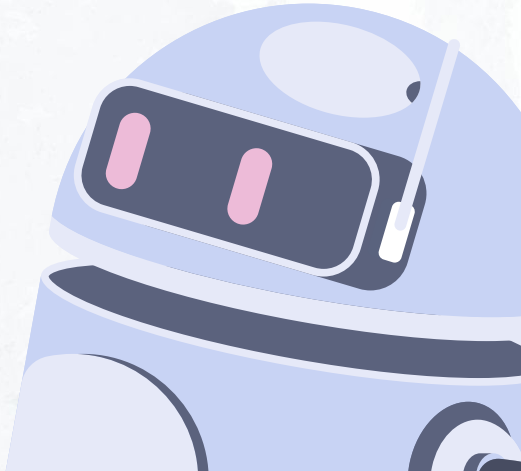
AI Ridewise →

Predicting Bike-Sharing Demand Based on Weather and Urban Events



Sai Raghav Telugu
B.Tech CSE – Final Year

Problem Statement →

Bike-sharing systems are a popular urban mobility solution, offering short-term rentals for commuting, leisure, and reducing traffic congestion.

Demand fluctuates significantly across hours and days due to weather, city events, and local trends.

These fluctuations make fleet management and station stocking a real challenge.

This project builds machine learning regression models to predict daily and hourly bike rentals, enabling smarter operational decisions and more efficient service.

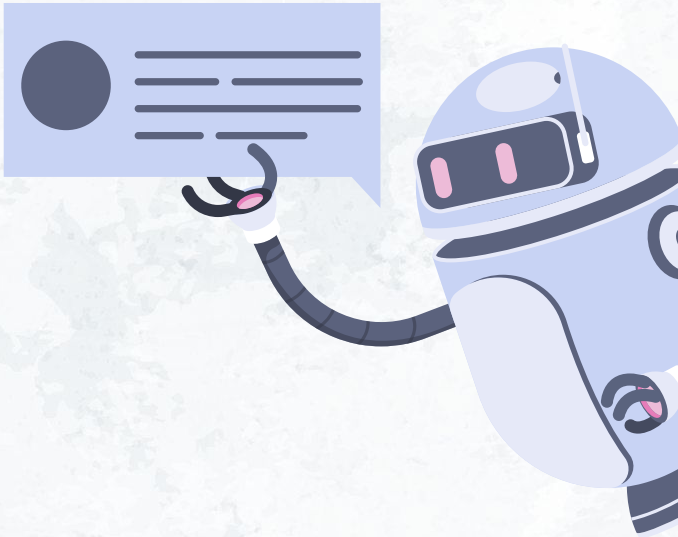
Objectives

Forecast daily and hourly bike-sharing demand using historical data, weather, and city events.

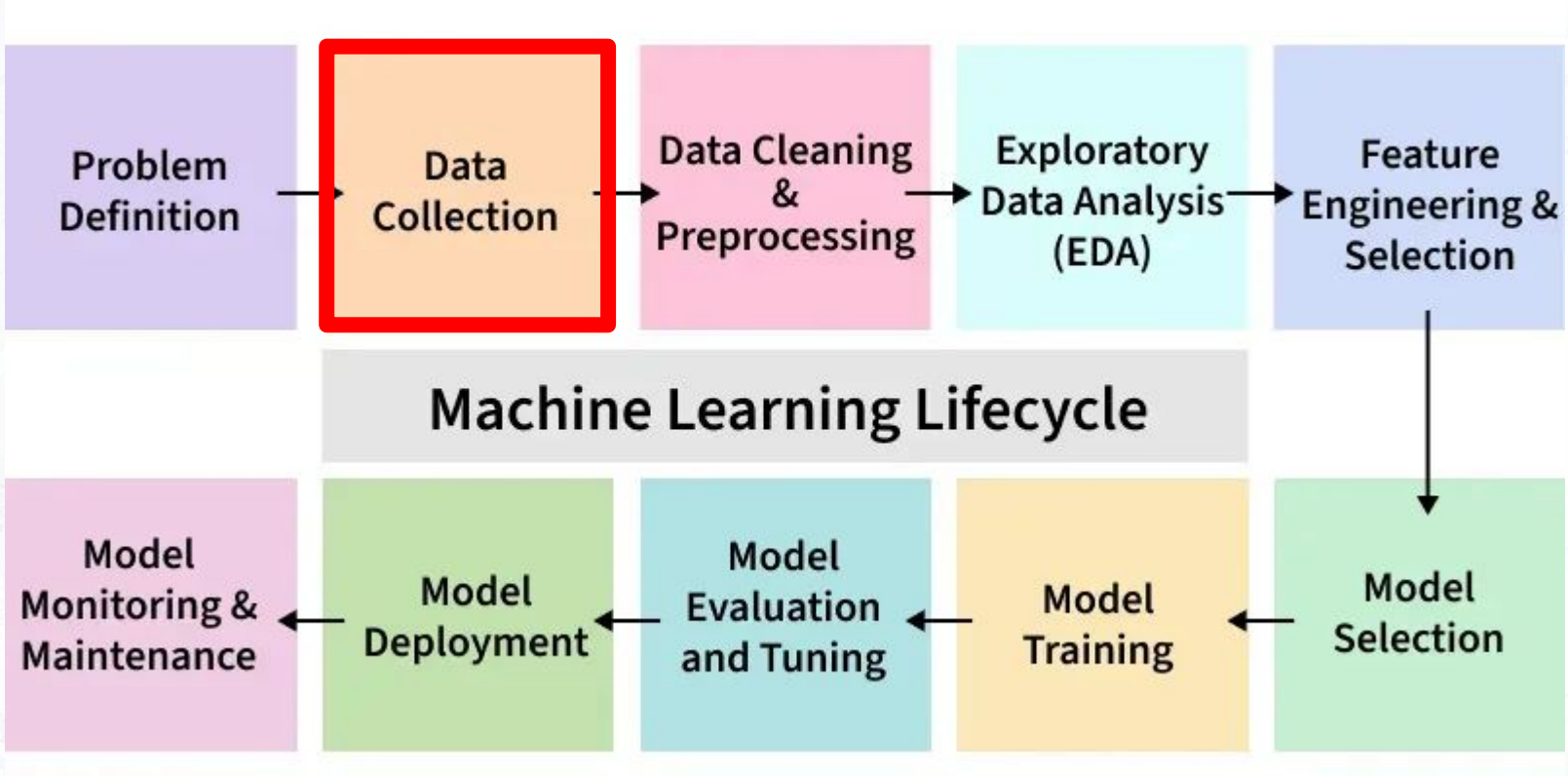
Analyze how factors like weather, events, or location influence bike demand.

