ML Project:

Capital Bikeshare Intelligence: Demand Prediction & Optimization Dashboard

Team

- Lakshhmi Narayanan T - 23BAI1048
- Vijay Aravindh S - 23BAI1160
- Mohammed Farhan Rashid - 23BAI1529

The Core Problem: System Imbalance

Bike-share systems are inherently unstable. Natural commuter flows quickly lead to shortages and surpluses across the network, demanding costly manual intervention.

Morning Rush Dynamic

Commuters move from residential areas (deficit) to downtown stations (surplus).

Evening Rush Dynamic

The exact reverse occurs, leaving residential stations overfull and central stations empty.

The Impact of Imbalance

Customer Frustration

Riders find empty docks (deficit) or full docks (surplus), leading to lost business and reduced user trust.

High Operational Costs

Companies spend significant budget on manual rebalancing via diesel trucks, often without an optimized route plan.

Algorithmic Inefficiency

TSP is computationally infeasible for 900+ stations and fails to model the many-to-many flow of rebalancing.

Our Solution: Predict, Then Optimize

Our goal is to deliver an end-to-end intelligent tool that guides bike-share operators toward proactive, data-driven fleet management.

Phase 1: Prediction

Machine Learning models forecast exactly where and when supply/demand issues will occur.

Phase 2: Optimization

Operations Research calculates the minimum-cost rebalancing plan to resolve the predicted issues.

User Input & Scenarios

Prediction Engine (XGBoost)

Optimization Engine (PuLP)

Actionable Dashboard

Step 1: K-Means Clustering for Station Segmentation

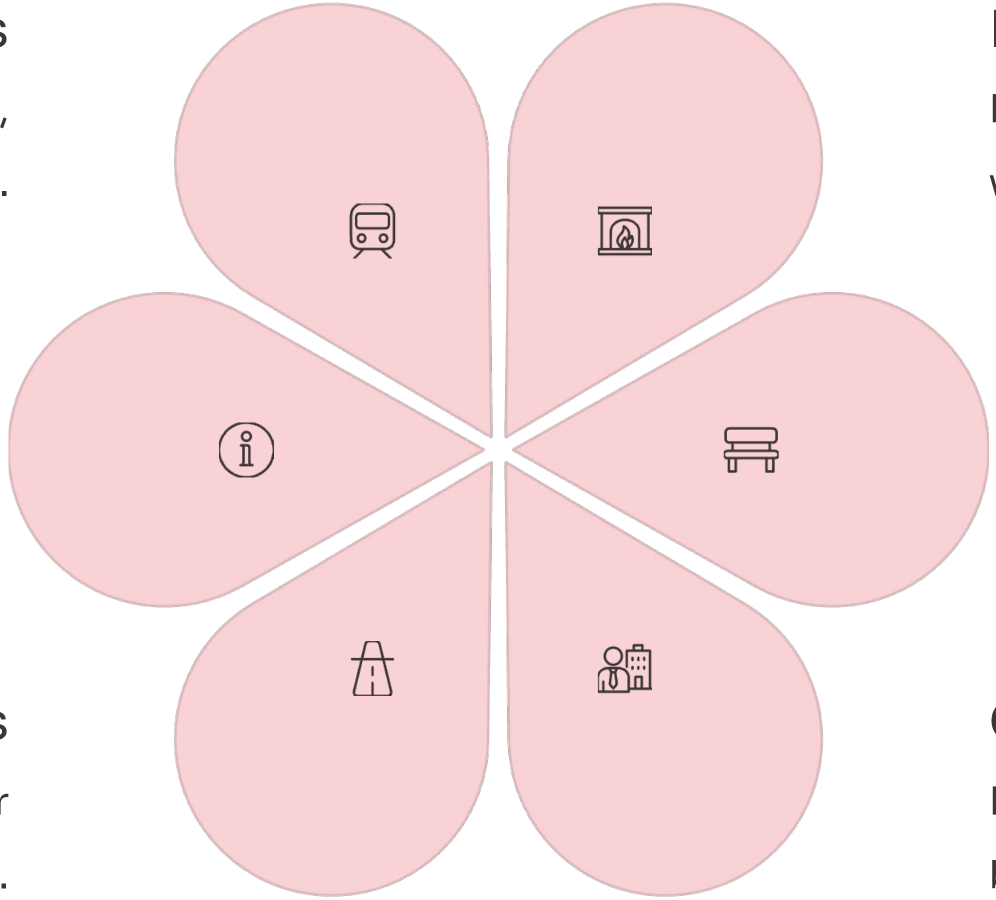
To avoid a "one-size-fits-all" approach, we segmented the entire network into functional groups based on usage behavior.

