New York City Yellow Cab: Processing Large Datasets Project Report

Introduction:

New York City Yellow Cab is the hallmark of New York City. Anyone who has ever been there or seen NYC in a movie, has clearly seen the yellow taxis that people raise their hand to stop. With the rise of ride sharing companies like Uber and Lyft, NYC Yellow Cab too is not untouched. It has taken a hit. With the trip record dataset of this yellow cab, this project focuses on the January 2025 dataset to draw useful insights and transform a big taxicab dataset into actionable insights for stakeholders.

Problem Statement:

NYC Yellow Cab is a big company run by NYC Taxi Cab Commission which handles many cabs and medallions each year. There are millions of trips every month with hundreds of millions of revenue. Raw data is all over the place and too messy. Cleaning and analyzing this data helps NYC stakeholders to see the details of these trips and ultimately the revenue in detail.

Objectives:

- Data Acquisition and Preprocessing: Collecting the dataset, cleaning duplicates, handling outliers, missing values, standardizing datetime formats, creating necessary columns if needed.
- Data Processing and Analysis: Exploring trip demand patterns by time and location, analyzing fare and revenue distribution across various features.
- Storage and Visualization: Storing the cleaned data efficiently and presenting insights using clear visualizations and report.

Methodology:

- Pandas library in python was used to effectively clean and process the dataset. It was also used to save the cleaned dataset in parquet format which handles large file sizes.
- Pyspark was also used to visualize the dataset but was not used for the entirety of the analysis.
- Matplotlib and Seaborn in python were used for visualizations.
- Optimization of memory usage by saving data in Parquet format instead of CSV.
- Removal of unnecessary columns with little impact.
- Handling outliers early, reducing dataset size from 3.5 million rows to 3.2 million rows.

- Creating simple pandas ETL pipeline.
- Storing dataset in s3 for future use possibly with PySpark.
- Visualizing dataset with Pyspark.

Results/Analysis:

- RatecodeID is a feature which in context of NYC taxi datasets is the type of fare structure used for the ride. Apparently, NYC cabs have different fare structures for different sectors like Standard, or going to John F Kennedy(JFK) airports, or negotiated fares between passenger and driver outside of standard operating areas.
- More congested areas like core NY city has something called congestion surcharge which is not much, just 2 and half dollars, but still applies to a lot of trips.
- Just going to the airport has a airport fee around 2 dollars.
- Some unrealistically high and low values are seen in large datasets like this one - which could be an indication of unreliable data entry processing system in the cab company.
- Demand of taxi cab peaks during morning and evening rush hours with lowest demand between 3 am and 5 am.
- Weekdays especially Wednesday to Friday see higher demands than weekends.
- Trip on a daily basis drops on Jan 1st, which is new years day obviously - NYC is too crowded by foot on that day itself.
- Expensive trips are less. Most trips peak between 10 to 15 dollars.
- Some pickup zones - maybe airports have really high average fares. Others are usually around 30 dollars mark.
- People like using credit card and the new FlexFare trip introduced in January 2025 more followed by cash. Flex Fare trip in a New York City taxi is a type of upfront, fixed-price fare for yellow taxis, similar to what is offered by ride-hailing apps like Uber and Lyft. This pilot program, proposed to become permanent by the NYC Taxi and Limousine Commission (TLC), allows passengers to see the total fare before the trip begins, providing a predictable cost rather than a metered fare that can change based on traffic or distance.
- Short(0-2 miles) and medium(2-5miles) dominate the trip distances compared to others. Very long trips like over 10 miles, which could be from JFK airport to Queens for example - do generate high revenue too.

Recommendations for Stakeholders:

- Demand is highest in the morning and evening, so more taxi allocation during these times could decrease wait times and increase revenue.
- Weekends bring less money, so offer discounts to discourage people from using metro and other mass transportations on weekdays so that it brings more revenue to the company during weekdays - like daily loyalty discounts for certain office going people.
- Airports showed the highest pickup zone revenue. So this is the key area to make money. Ensure more taxis are there when needed.
- Again, credit cards and flexfare are on demand for payment choices. This is faster and safer. All taxis must have this facility.
- Instead of congestion surcharge, maybe build a system for high demand time surcharge. Like evenings and mornings, where people use taxis more.

Conclusion:

In conclusion, handling the NYC taxi cab dataset was an experience. There were many hurdles along the way especially in being able to work with pyspark. A small attempt was made but Pandas came in handy as it was able to process data with millions of rows too.

Although the dataset did not have any duplicates, it did come with a lot of null values in multiple columns. As these features were important for the analysis that I was planning to do, dropping them was not an option. Hypotheses were made about how to handle them. Data was analysed well before coming to any decision which was a good learning curve.

Interesting aspect was coming across negative values in financial terms, like negative airport surcharge. Although the actual reason will remain a mystery, thinking over it if it was a refund issue or a data entry issue and handling them was challenging.

Although date and time were already in US format, still forcing them to the datetime format as needed was done.

It was amazing to see how unrealistically high and low values are usually present in a dataset. This could be an indication of lack of proper funding to enter the datas, or system issues. Although some of them could have been real but 90893 distances of under zero is clearly not acceptable in a big corporation like NYC taxi commission.

Data was stored in an amazon s3 bucket and ETL pipeline was created to extract from the local storage. Pyspark was not used, thus despite storing the dataset in s3 bucket, only local pipeline was developed.

Temporal analysis was done to identify patterns, trends and relationships. Having been the first project to analyse and visualize trends and patterns over a course of time, it was a good experience was well to learn and make mistakes and debug.

Grouping distances into categories created was a reminder of the basics of creating labels and bins.

This analysis gave an idea into how to optimize taxi numbers during peak hours like evening, making a weekend vs weekday strategy, making system more robust and better for credit card and flexfare transactions, rethinking about surcharges as well.

Overall this project showed how processing a dataset structurally and analysing it can help in making decisions in transportation and taxi services.