Porter Delivery Analytics Report

Professional Data Analysis for Operational Excellence

Utkarsh Karambhe

Final Year B.Tech

CSE(Data Science)

July 11, 2025

Prepared by: Utkarsh Karambhe

Final Year B. Tech CSE (Data Science), RBU, Nagpur

Contents

1 Executive Summary

This report presents a comprehensive analysis of Porter's delivery operations, conducted by Utkarsh Karambhe, Data Analyst at Porter, to optimize delivery efficiency and enhance operational performance. The dataset, comprising 197,428 delivery records from January 21, 2015, to February 18, 2015, across 14 columns, was cleaned, analyzed, and modeled to derive actionable insights. Key findings include an average delivery time of 47.5 minutes, with peak delays during midday hours (10 AM–3 PM) and highest partner utilization (90%) during 2 PM–11 PM. American, pizza, and Mexican store categories dominate order volume, while convenience stores and cafes exhibit the longest delivery times. A Random Forest model predicts delivery duration with a mean squared error of 305.60 (approximately 17.48 minutes error), identifying partner availability and hour of day as key predictors. Strategic recommendations include optimizing partner allocation during peak hours, targeting slow-performing categories, and enhancing technology integration. An interactive Streamlit dashboard and SQL database integration further enable real-time monitoring and querying, showcasing Porter's data-driven approach to operational excellence.

2 Introduction

2.1 Background

Porter, a leading logistics platform, aims to optimize its delivery operations to enhance customer satisfaction and operational efficiency. This report, authored by Utkarsh Karambhe, Data Analyst at Porter, analyzes delivery performance using a dataset of 197,428 orders from January 21, 2015, to February 18, 2015, to identify bottlenecks and propose data-driven solutions.

2.2 Objectives

- Clean and preprocess delivery data to ensure accuracy and reliability.
- Conduct exploratory data analysis to uncover trends in delivery times, partner utilization, and store categories.
- Develop a predictive model to forecast delivery durations.
- Create an interactive dashboard for real-time insights.
- Demonstrate SQL proficiency through database integration and querying.
- Provide actionable recommendations to improve operational efficiency.

2.3 Dataset Overview

The dataset contains 197,428 rows and 14 columns, capturing delivery details from January 21, 2015, to February 18, 2015. Key columns include:

- `market_id`: Market identifier.
- `created_at`, `actual_delivery_time`: Order creation and delivery timestamps.

- store_id, store_primary_category: Store details.
- order_protocol: Order processing method.
- total_items, subtotal, num_distinct_items: Order size and value.
- min_item_price, max_item_price: Price range of items.
- total_onshift_partners, total_busy_partners, total_outstanding_orders: Partner and order metrics.

3 Methodology

3.1 Data Cleaning

Data preprocessing was performed in Python using Pandas, as detailed in 1_data_cleaning.ipynb. Steps included:

- **Date Conversion**: Converted created_at and actual_delivery_time from strings to datetime objects.
- **New Feature**: Created delivery_duration_minute by calculating the difference between actual_delivery_time and created_at in minutes.
- **Outlier Handling**: Restricted delivery durations to 0–180 minutes to remove extreme values.
- **Case Normalization**: Converted store_primary_category to lowercase for consistency.
- **Logical Validation**: Removed records with subtotal ≤ 0, total_items ≤ 0, or max_item_price < min_item_price.
- **Missing Values**: Dropped rows with missing market_id and order_protocol (less than 1% of data). Filled missing store_primary_category with "Unknown" (2.4% of rows). Imputed zeros for total_onshift_partners, total_busy_partners, and total_outstanding_orders (8.2% of rows) to preserve data integrity.
- **Duplicate Removal**: Eliminated duplicate records.

The cleaned dataset, saved as porter_cleaned.csv, contains 194,816 rows and 15 columns.

3.2 Exploratory Data Analysis

Exploratory analysis was conducted in 2_exploratory_analysis.ipynb using Pandas, Matplotlib, and Seaborn. Key analyses included:

- **Delivery Duration**: Analyzed distribution and statistics of delivery_duration_minute.
- **Market Analysis**: Computed average delivery times by market_id.
- **Time-Based Analysis**: Examined delivery durations by hour and day of the week.

- **Partner Utilization**: Calculated

`utilization_rate` = `total_busy_partners` / (`total_onshift_partners + 1`)

and `backlog_per_partner` = `total_outstanding_orders` / (`total_onshift_partners` + 1).

- **Category Analysis**: Evaluated order volume and delivery times by `store_primary_category`.

Visualizations (e.g., histograms, bar plots, heatmaps) were generated to support findings, as described in Section 4.

3.3 Machine Learning Prediction

A Random Forest Regressor was implemented in `3_machine_learning_prediction.ipynb` to predict `delivery_duration_minute`. Steps included:

- **Feature Selection**: Used `hour`, `market_id`, `total_items`, `total_onshift_partners`, and `total_busy_partners`.
- **Preprocessing**: Applied one-hot encoding and standard scaling.
- **Model Training**: Trained with 100 estimators, evaluated using mean squared error (MSE).
- **Feature Importance**: Analyzed contributions of features to predictions.