Commuter Hubs

High volume, strong AM/PM peaks, weekday-heavy.

Residential Zones

High AM departures, high PM arrivals, lower weekend use.



Central Core

Our analysis grouped all 913 stations into 5 distinct clusters.

Leisure Spots

High weekend and midday demand, near parks or attractions.

Transit Connectors

Mid-volume, highly variable use, near bus or metro lines.

CBD/Work Stations

High AM arrivals, high PM departures, central business district focus.

This segmentation allows us to train **specialized prediction models** that are significantly more accurate than a single, generalized model.

Step 2: Cluster-Based XGBoost Prediction

We trained five distinct XGBoost Regressor models, ensuring each model is an expert at predicting demand for its specific station cluster.

Prediction Model Inputs

- **Temporal Features**

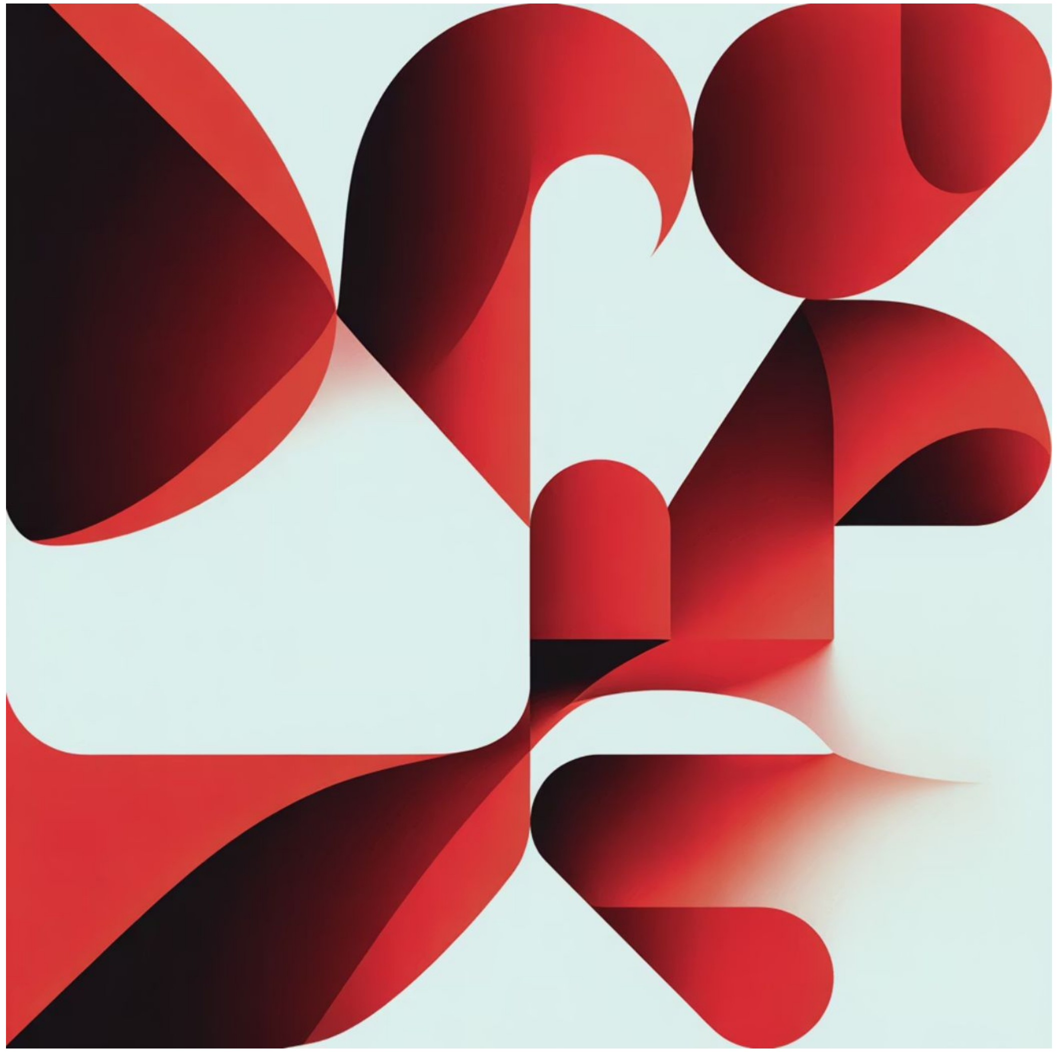
  Hour, Day of Week, and Month were included, with cyclical encoding (sin/cos) to capture continuity (e.g., 23:00 is close to 00:00).