Build and compare multiple regression models to find the most accurate predictions.

Provide insights for fleet management, station stocking, and city planning.



Machine Learning Life Cycle



Data Collection

Dataset :https://www.kaggle.com/datasets/lakshmi25npathi/bike-sharing-dataset

The dataset used in this project is the Bike Sharing Dataset from the UCI Machine Learning Repository and Kaggle, containing data from the Capital Bikeshare system in Washington D.C. (2011–2012).

It includes both hourly (~17,000 records) and daily (~730 records) data, covering details such as season, month, hour, weather conditions, temperature, humidity, windspeed, and user counts.

The target variable is the total number of bike rentals (cnt), which combines both casual and registered users.

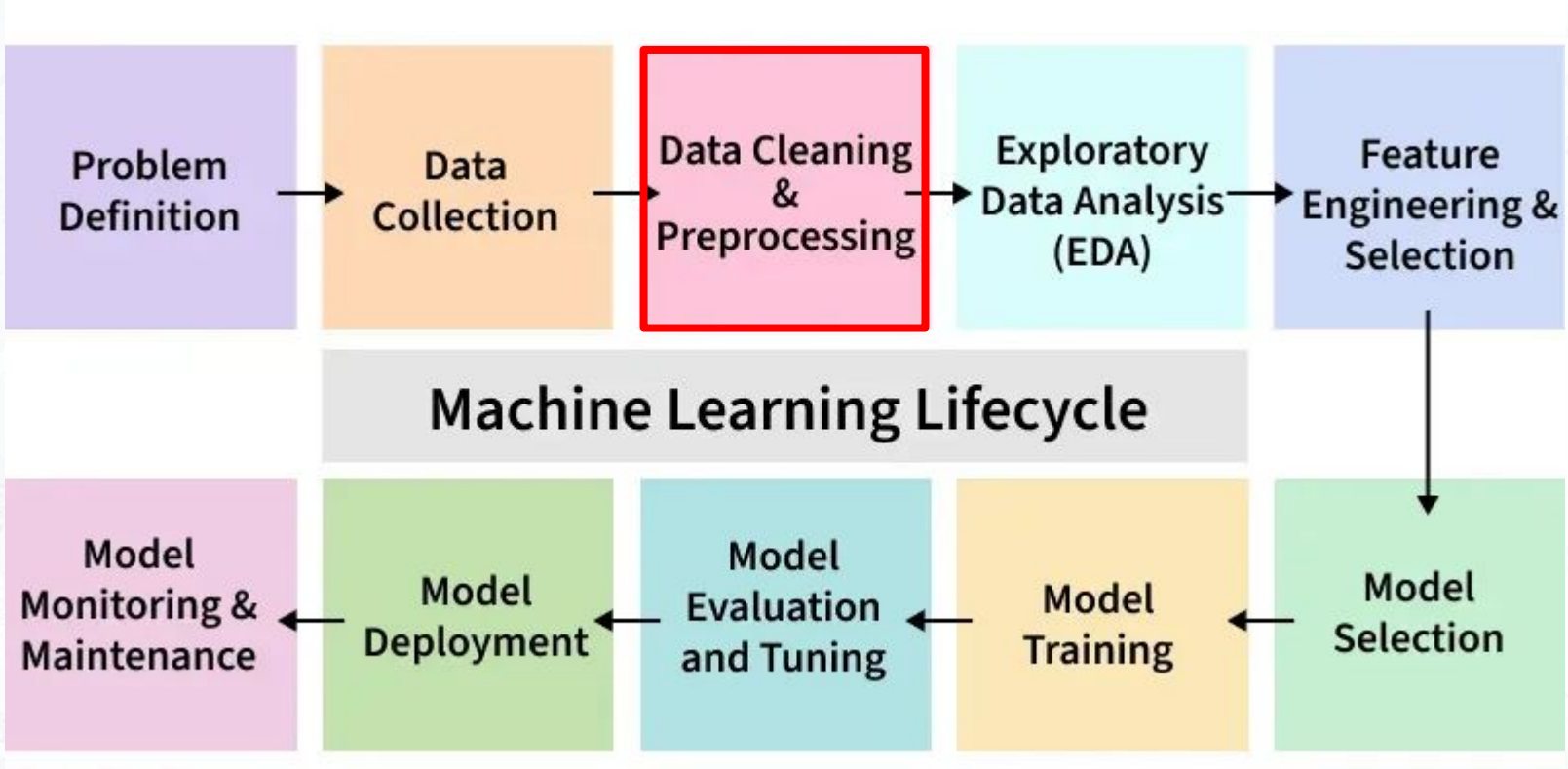