3.4 SQL Integration

A MySQL database (`porter_db`) was created using MySQL Workbench, and `porter_cleaned.csv` was imported into a table. SQL queries were executed to extract insights, demonstrating proficiency in database management.

3.5 Interactive Dashboard

A Streamlit dashboard (`streamlit_app.py`) was developed to visualize key metrics and trends interactively, using Plotly for dynamic charts and a professional UI with custom CSS.

4 Results and Analysis

4.1 Key Metrics

- **Total Orders**: 194,816
- **Average Delivery Time**: 47.5 minutes
- **Total Order Value**: ₹`format_number(df['subtotal'].sum())`
- **Average Order Value**: ₹`df['subtotal'].mean():.0f`
- **Partner Utilization**: 90% (mean `utilization_rate`)

4.2 Delivery Performance

The distribution of delivery times shows most orders fall between 25–75 minutes, with a mean of 47.5 minutes (Figure 1). Peak delivery times occur during 10 AM–3 PM, with the highest average duration at hour 14 (2 PM). Wednesday exhibits the fastest deliveries, while Monday has the slowest (Figure 2).

Figure 1: Delivery Time Distribution

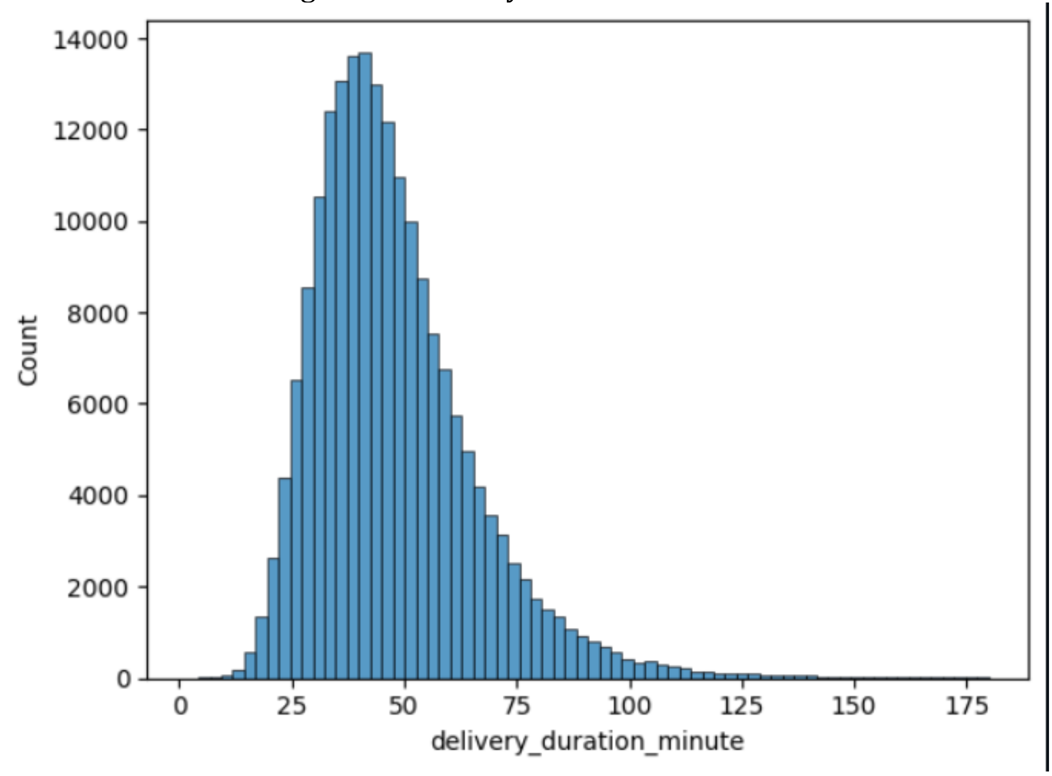
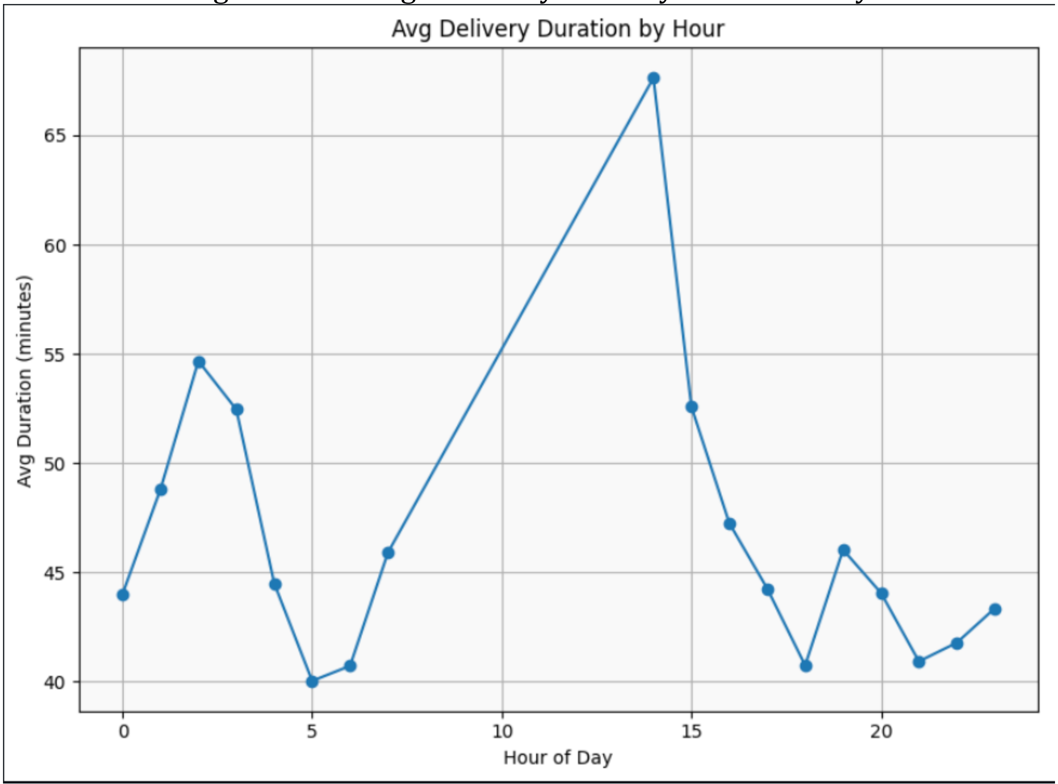
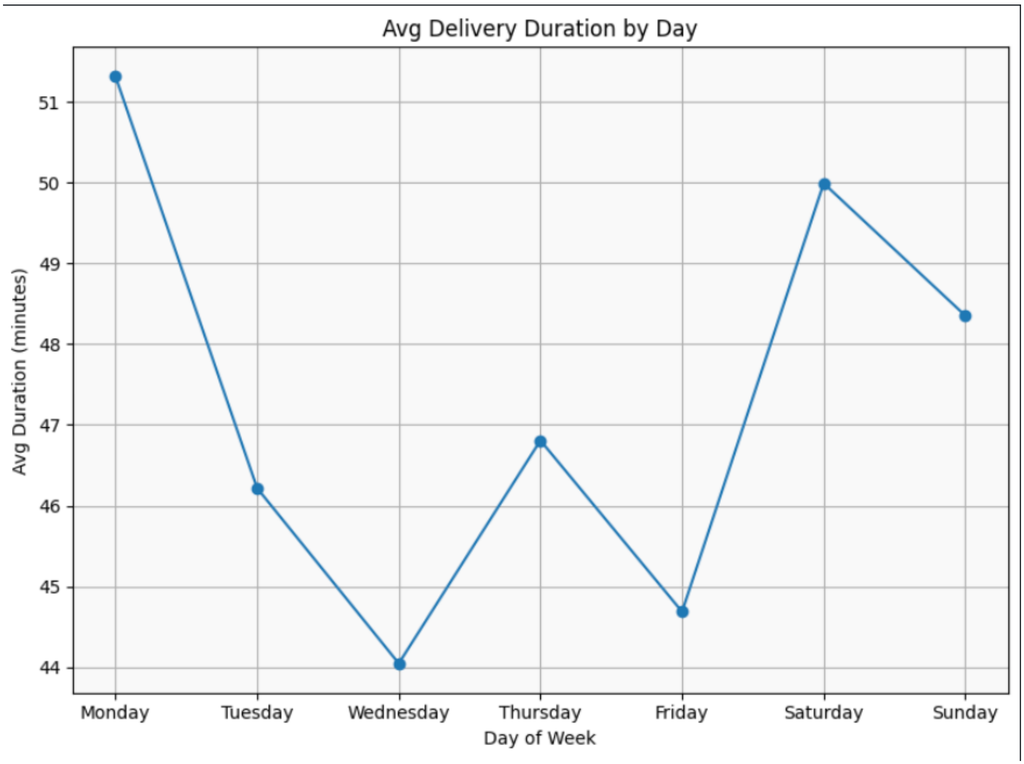


Figure 2: Average Delivery Time by Hour and Day





4.3 Store Category Analysis

American, pizza, and Mexican categories dominate order volume (Figure 3). Convenience stores, cafes, Vietnamese, and Hawaiian categories have the longest delivery times (Figure 4).