- **Weather Variables**

  9 weather features, including Temperature, Feels Like, Precipitation, Wind Speed, and Solar Radiation, were used.

- **Output & Precision**

  The model outputs a precise decimal-value prediction of expected demand (e.g., 1.85 bikes) for every station at the target hour.



Step 3: Rebalancing Optimization (PuLP)

The predictions feed directly into a Mixed Integer Linear Programming (MILP) model to determine the most cost-effective rebalancing solution.

1

Objective Function

Minimize the **Total Cost**, which is calculated as: (Transport Cost × km) + (Shortage Penalty × bikes short).

2

Shortage Penalty

A configurable penalty (**$10 per bike**) is applied for failing to meet predicted demand, quantifying the cost of a lost customer.

3

Transport Constraint

Transport cost is set at **$0.50 per km**, incentivizing shorter, consolidated routes for the truck fleet.

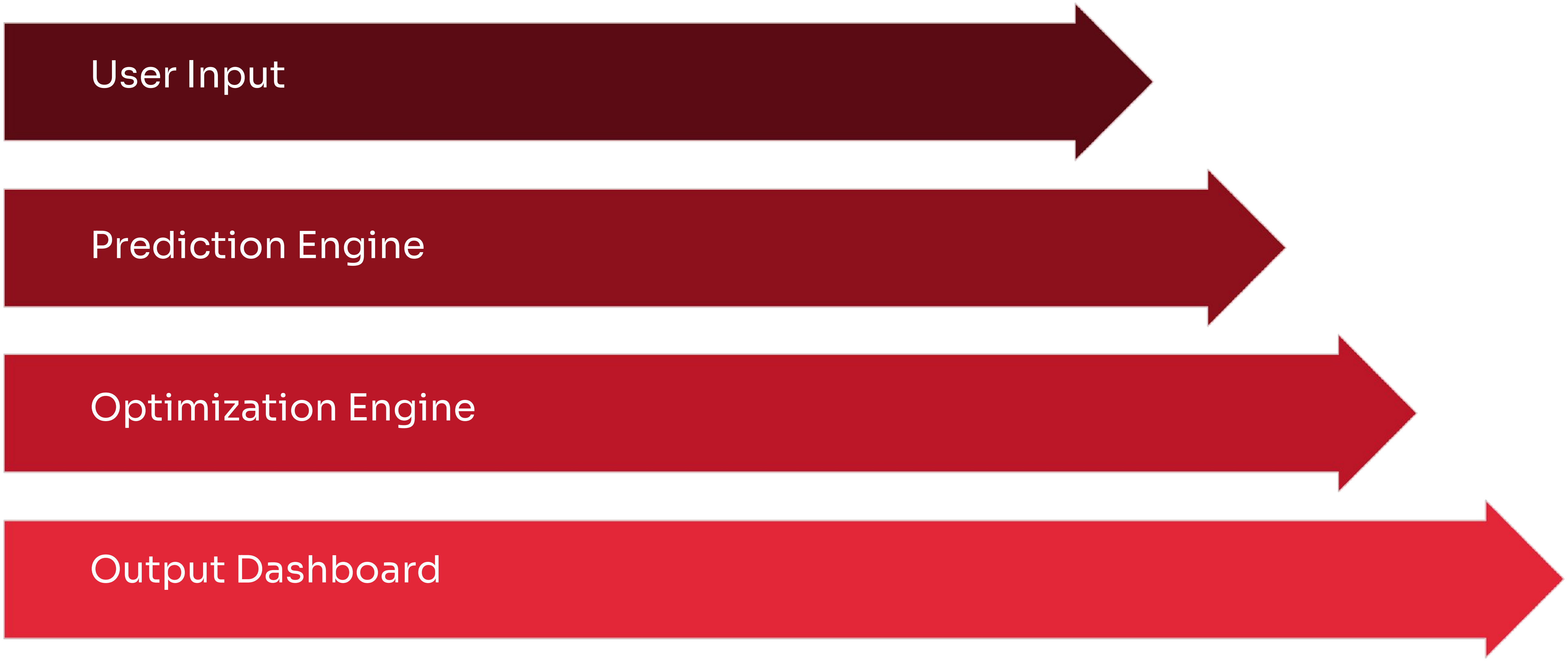
4

Capacity Constraints

The model ensures trucks only move bikes they have (Supply limit) and do not overfill station docks (Capacity limit).