Data Collection

Sample of the hourly dataset showing key features such as hour, temperature, humidity, weather, and bike rental counts (‘count’).

	instant	dteday	season	yr	mnth	hr	holiday	weekday	workingday	weathersit	temp	atemp	hum	windspeed	casual	registered	cnt
0	1	2011-01-01	1	0	1	0	0	6	0	1	0.24	0.2879	0.81	0.0	3	13	16
1	2	2011-01-01	1	0	1	1	0	6	0	1	0.22	0.2727	0.80	0.0	8	32	40
2	3	2011-01-01	1	0	1	2	0	6	0	1	0.22	0.2727	0.80	0.0	5	27	32
3	4	2011-01-01	1	0	1	3	0	6	0	1	0.24	0.2879	0.75	0.0	3	10	13
4	5	2011-01-01	1	0	1	4	0	6	0	1	0.24	0.2879	0.75	0.0	0	1	1

Cnt = target variable

Machine Learning Life Cycle



Data Cleaning

Checked dataset structure:

- Verified columns and data types (dtypes) for **hourly** dataset.

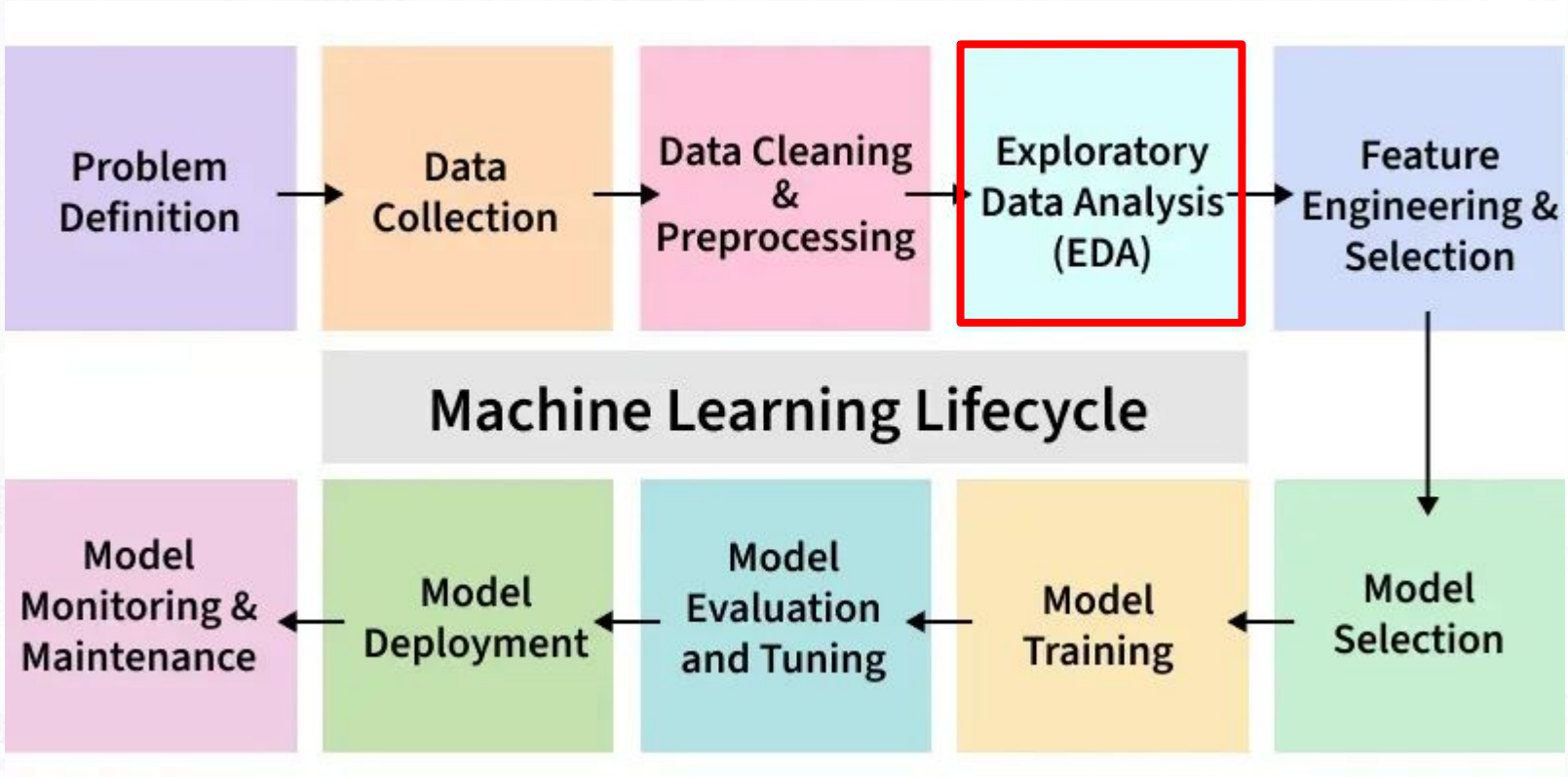
Checked for missing values:

- Confirmed **no nulls** in any column of both datasets.

Checked for duplicates:

- No duplicate records found — data is consistent and ready for analysis.

Machine Learning Life Cycle



Data Visualization

Data visualization, also known as Exploratory Data Analysis (EDA), is used to examine the dataset visually, uncover patterns, and gain insights.

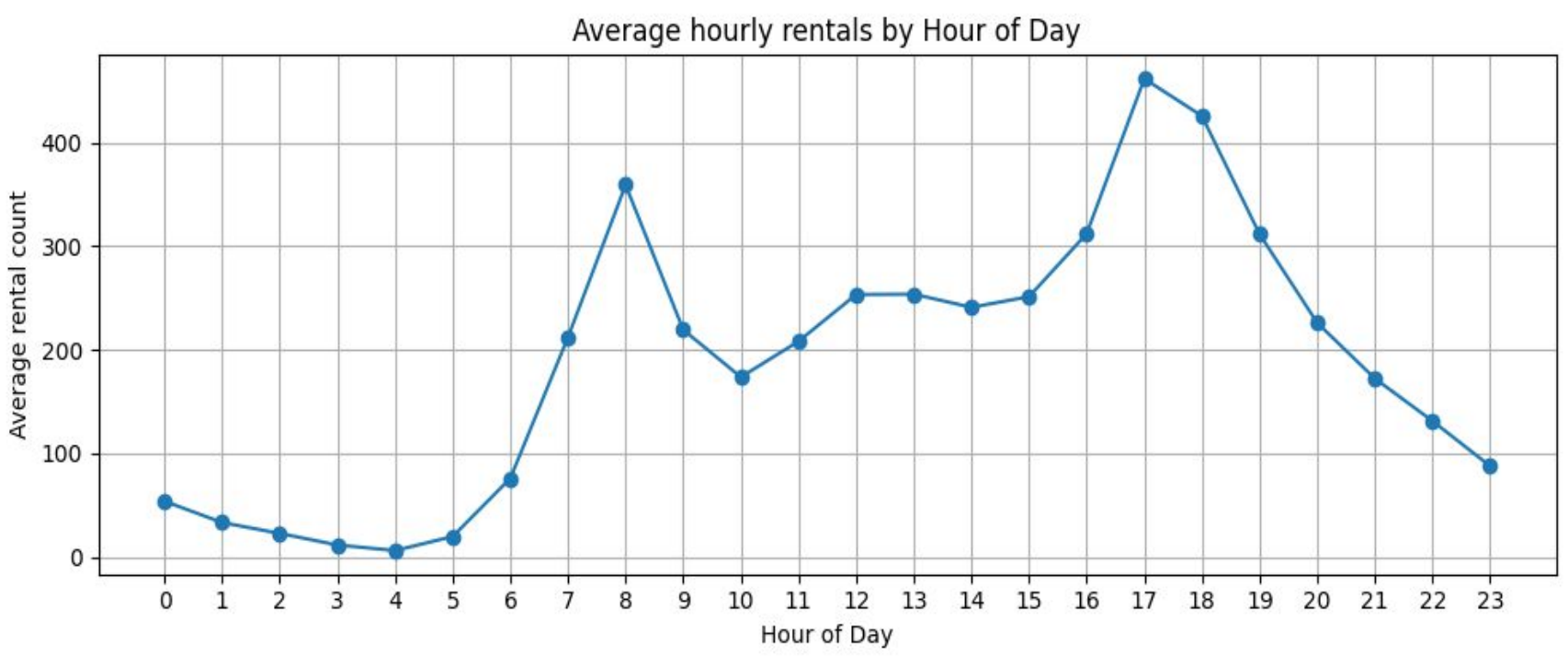
It helps in understanding how features such as time, weather, and season influence bike rentals and guides the creation of meaningful features for machine learning models.

Patterns and relationships between features such as hour, weekday, season, and weather and the number of bike rentals (`count`) were analyzed.