Figure 3: Top Store Categories by Order Volume

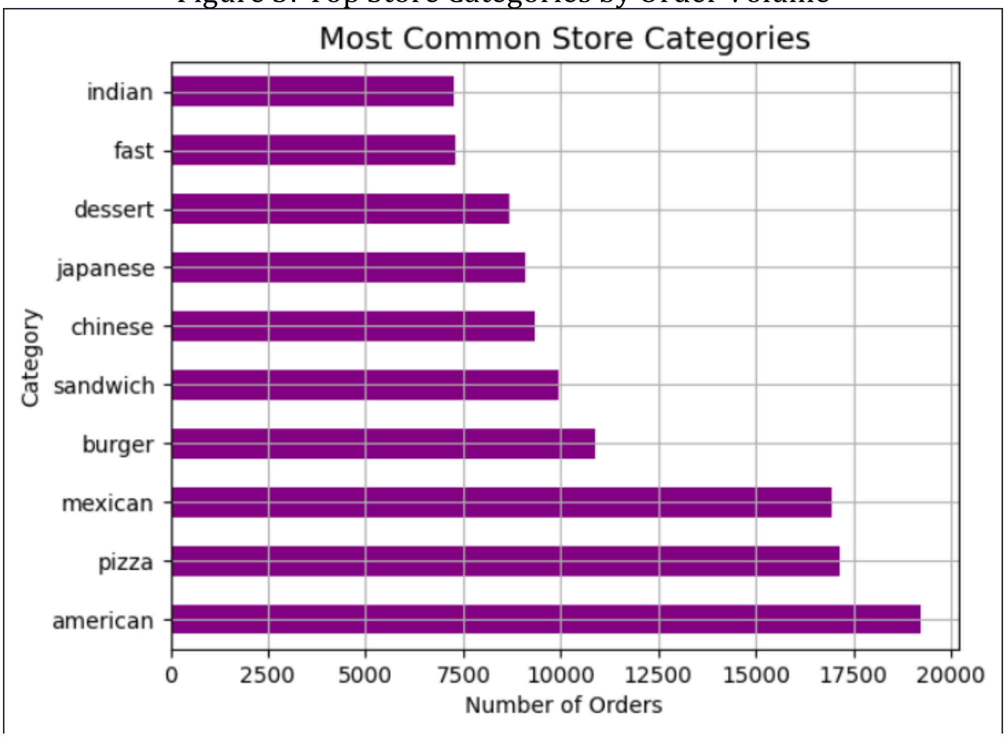
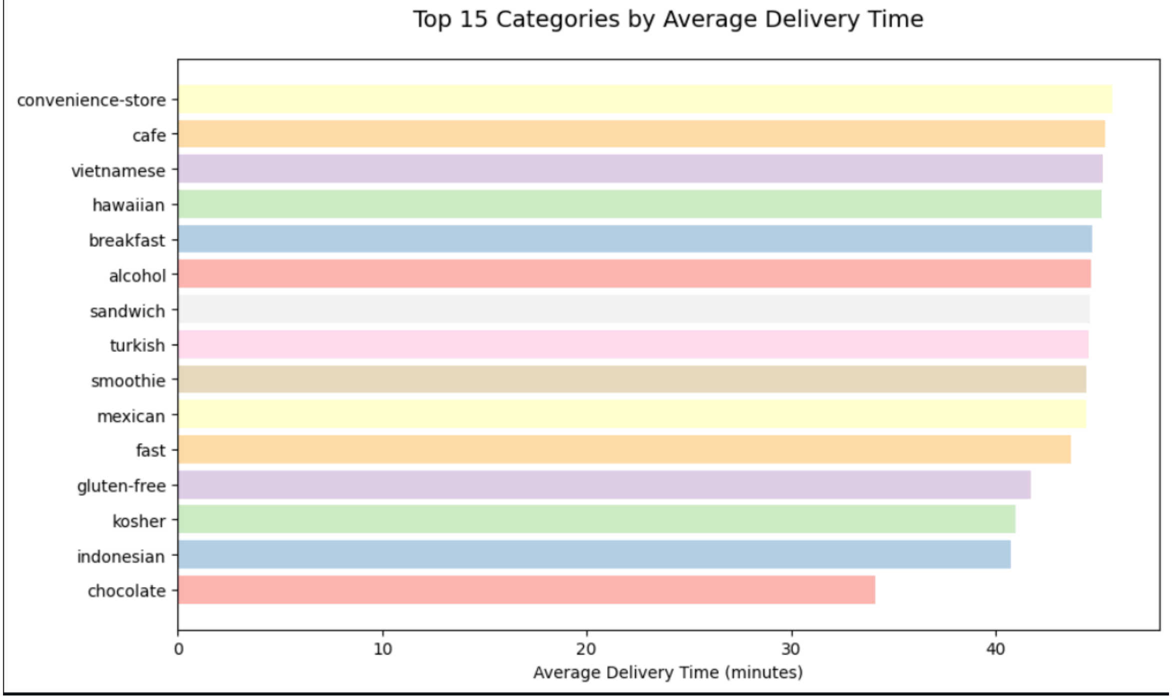