End-to-End System Architecture

The system is built as a cohesive pipeline, moving from user defined scenarios to actionable rebalancing plans.



The Deployed Solution: Interactive Dashboard

The final output is an interactive Gradio dashboard designed for operators to run "what-if" simulations and generate live rebalancing reports.

Key Operator Features

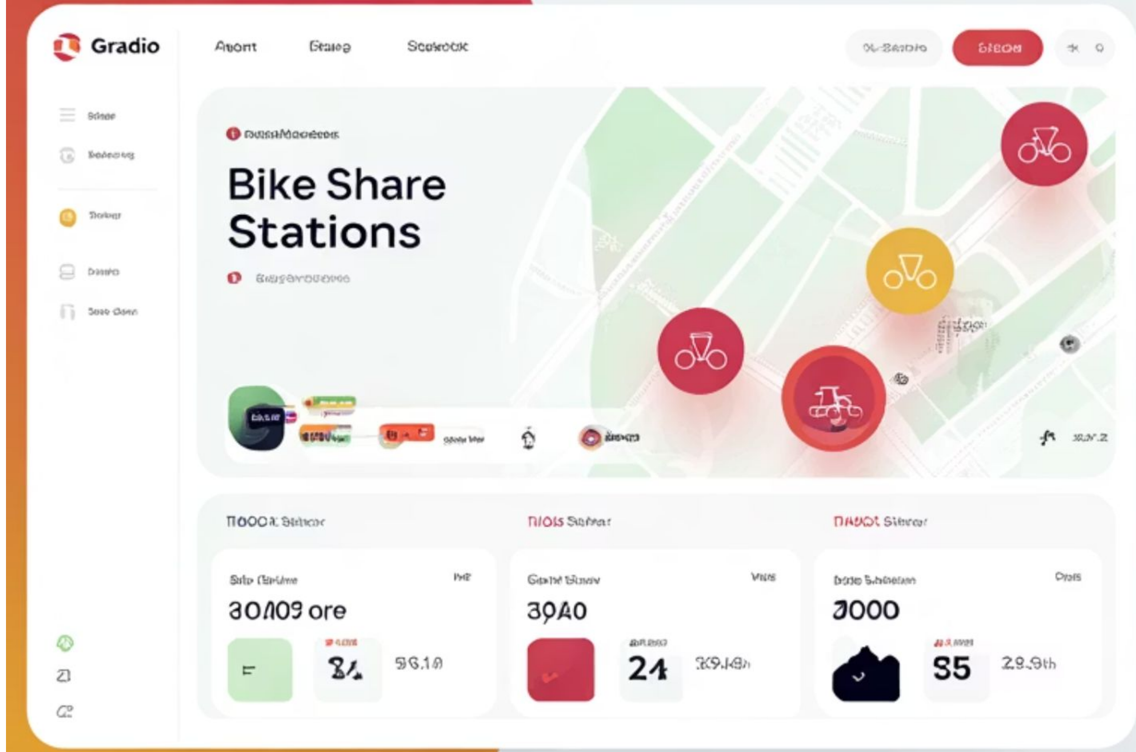
Scenario Control: Operators can adjust time, total bike count, and all 9 weather inputs.

Interactive Map: Visual display of predicted station status (Red: Deficit, Green: Surplus).

AI-Generated Insights: Plain-language recommendations and summaries of the rebalancing requirement.

Actionable Reports

The system generates a downloadable **rebalancing_plan.csv** containing precise move instructions for truck drivers (e.g., "Move 3 bikes from Station A to Station B").



Spaces LakshhmiNarayanan/MLProject-Bike_Predicting_Optimization like 0 Running App Files Community Settings

🚴 Capital Bikeshare Intelligence Dashboard

Demand Prediction & Operations Optimization

This dashboard uses XGBoost with K-Means clustering and does Operations Optimization (Mixed Integer Linear Programming) to predict bike demand and optimize rebalancing operations.