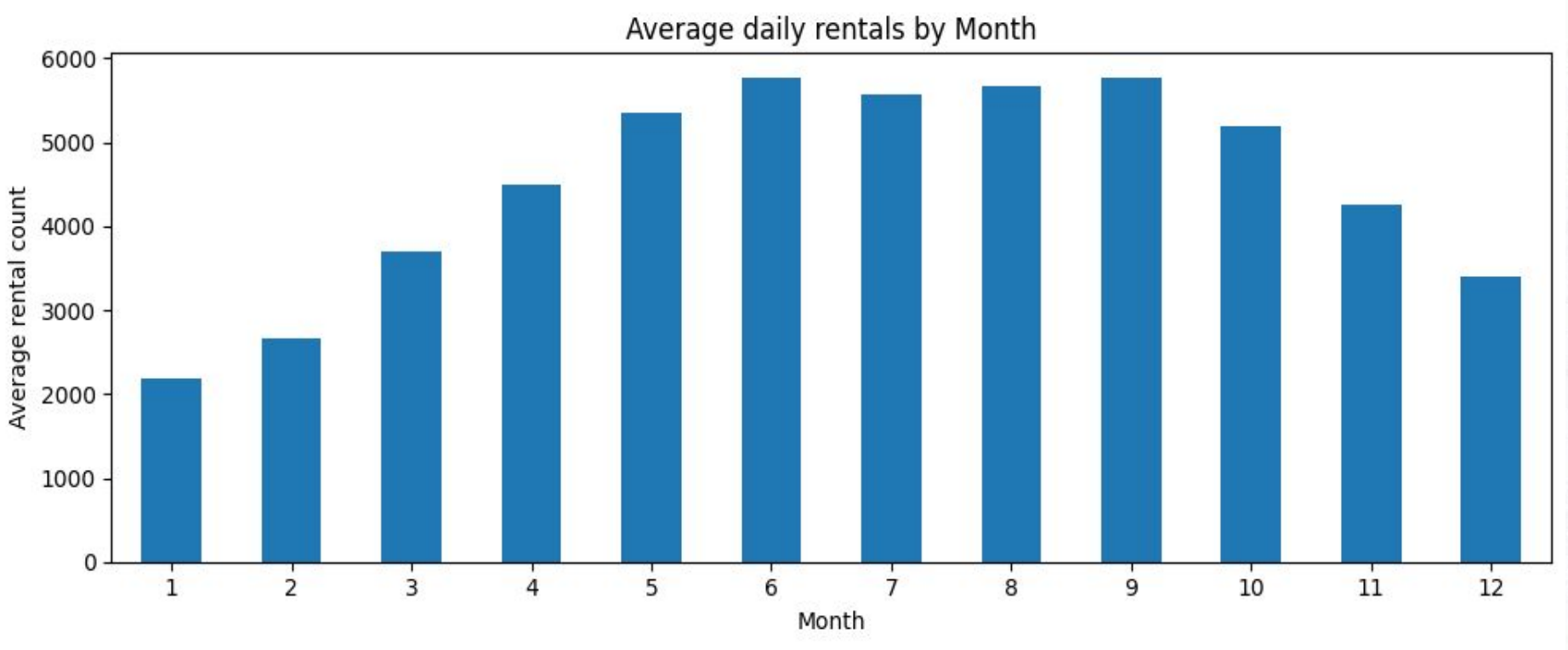
Trends based on time, weather, and season, including peak hours and seasonal variations, were identified.

Outliers or unusual values were checked to ensure they do not affect model performance.