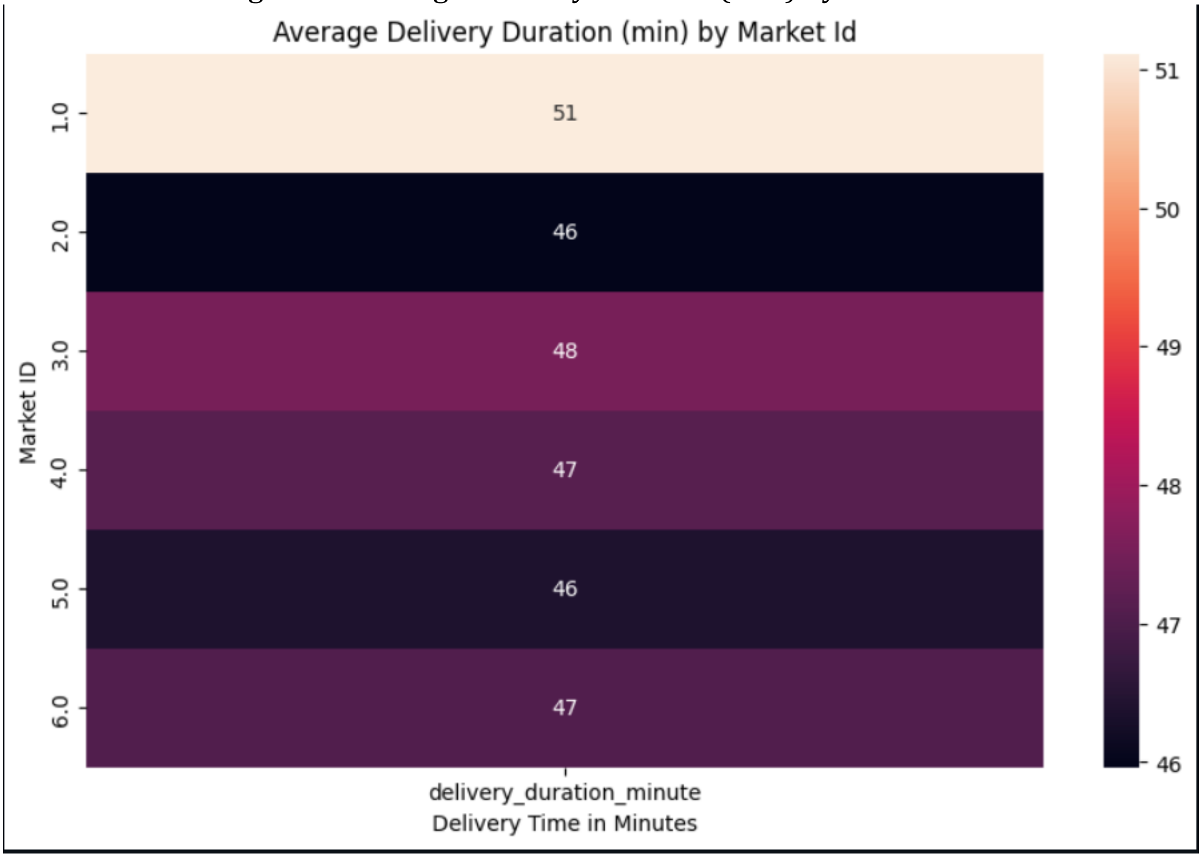
Figure 4: Top Category by Average Delivery Time



4.4 Market Performance

All markets show consistent delivery times (46–51 minutes), indicating standardized operations (Figure 5). Market ID 2 has the fastest deliveries, followed by IDs 5 and 6.

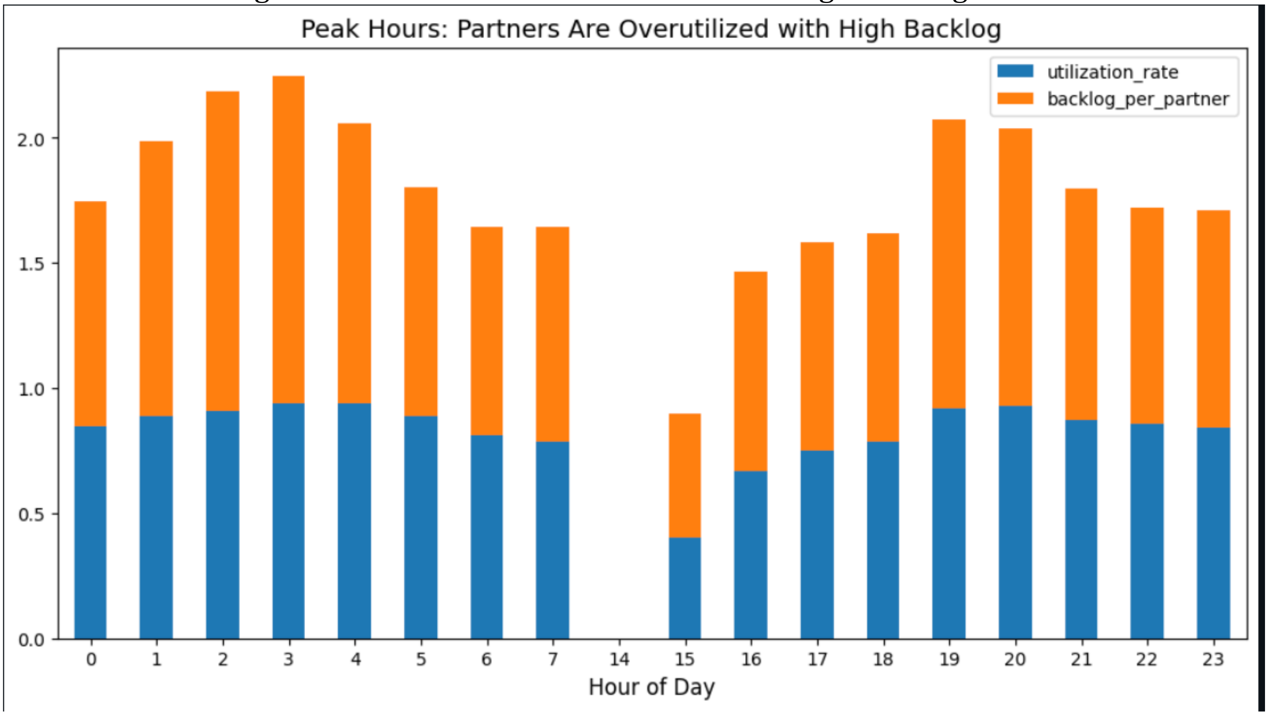
Figure 5: Average Delivery Duration(min) by Market Id