⚙️ Simulation Controls

📅 Date & Time Settings ▼

📅 Date (YYYY-MM-DD)

Select any date to simulate

2024-08-15

🕗 Hour of Day

Peak hours: 7-9 AM, 5-7 PM on weekdays

08:00

🚲 Bike Availability ▼

🚲 Total Bikes in System 6500

Set the total bike availability across all 913 stations

0 30000

🌧️ Weather Conditions ▼

📊 Simulation Results

📋 Simulation Summary

Date & Time: Thursday, August 15, 2024 at 08:00 AM

Bike Availability: 6500 total bikes (Avg: 7.1 per station)

Weather Conditions:

- Temperature: 20°C (Feels like: 20°C)
- Humidity: 60% | Wind: 10 km/h
- Precipitation: 0 mm | Cloud Cover: 50%
- Solar Radiation: 200 W/m² | UV Index: 5

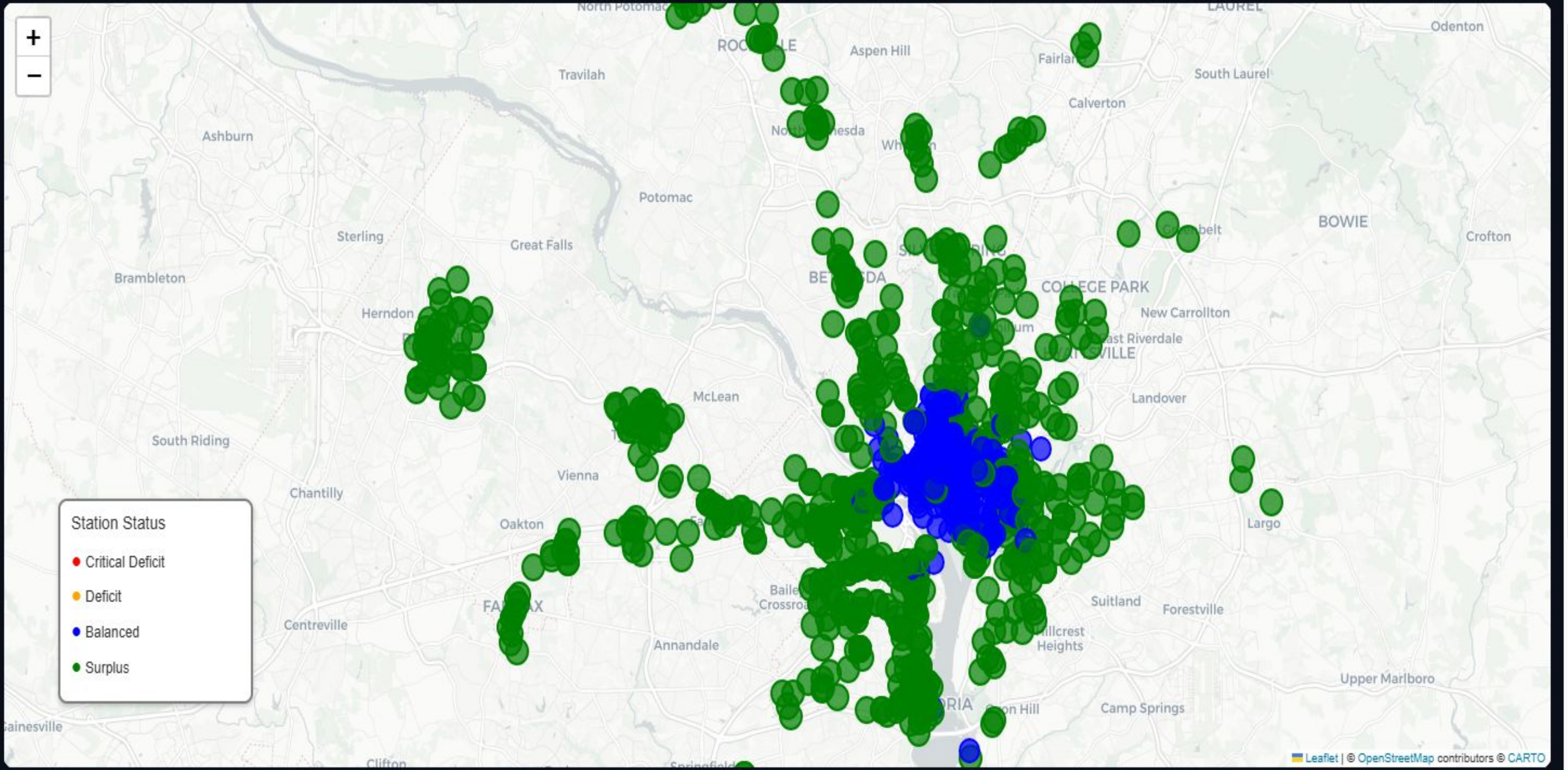
Key Metrics:

