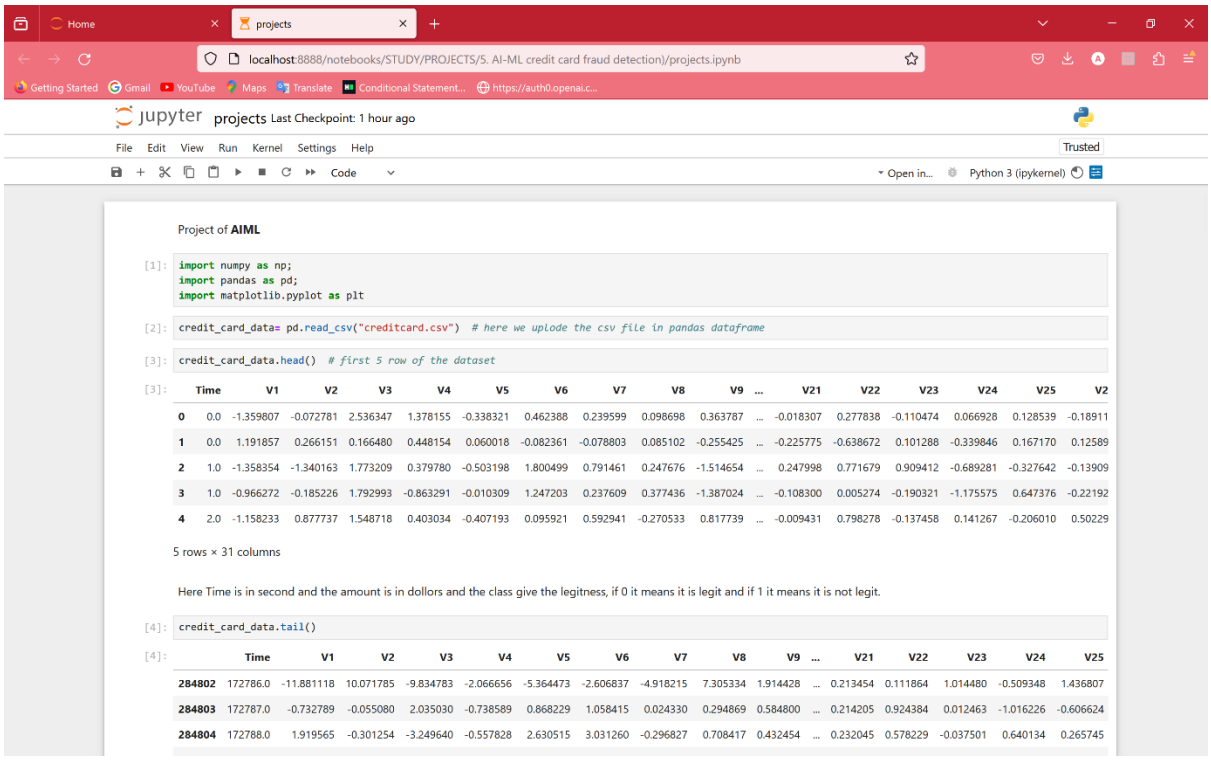
How It Works: Splits the data into subsets based on the value of input features, creating a tree-like model of decisions.

Random Forest:

Purpose: Improve prediction accuracy and control overfitting.

How It Works: Combines multiple decision trees (bagging) to improve generalization and accuracy.

4 Result: -



Project of **AIML**

```
[1]: import numpy as np;
     import pandas as pd;
     import matplotlib.pyplot as plt
```

```
[2]: credit_card_data= pd.read_csv("creditcard.csv")  # here we uplode the csv file in pandas dataframe
```

```
[3]: credit_card_data.head()  # first 5 row of the dataset
```

[3]:

	Time	V1	V2	V3	V4	V5	V6	V7	V8	V9	...	V21	V22	V23	V24	V25	V2
0	0.0	-1.359807	-0.072781	2.536347	1.378155	-0.338321	0.462388	0.239599	0.098698	0.363787	...	-0.018307	0.277838	-0.110474	0.066928	0.128539	-0.18911
1	0.0	1.191857	0.266151	0.166480	0.448154	0.060018	-0.082361	-0.078803	0.085102	-0.255425	...	-0.225775	-0.638672	0.101288	-0.339846	0.167170	0.12589
2	1.0	-1.358354	-1.340163	1.773209	0.379780	-0.503198	1.800499	0.791461	0.247676	-1.514654	...	0.247998	0.771679	0.909412	-0.689281	-0.327642	-0.13909
3	1.0	-0.966272	-0.185226	1.792993	-0.863291	-0.010309	1.247203	0.237609	0.377436	-1.387024	...	-0.108300	0.005274	-0.190321	-1.175575	0.647376	-0.22192
4	2.0	-1.158233	0.877737	1.548718	0.403034	-0.407193	0.095921	0.592941	-0.270533	0.817739	...	-0.009431	0.798278	-0.137458	0.141267	-0.206010	0.50229

