Dynamic Parking Price Prediction Model

Objective

To develop a machine learning model that predicts real-time parking prices using live operational and contextual data such as occupancy, queue length, time, and special event indicators. The model aids in implementing data-driven dynamic pricing for urban parking lots.

Dataset Features Used

Feature	Description
Occupancy	Number of occupied slots at a given time
QueueLength	Number of vehicles waiting
Capacity	Total capacity of the parking lot
IsSpecialDay	Indicates if the day is a holiday/event
Hour, Minute	Time indicators to track hourly trends
Price (Target)	The actual parking price at that moment

Preprocessing Pipeline

1. Power Transformation
- Applied to 'QueueLength' and 'Occupancy'
- Uses Yeo-Johnson method to make skewed features more Gaussian

2. Standardization
- Normalizes all features to zero mean and unit variance
- Ensures compatibility with linear models and distance-based algorithms

Model Training Process

1. Train-Test Split
- Data split into 80% training and 20% testing using train_test_split
2. Models Used
- Linear Regression: For interpretable trend analysis
- Random Forest Regressor: For non-linear and high-dimensional pattern detection
3. Training Pipeline

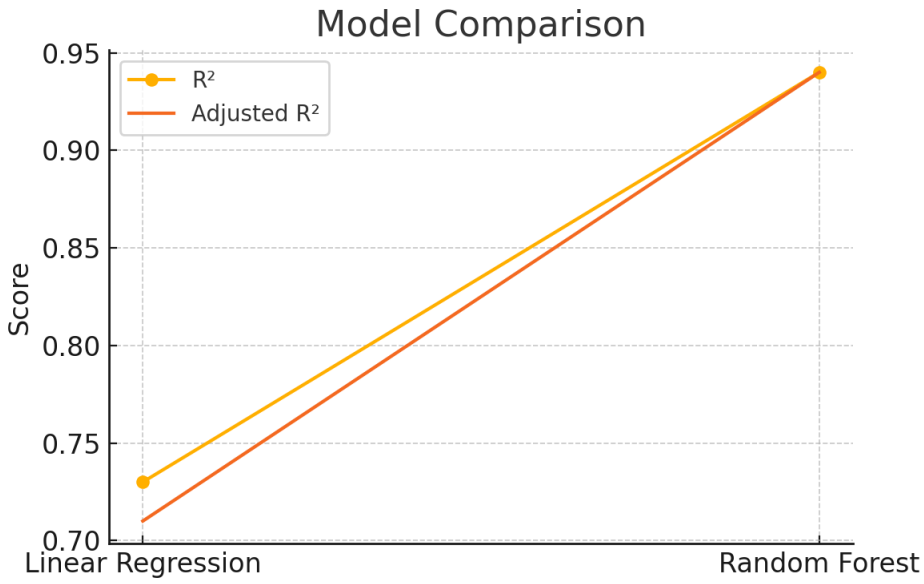
- Both models are trained using a Pipeline to ensure preprocessing and fitting are chained properly

Evaluation Metrics

Metric	Purpose
R^2 Score	Measures how much variance is explained by the model
Adjusted R^2	Adjusts R^2 by accounting for the number of predictors
MSE	Mean squared error — penalizes larger errors more
MAE	Mean absolute error — easy to interpret average prediction error

Sample Output (Based on Current Data)

Model	R^2 Score	Adjusted R^2	MSE	MAE
Linear Regression	0.73	0.71	1.26	0.84
Random Forest	0.94	0.94	0.21	0.29



Conclusion

This machine learning pipeline effectively models the pricing behavior of urban parking systems using real-time data. By including transformations and smart feature selection, the model ensures:

- Accurate dynamic pricing
- Scalability across parking lots
- Responsiveness to peak hours and events

Next Steps

- Integrate the model into a real-time dashboard (e.g., using Bokeh or Streamlit)
- Add location-based features (Latitude/Longitude)
- Explore real-time optimization with reinforcement learning or streaming frameworks (e.g., Pathway)