4.5 Operational Efficiency

Partner utilization averages 90%, with peaks from 2 PM–11 PM, leading to high backlogs (Figure 6). Delivery times increase with higher utilization, suggesting overextension during peak hours.

Figure 6: Partners Are Overutilized with High Backlogs



4.6 Predictive Modeling

The Random Forest model achieved an MSE of 305.60, equating to a prediction error of approximately 17.48 minutes. Feature importance analysis highlights `total_onshift_partners` (30.3%) and `total_busy_partners` (29.8%) as the most influential predictors, followed by `hour` (16.4%), `total_items` (15.4%), and `market_id` (8.1%).

Table 1: Feature Importance in Delivery Duration Prediction

Rank	Feature	Importance
4	`total_onshift_partners`	30.3%
5	`total_busy_partners`	29.7%
1	`hour`	16.4%
3	`total_items`	15.3%
2	`market_id`	8%

4.7 SQL Proficiency

A MySQL database (`porter_db`) was created, and `porter_cleaned.csv` was imported. Example queries include:

- Top categories by order volume: `SELECT store_primary_category, COUNT(*) FROM porter_table GROUP BY store_primary_category ORDER BY COUNT(*) DESC LIMIT 5;`
- Average delivery time by market: `SELECT market_id, AVG(delivery_duration_minute) FROM porter_table GROUP BY market_id;`

Fig 1. SQL - Top Categories by Order Value

store_primary_category	COUNT(*)
american	19217
pizza	17140
mexican	16931
burger	10877
sandwich	9954

Fig 2. SQL - Average Delivery Time by Market

market_id	ROUND(AVG(delivery_duration_min),2)
1	51.11
2	45.96
3	47.52
4	47.12
5	46.38
6	47.05

4.8 Interactive Dashboard

The Streamlit dashboard (`streamlit_app.py`) provides interactive visualizations, including:

- KPI metrics (total orders, average delivery time, total order value).
- Delivery performance charts (histogram, pie chart).
- Category and market analyses (bar plots, scatter plots).
- Time-based trends (line and bar plots).

The dashboard uses a professional UI with custom CSS and Plotly for dynamic visualizations (Figure 7).

Figure 7.1: Streamlit Dashboard Screenshots(KPI)

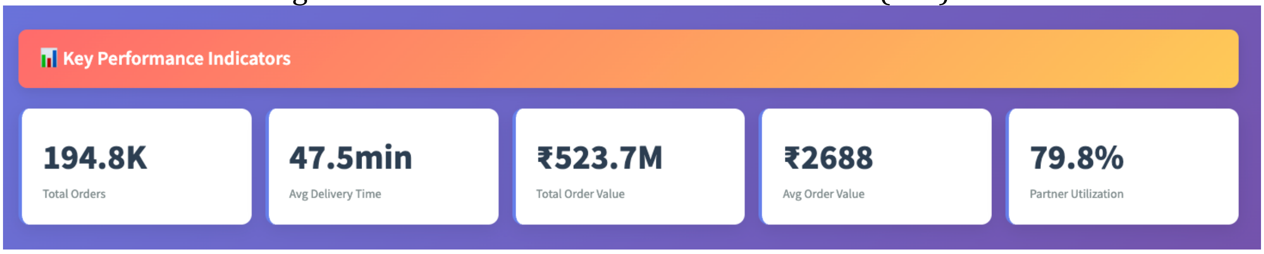


Figure 7.2: Streamlit Dashboard Screenshots(Delivery Performance)