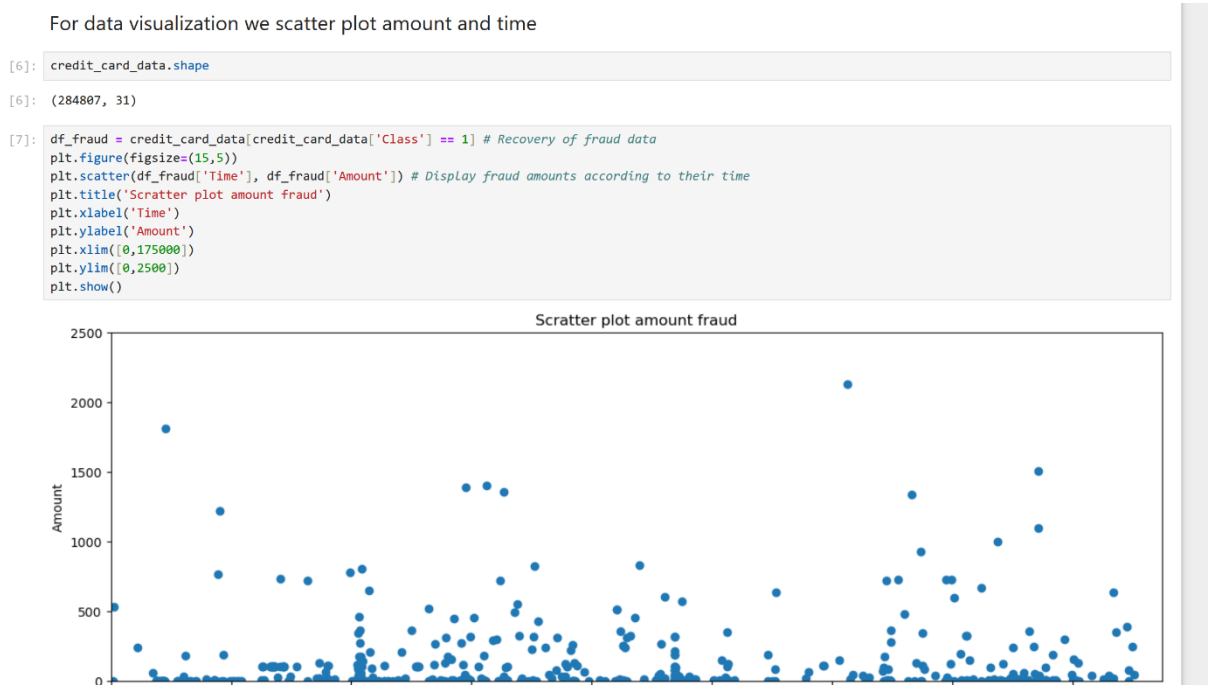
5 rows × 31 columns

Here Time is in second and the amount is in dollors and the class give the legitness, if 0 it means it is legit and if 1 it means it is not legit.

```
[4]: credit_card_data.tail()
```

[4]:

	Time	V1	V2	V3	V4	V5	V6	V7	V8	V9	...	V21	V22	V23	V24	V25
284802	172786.0	-11.881118	10.071785	-9.834783	-2.066656	-5.364473	-2.606837	-4.918215	7.305334	1.914428	...	0.213454	0.111864	1.014480	-0.509348	1.436807
284803	172787.0	-0.732789	-0.055080	2.035030	-0.738589	0.868229	1.058415	0.024330	0.294869	0.584800	...	0.214205	0.924384	0.012463	-1.016226	-0.606624
284804	172788.0	1.919565	-0.301254	-3.249640	-0.557828	2.630515	3.031260	-0.296827	0.708417	0.432454	...	0.232045	0.578229	-0.037501	0.640134	0.265745



For data visualization we scatter plot amount and time

```
[6]: credit_card_data.shape
```