- Total Moves Required: **0**
- Total Cost: **$0.00**
- Bikes to Relocate: **0**

🗺️ Interactive Station Status Map

Station Status Map: Shows current bike availability vs predicted demand

- 🔴 **Red circles** = Critical deficit (need bikes ASAP)
- 🟠 **Orange circles** = Deficit (need bikes)
- 🔵 **Blue circles** = Balanced (good condition)
- 🟢 **Green circles** = Surplus (extra bikes available)
- *Circle size indicates severity of imbalance*
- *Click on markers for detailed information*

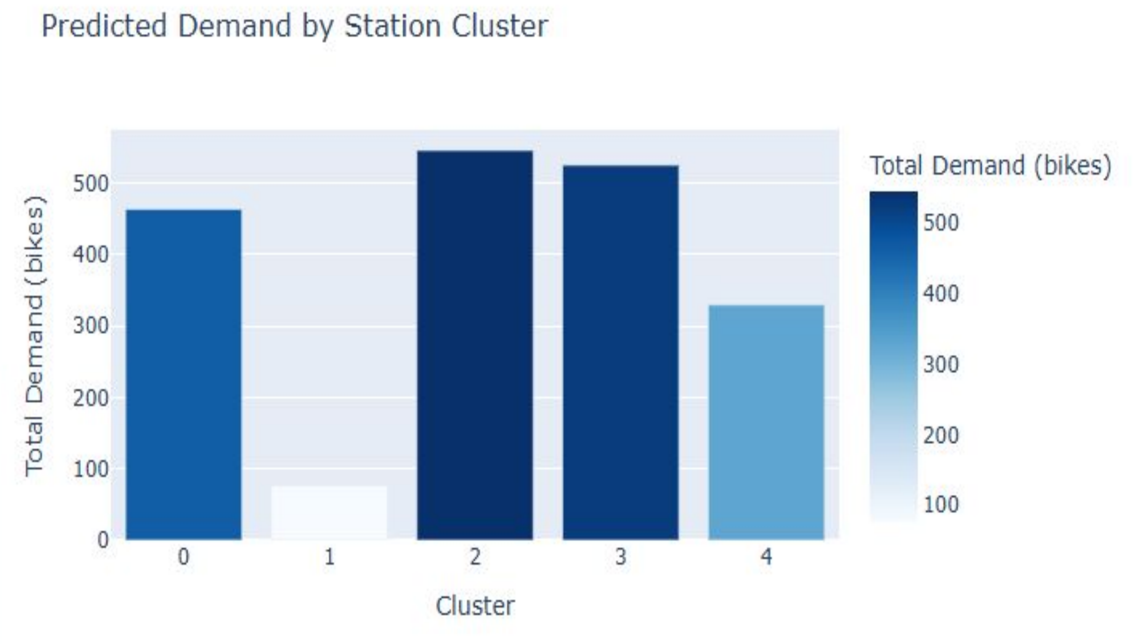


📈 Analytics Dashboard

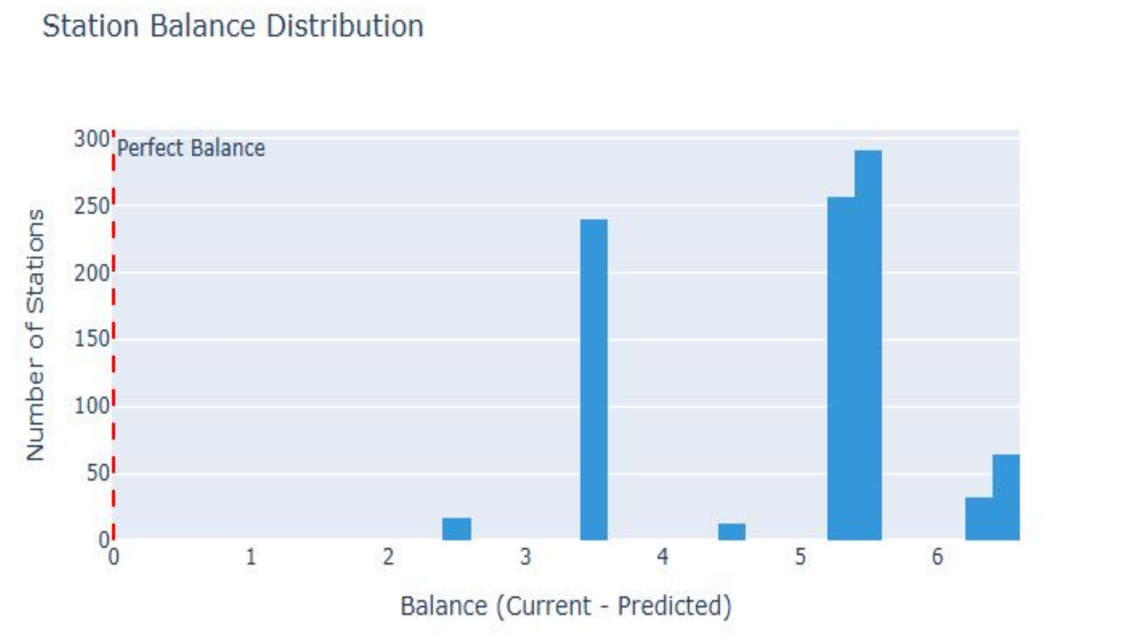
Three analytical views to understand patterns:

