PYTHON-BASED DAILY DEMAND FORECASTING FOR A U.S. URBAN RIDESHARE AND MICROMOBILITY STARTUP

1. Background

A U.S.-based urban mobility startup operating electric scooters and shared bikes across five cities experienced sharp variability in daily ride volume. Factors like weather, weekdays vs weekends, holidays, and local events influenced demand unpredictably. Their fleet operations team lacked a robust forecasting system to predict usage and plan vehicle deployment effectively.

We were hired to build and compare multiple time series models in Python for daily ride demand forecasting. The output was designed to support route balancing, recharging schedules, and in-app dynamic pricing strategies.

2. Objective

- To forecast **daily ride demand per city** using SARIMA and Prophet models in Python
- To incorporate **exogenous factors** such as weather, holidays, and special events
- To provide a real-time, automated tool that supports **daily fleet deployment and pricing adjustments**
- To deliver forecast visualizations, model comparison, and integration-ready outputs

3. Data Used

Source: Internal trip logs + external weather APIs

Data Coverage:

- Period: Jan 2021 to Dec 2022
- Cities: Austin, Nashville, Denver, Seattle, and Tampa
- Daily granularity

Variables:

- City, Date, Total_Rides
- Temperature, Rain_Indicator, Wind_Speed, Humidity

- Day_of_Week, Holiday_Flag, Major_Event_Flag

4. Methodology

4.1 Data Preprocessing

- Cleaned missing ride counts and outliers using IQR filtering
- Fetched weather data using OpenWeatherMap API
- Merged external event calendars per city for local holidays and sports events
- Created lagged variables and rolling averages (3-day and 7-day)
- Normalized ride counts for visualization purposes

4.2 Models Used

- **SARIMA (Seasonal ARIMA)**: Accounted for weekly seasonality and trend decomposition
- **Facebook Prophet**: Used regressors for holidays and weather
- Evaluated using rolling window validation (last 90 days)
- Forecast horizon: 14 days forward, daily intervals

5. Forecasting Results

City	Best Model	RMSE (7-day forecast)	MAPE
Austin	Prophet	124.7	9.1%
Denver	SARIMA	137.3	11.4%
Nashville	Prophet	98.5	8.6%
Seattle	SARIMA	143.8	10.9%
Tampa	Prophet	116.2	9.8%

- **Prophet** outperformed SARIMA in cities with **event-heavy calendars**
- SARIMA performed better in **weather-sensitive cities** (Seattle, Denver)
- Including rain and temperature as regressors improved accuracy significantly (drop in MAPE by ~3–4%)

6. Insights and Applications

- **Weekend and event days** had predictable surges—Prophet captured these with greater reliability
- **Rain was the single strongest negative predictor** of daily ride volume (avg. 23% drop on rainy days)
- Enabled **30% better planning of battery swap operations** for e-scooter hubs
- Supported the design of **price surge windows** based on predicted demand thresholds

7. Reporting Output

- **Python Notebook**:
  - Modular SARIMA and Prophet modeling scripts with city filter
  - Integration with weather API and event CSV
  - Functions for accuracy tracking and error logging
- **PDF Forecast Report (15 pages)**:
  - City-level forecast graphs with upper/lower confidence bands
  - Model comparison summary table with commentary
  - Visual overlays of ride count vs weather and events
- **Excel Sheet**:
  - Forecasted ride volume for next 14 days (by city)
  - Confidence intervals for demand-sensitive staffing
  - Tags for “Surge Alert Days” based on >20% predicted increase vs average

8. Strategic Impact

- Client used forecast data to reduce **daily fleet repositioning costs by 18%**
- Recharging staff schedules optimized based on next-day usage predictions
- Surge pricing logic was refined using Prophet’s confidence interval outputs
- The Python system is now scheduled to auto-run every night, feeding forecasts into the internal dashboard