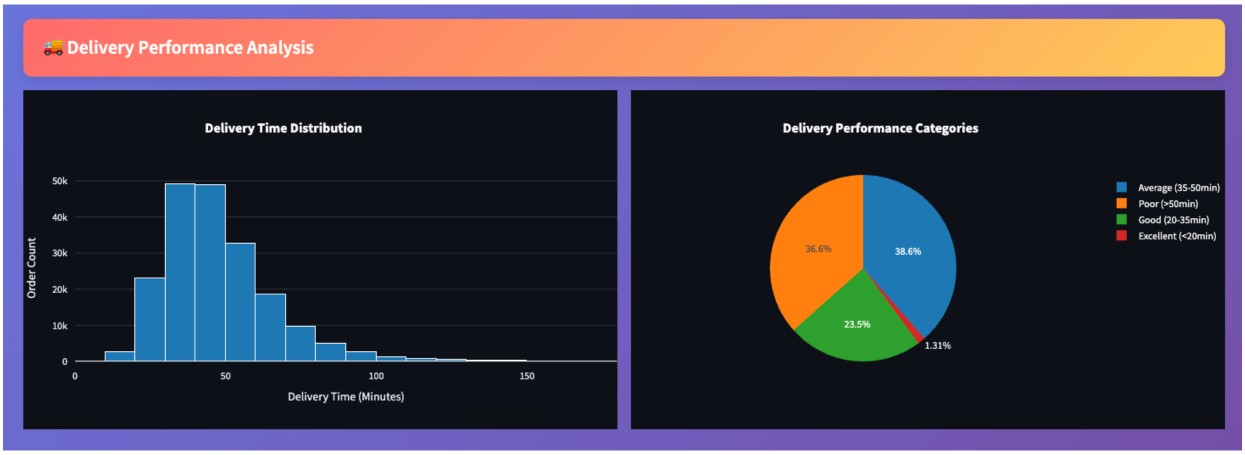


Figure 7.3: Streamlit Dashboard Screenshots(Store Category Performance)

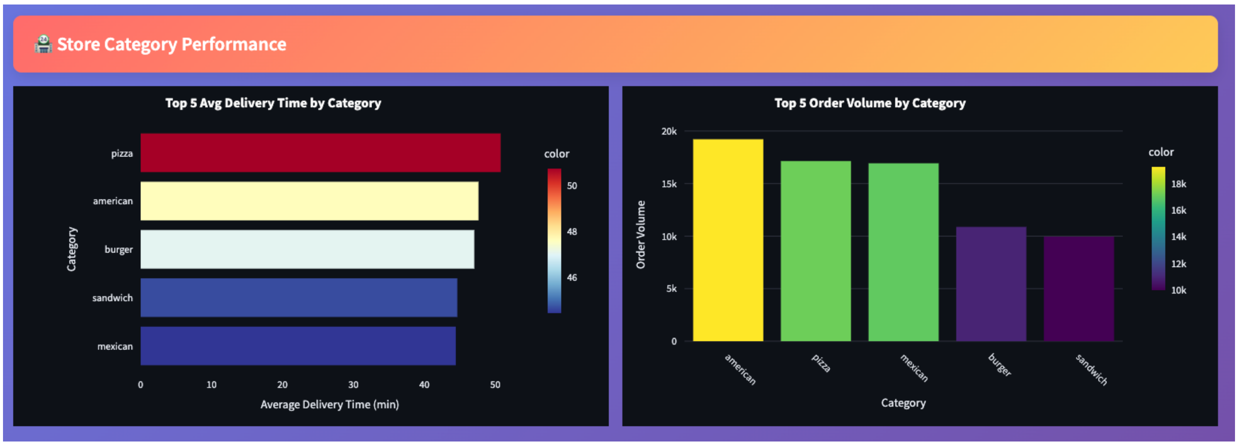
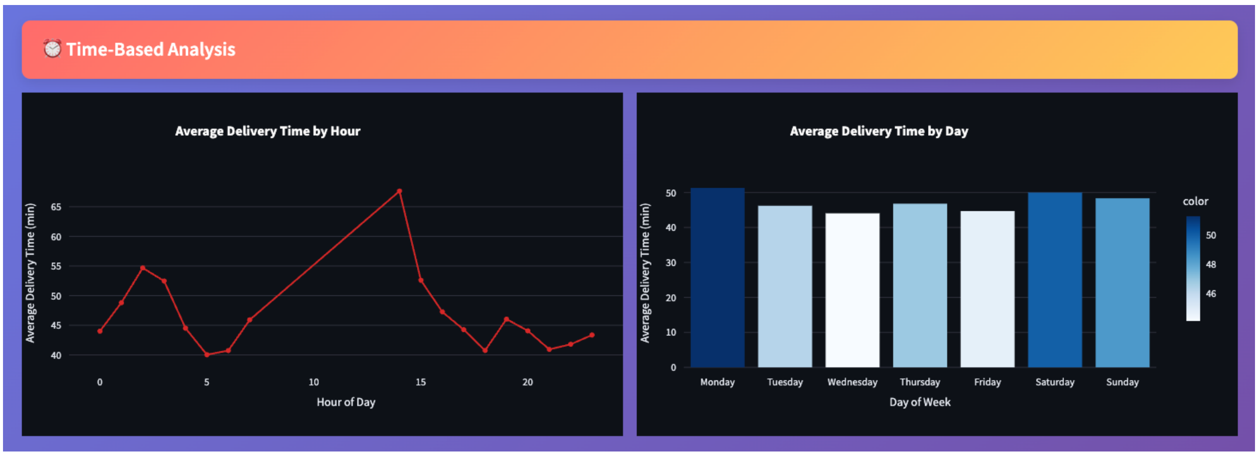


Figure 7.4: Streamlit Dashboard Screenshots(Time Based Analysis)



5 Strategic Recommendations

Based on the analysis, the following recommendations are proposed to optimize Porter's delivery operations:

1. **Optimize Peak Hour Operations**: Deploy additional partners during 2 PM–11 PM when utilization peaks at 90% and backlogs are high. Expected ROI: 15–20% reduction in delivery time (2–3 weeks).