1. **Demand by Cluster** - Which station types need most bikes
2. **Balance Distribution** - How well-balanced is the system
3. **Move Analysis** - Relationship between distance and bikes moved

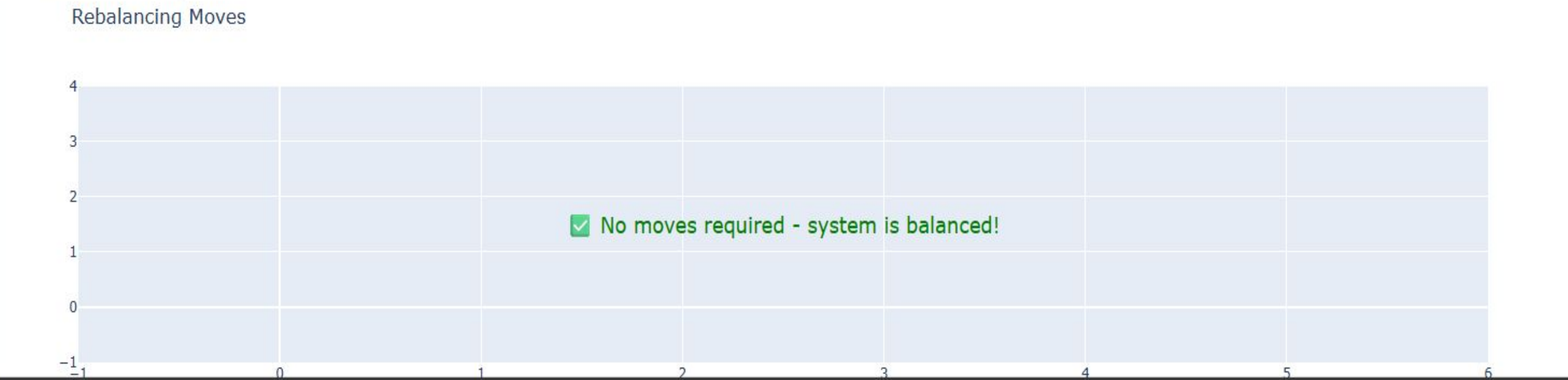
Demand by Cluster



Balance Distribution



Rebalancing Move Analysis



📥 Download Rebalancing Plan

CSV file contains:

- All moves with station names
- Number of bikes to move
- Distances and costs
- GPS coordinates for navigation
- Priority information

Use this for:

- Operations planning
- Truck route optimization
- Cost estimation
- Performance tracking

Complete CSV Report

no_moves_needed.txt 40.0 B

📍 Try These Pre-Configured Scenarios

Click any scenario below to automatically fill in the parameters:

Scenario Examples

📅 Date (YYYY-MM-DD)	🕗 Hour of Day	🚲 Total Bikes in System	🌡 Temperature (°C)	🤔 Feels Like (°C)	💧 Humidity (%)	💨 Wind Speed (km/h)	🌧 Precipitation (mm)	☁ Cloud Cover (%)	🌞 Solar Ra
2024-08-15	08:00	13695	22	22	65	8	0	30	
2024-08-15	17:00	13695	28	30	70	12	0	40	
2024-08-17	14:00	13695	25	26	60	10	0	50	
2024-08-15	23:00	13695	18	17	75	5	0	10	
2024-12-15	08:00	13695	5	3	80	15	2	90	
2024-08-15	08:00	9130	22	22	65	8	0	30	
2024-08-15	08:00	18260	22	22	65	8	0	30	

📊 Technical Details & Performance Metrics

Results & Business Impact: High Accuracy, Lower Cost

Model Performance

~0.2

R^2 score

~0.18

R^2 score

~1.28

Average Error

We predict hourly demand with an average error of only 1.28 bikes per station.