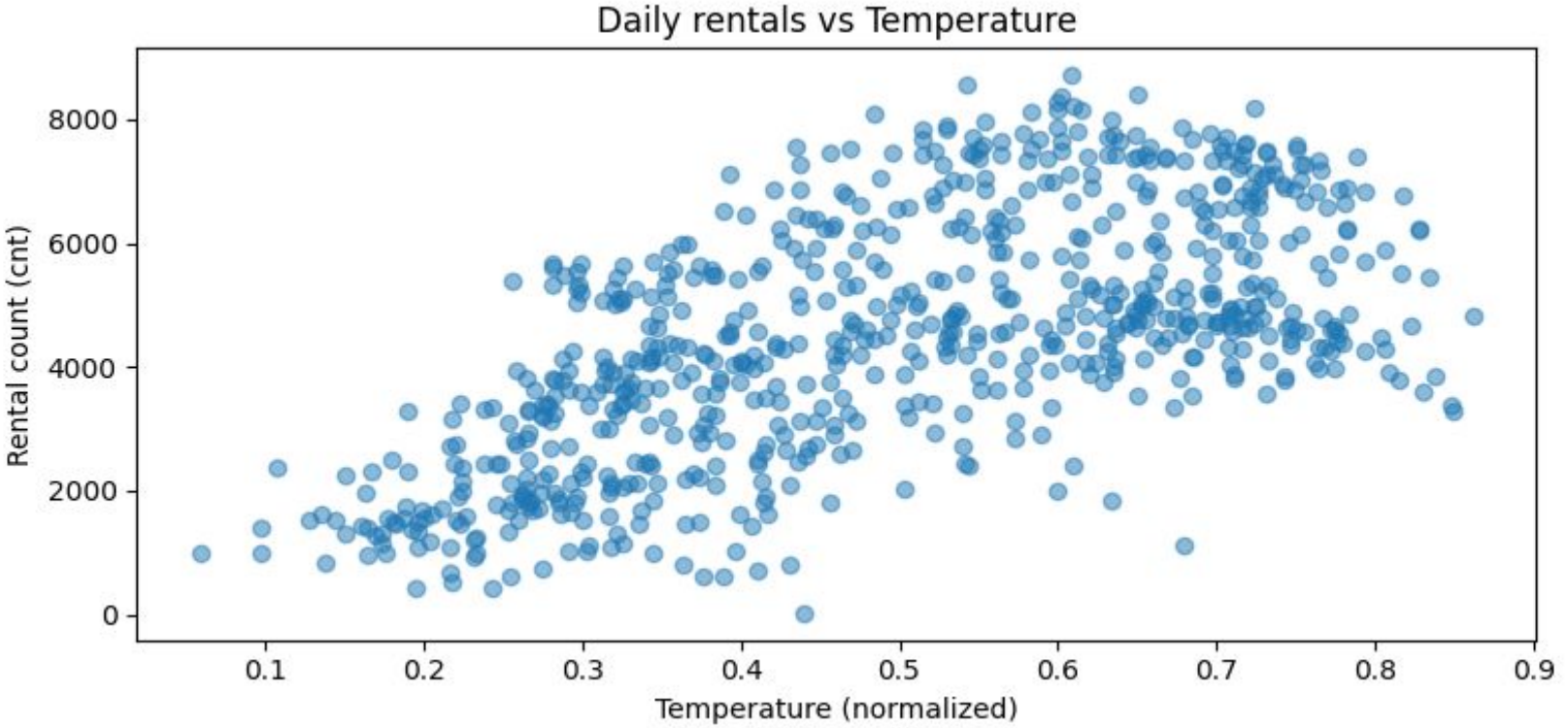
Rentals vs Hour of the Day



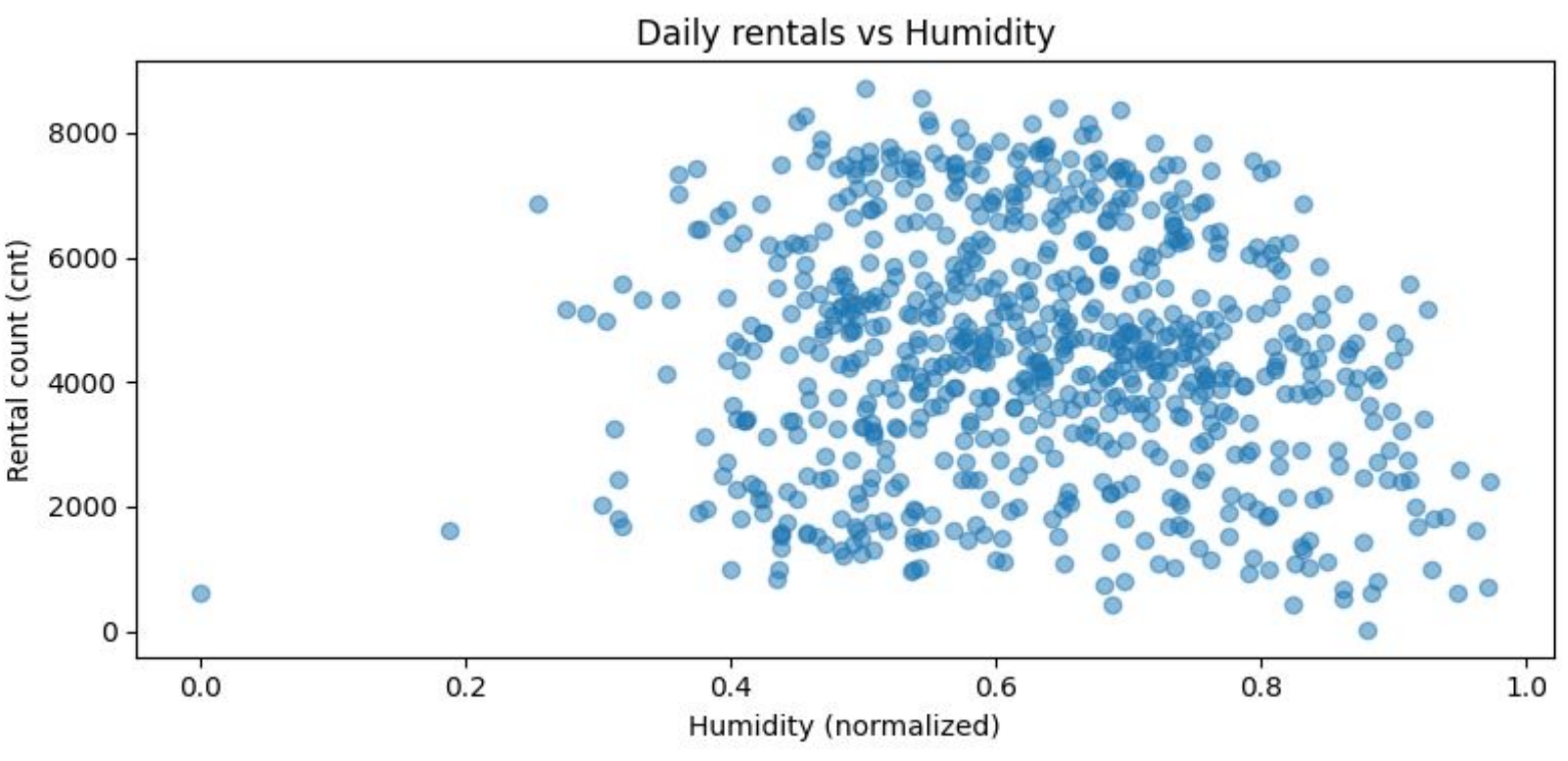
Average Rentals by month



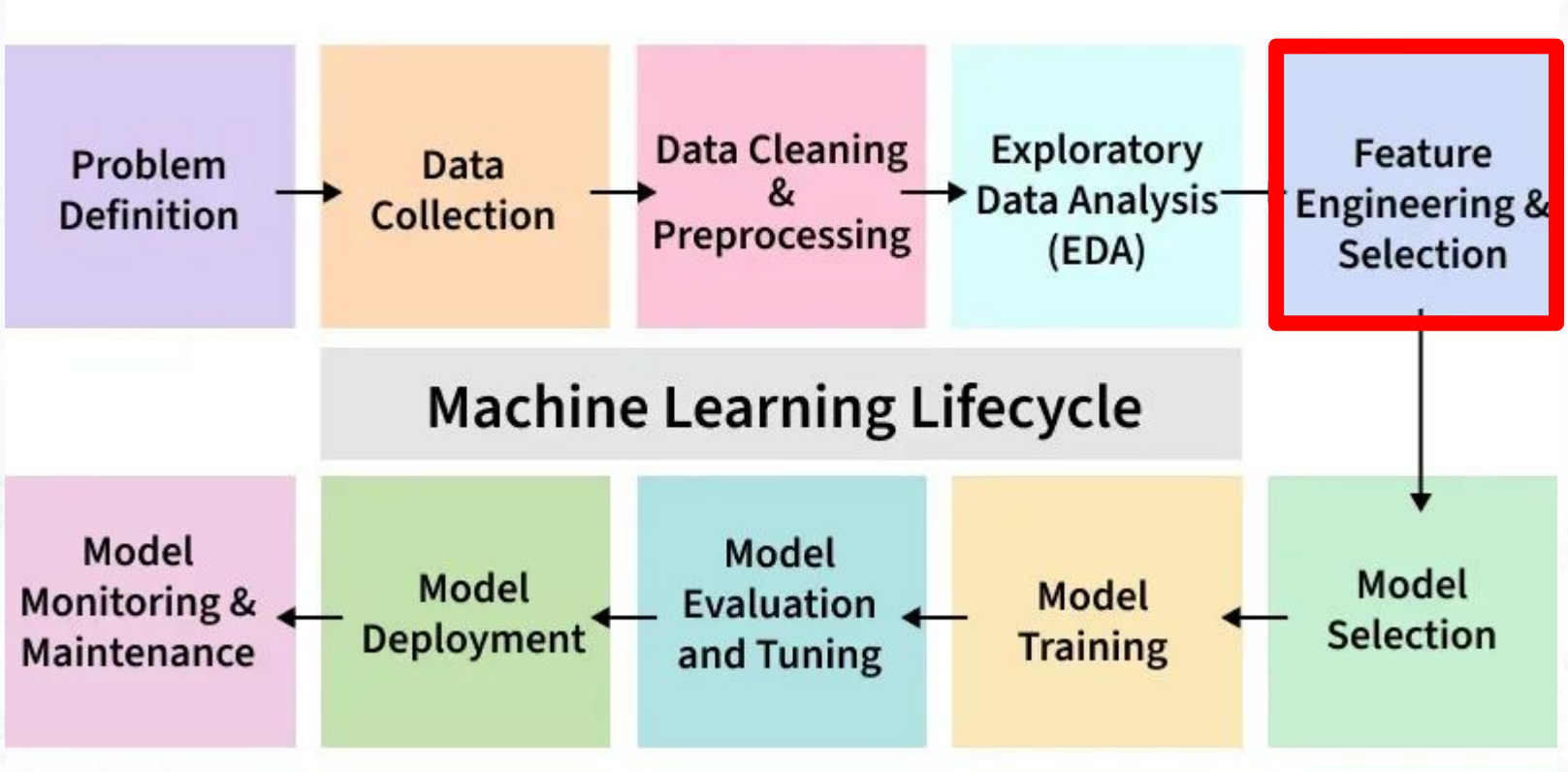
Daily Rentals vs Temperature



Daily Rentals vs Humidity



Machine Learning Life Cycle



Feature Engineering

1. **Time-based features** were extracted from the date and time, including **year, month, weekday, and hour** (for the hourly dataset), to capture temporal patterns.

2. **Cyclic seasonal patterns** were encoded using **sine and cosine transformations** of month, weekday, and hour, enabling the models to understand repeating trends.

3. A **trend feature** was created to represent the number of days (or hours) since the start of the dataset, capturing long-term growth or decline in rentals.