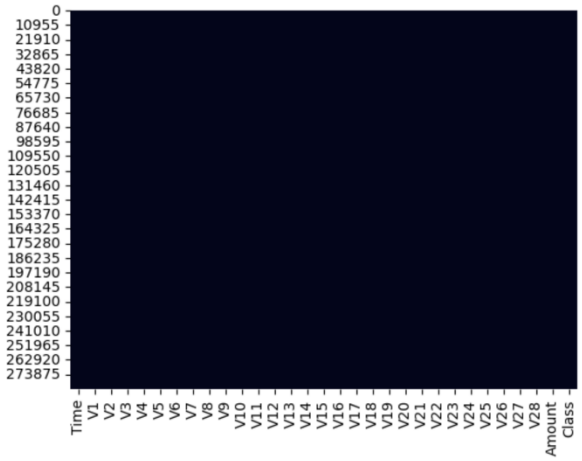
```
[6]: (284807, 31)
```

```
[7]: df_fraud = credit_card_data[credit_card_data['Class'] == 1] # Recovery of fraud data
     plt.figure(figsize=(15,5))
     plt.scatter(df_fraud['Time'], df_fraud['Amount']) # Display fraud amounts according to their time
     plt.title('Scratter plot amount fraud')
     plt.xlabel('Time')
     plt.ylabel('Amount')
     plt.xlim([0,175000])
     plt.ylim([0,2500])
     plt.show()
```

[8]:
```
import seaborn as sns
```

[9]:
```
#check the missing values in each columns
sns.heatmap(credit_card_data.isnull(), cbar=False)       # As we see no empty data is here
```

[9]: <Axes: >



jupyter projects Last Checkpoint: 1 hour ago

File Edit View Run Kernel Settings Help

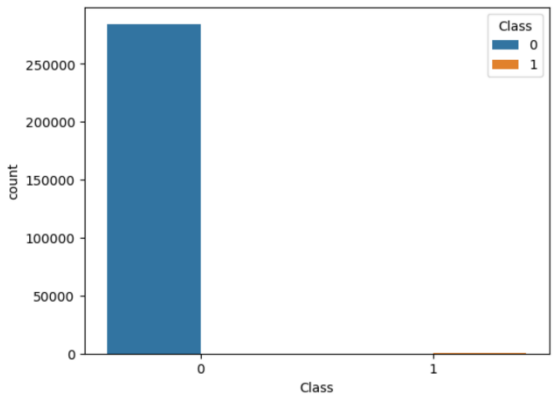
Trusted

Code

Open in... Python 3 (ipykernel)

[11]:
```
sns.countplot(x='Class', hue="Class", data=credit_card_data)     #
```

[11]: <Axes: xlabel='Class', ylabel='count'>



[12]:
```
credit_card_data['Class'].value_counts() # here we get the legit and non-legit data's
```

[12]:
```
Class
0    284315
1       492
Name: count, dtype: int64
```

0 -> legit transaction

Take Another model for prediction method

[39]:
```
from sklearn import svm
```

[40]:
```
svc_model= svm.SVC(kernel= 'linear')
```

[41]:
```
svc_model.fit(x_train, y_train)
```

[41]:
```
SVC
SVC(kernel='linear')
```

[42]:
```
predict_x_trains = svc_model.predict(x_train)
```

[43]:
```
from sklearn.metrics import confusion_matrix
```

[44]:
```
cm=confusion_matrix(y_train, predict_x_trains)
print(cm)
```

```
[[386   7]
 [ 74 320]]
```

[45]:
```
print("Accuracy of train data : " + str((cm[0][0]+cm[1][1])/ (cm[0][0]+cm[0][1]+cm[1][0]+cm[1][1])))
```

```
Accuracy of train data : 0.8970775095298602
```

[46]:
```
predict_x_test = svc_model.predict(x_test)
```

[47]:
```
cm1=confusion_matrix(y_test, predict_x_test)
print(cm1)
```

```
[[94  5]
 [20 78]]
```

[48]:
```
print("Accuracy of train data : " + str((cm1[0][0]+cm1[1][1])/ (cm1[0][0]+cm1[0][1]+cm1[1][0]+cm1[1][1])))
```

```
Accuracy of train data : 0.8730964467005076
```

5. References:-

https://www.kaggle.com/datasets/elemento/nyc-yellow-taxi-trip-data/data/