Optimization Value

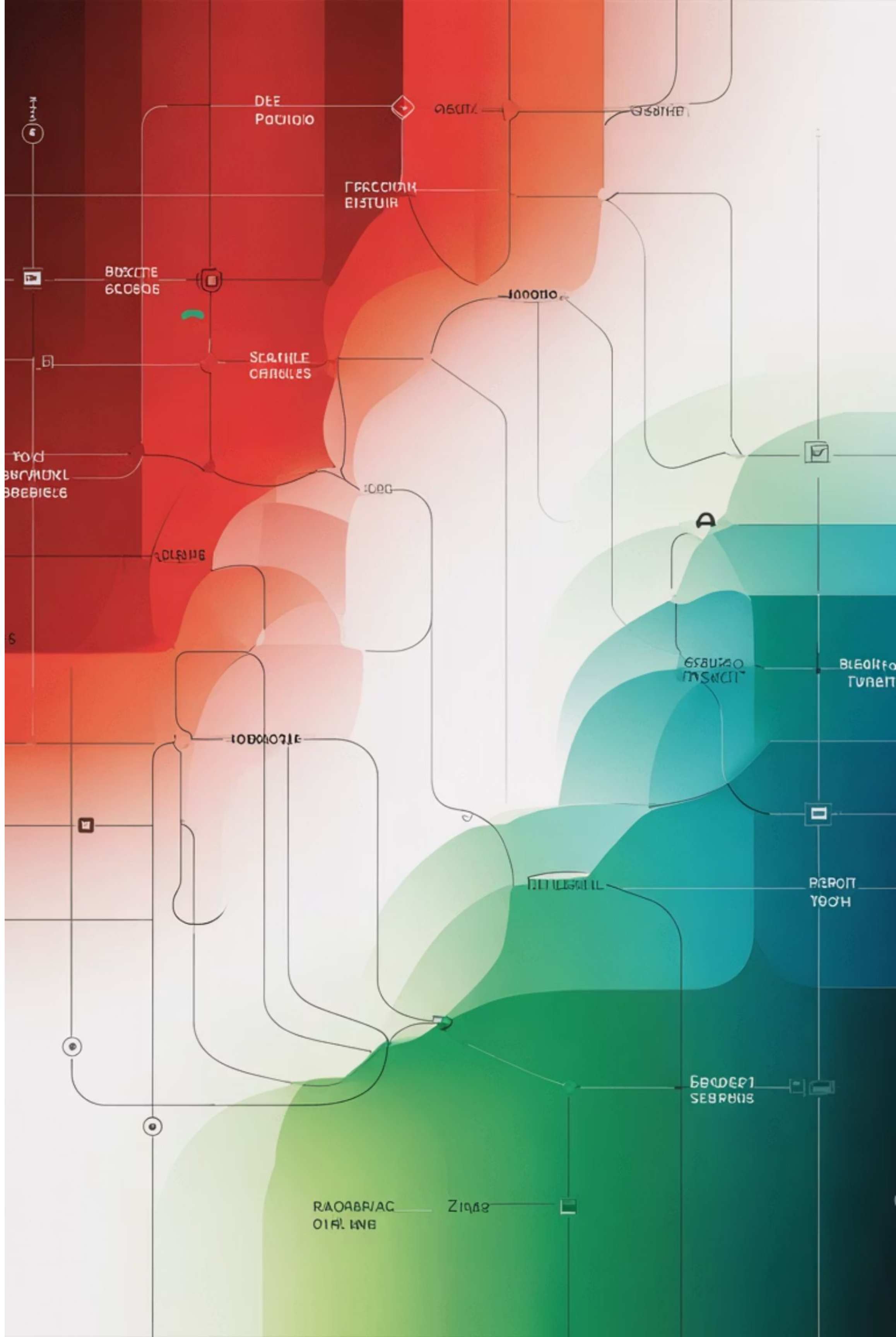
Reduces Deficits

Our model eliminates over **90%** of predicted deficit stations in typical scenarios.

Proactive Operations

Moves the operation from reactive crisis management to predictive logistics, hours in advance.

The system ensures bikes are precisely where customers demand them, preventing lost sales and saving significant operational costs on rebalancing runs.



Thank You

View Project Codebase