Online Food Delivery Analysis: Data-Driven Business Insights

1️⃣ Project Overview

This project focuses on analyzing online food delivery data to extract meaningful business insights related to customer behavior, revenue trends, delivery performance, restaurant efficiency, and operational challenges.
The dataset represents real-world noisy data, requiring extensive data cleaning, preprocessing, and feature engineering before analysis and visualization.

The final output includes:

- A cleaned and structured dataset
- Business-oriented analytical insights
- Interactive dashboards using Streamlit / Power BI
- SQL-based validation for scalability and reporting

2️⃣ End-to-End Project Process

🔹 Step 1: Data Collection

- Dataset contains ~100,000 food delivery orders.
- Includes customer details, restaurant information, order values, delivery metrics, payment modes, and cancellation data.

🔹 Step 2: Data Understanding

❖ Initial exploration of column names, data types, and value distributions.

❖ Identification of missing values, invalid entries, and inconsistencies such as:

❖ Missing delivery times

❖ Ratings outside valid range

❖ Inconsistent cancellation records

🔹 Step 3: Data Cleaning & Preprocessing

Data cleaning was performed using Python (Pandas & NumPy):

Key actions:

- Missing values handled using:
    - Median for numerical columns (delivery time)
    - Mean for ratings
    - Mode for categorical fields (payment mode)
- Invalid values corrected:
    - Ratings capped within valid range
    - Logical consistency ensured (cancelled orders not rated)
- Outliers in delivery time and order value treated using statistical limits
- Column names standardized for consistency

Result:
A clean and reliable dataset suitable for analytics.

🔹 Step 4: Feature Engineering

New analytical columns were derived to support business analysis:

- `day_type` → Weekend vs Weekday

- `order_hour` → Extracted from order time
- `peak_hour` → Peak vs Non-Peak demand
- `profit_margin_pct` → Normalized profitability metric
- `delivery_performance` → On-Time vs Delayed

These features enabled deeper insights beyond raw data.

🔹 Step 5: Exploratory Data Analysis (EDA)

EDA was conducted to uncover patterns and relationships:

- Customer ordering behavior
- City-wise and cuisine-wise demand
- Monthly revenue trends
- Impact of delivery time on customer ratings
- Distance vs delivery delay relationship
- Cancellation reasons and frequency

EDA helped validate assumptions and guided dashboard design.

🔹 Step 6: Data Storage (MySQL)

- Cleaned and engineered data was uploaded to MySQL using SQLAlchemy.

- Tables were automatically created with appropriate data types using Pandas `to_sql()`.
- MySQL serves as a scalable backend for:
    - Analytical SQL queries
    - Power BI integration or streamlit
-
    - Streamlit dashboards

🔹 Step 7: Dashboard & Visualization

Two visualization approaches were implemented:

- **Power BI** for enterprise-style BI reporting
- **Streamlit** for interactive, Python-based dashboards

Dashboard features:

- KPI cards (orders, revenue, delivery time, cancellation rate, profit margin)
- Interactive filters and dropdown-based analyst tasks
- Business-focused charts for trends and comparisons

3️⃣ Challenges Faced & Solutions Implemented

Challenge	Solution

Missing and inconsistent data	Applied statistical imputation and logical validation
Noisy real-world dataset	Outlier treatment and data standardization
Complex business questions	Feature engineering for analytical clarity
CSV not scalable for reporting	Migrated cleaned data to MySQL
Multiple analysis requirements	Dropdown-based Streamlit dashboard
Silent pipeline execution	Added logs and verification steps

4️⃣ Key Business Insights

- Weekend demand is significantly higher than weekdays.
- Certain cities and cuisines contribute disproportionately to revenue.
- Delivery delays negatively impact customer ratings.
- Peak-hour demand requires better rider allocation.

- Discounts increase order volume but may reduce profit margins if not controlled.
- Specific restaurants show consistently higher cancellation rates.

5️⃣ Tools & Technologies Used

- **Python:** Pandas, NumPy
- **Visualization:** Streamlit, Power BI
- **Database:** MySQL
- **ORM:** SQLAlchemy
- **IDE:** VS Code
- **Version Control:** GitHub (optional)

6️⃣ Project Outcomes

- Cleaned and normalized analytical dataset
- Business-ready KPIs and insights
- Interactive dashboards for decision-makers
- Scalable backend using MySQL
- Clear documentation ensuring reproducibility and understanding

Sample Code

```python
import numpy as np

def clean_data(df):

   # standarize clum name
   df.columns = df.columns.str.lower().str.strip()

   # missing value
   df['delivery_time_min'].fillna(df['delivery_time_min'].median(), inplace=True)

   df['delivery_rating'].fillna(df['delivery_rating'].mean(),inplace=True)

   df['payment_mode'].fillna(df['payment_mode'].mode()[0],inplace=True)

   df.loc[df['delivery_rating'] > 5, 'delivery_rating'] = 5
   df.loc[df['profit_margin'] < 0, 'profit_margin'] = 0

    # 🔹 Outlier capping (order_value)
   q1 = df['order_value'].quantile(0.25)
   q3 = df['order_value'].quantile(0.75)
   iqr = q3 - q1
   upper_limit = q3 + 1.5 * iqr

   df['order_value'] = np.where(
       df['order_value'] > upper_limit,
       upper_limit,
       df['order_value']
   )

   print(" Data Cleaning Completed")
   return df
```