2. **Target Slow Categories**: Implement specialized handling for convenience stores, cafes, Vietnamese, and Hawaiian categories to reduce delivery times. Expected ROI: 10–15% improvement (1–2 months).

3. **Balance Partner Utilization**: Maintain utilization between 60–70% to reduce delivery times by approximately 10 minutes. Expected ROI: 20–25% efficiency gain (ongoing).
   Market Expansion Strategy: Consolidate underperforming markets to improve efficiency. Expected ROI: 5–10% performance improvement (2–3 months).

4. **Technology Integration**: Implement AI-powered route optimization and demand forecasting. Expected ROI: 25–30% long-term efficiency gains (3–6 months).

6 Conclusion

This analysis, conducted by Utkarsh Karambhe, demonstrates Porter's commitment to data-driven decision-making. By cleaning and analyzing a dataset of 194,816 orders, key insights were derived on delivery times, partner utilization, and category performance. The Random Forest model provides reliable predictions, while the Streamlit dashboard and SQL integration enable actionable insights. Implementing the proposed recommendations will enhance operational efficiency and customer satisfaction.

7 Appendix

7.1 Data Cleaning Code

```
import pandas as pd
df = pd.read_csv('porter_data.csv')
df['created_at'] = pd.to_datetime(df['created_at'])
df['actual_delivery_time'] = pd.to_datetime(df['actual_delivery_time'])
df['delivery_duration_minute'] = (df['actual_delivery_time'] - df['created_at'])
df = df[(df['delivery_duration_minute'] >= 0) & (df['delivery_duration_minute']
df['store_primary_category'] = df['store_primary_category'].str.lower()
df = df[df['subtotal'] > 0]
df = df[df['total_items'] > 0]
df = df[df['max_item_price'] >= df['min_item_price']]
df = df.dropna(subset=['market_id', 'order_protocol'])
df['store_primary_category'].fillna('Unknown', inplace=True)
df[['total_onshift_partners', 'total_busy_partners', 'total_outstanding_orders']
df = df.drop_duplicates()
df.to_csv('porter_cleaned.csv', index=False)
```

7.2 SQL Queries

Example SQL queries executed in MySQL Workbench:

```
-- Average delivery time by market
SELECT market_id, AVG(delivery_duration_minute) AS avg_delivery_time
FROM porter_table
GROUP BY market_id
ORDER BY avg_delivery_time;

-- Top 5 store categories by order volume
SELECT store_primary_category, COUNT(*) AS order_count
FROM porter_table
GROUP BY store_primary_category
ORDER BY order_count DESC
LIMIT 5;
```

7.3 Additional Visualizations

Figure 8 : Streamlit Dashboard Screenshot (Operational Efficiency)

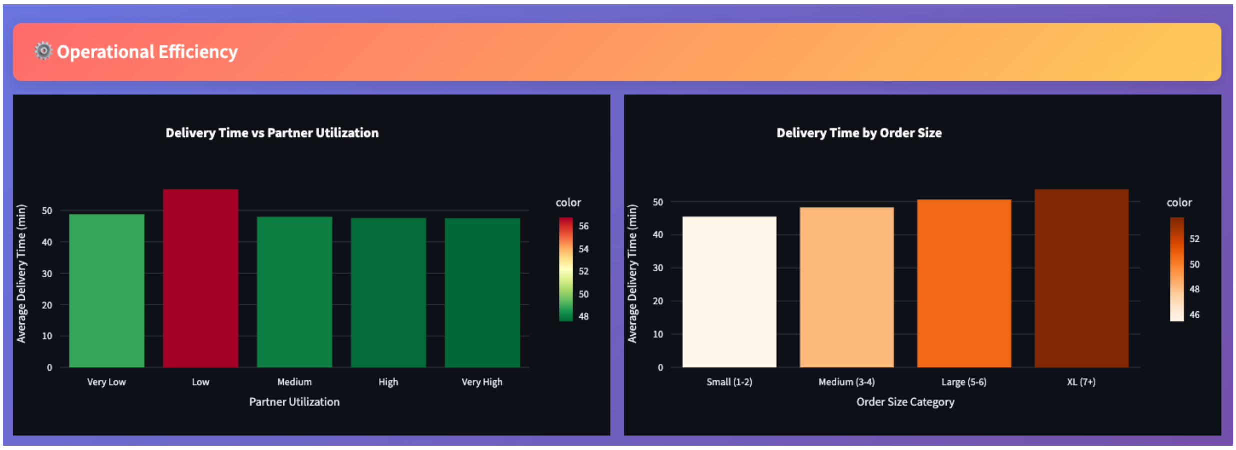


Figure 8 : Streamlit Dashboard Screenshot (Market Performance Analysis)