4. **Categorical variables** such as **season, weather situation, holiday, and working day** were **one-hot encoded** for machine learning models.

5. The final datasets include all relevant features for prediction while excluding unnecessary columns like `instant`, `dteday`, `casual`, `registered`, and the target (`cnt` or `count`).

.

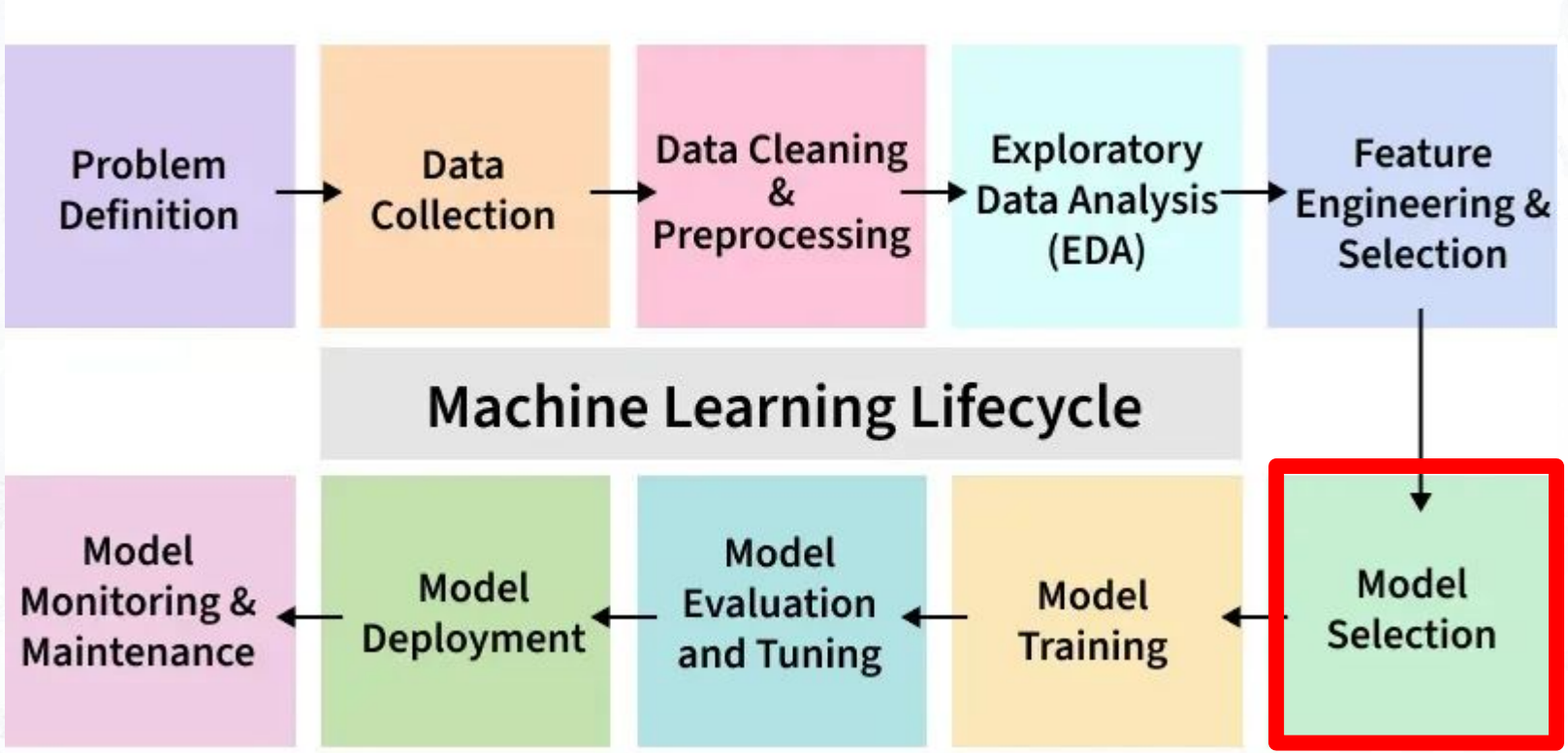
	yr	mnth	holiday	weekday	workingday	temp	atemp	hum	windspeed	\
0	0	1	0	5	0	0.24	0.2879	0.81	0.0000	
1	0	1	0	5	0	0.46	0.4545	0.88	0.2985	
2	0	1	0	5	0	0.40	0.4091	0.94	0.2239	
3	0	1	0	5	0	0.40	0.4091	0.87	0.1940	
4	0	1	0	5	0	0.40	0.4091	0.87	0.2537	

	year	...	cos_weekday	sin_hour	cos_hour	trend	season_2	season_3	\
0	2011	...	-0.222521	0.000000	1.000000	0.000000	False	False	
1	2011	...	-0.222521	-0.258819	0.965926	0.958333	False	False	
2	2011	...	-0.222521	-0.500000	0.866025	0.916667	False	False	
3	2011	...	-0.222521	-0.707107	0.707107	0.875000	False	False	
4	2011	...	-0.222521	-0.866025	0.500000	0.833333	False	False	

	season_4	weathersit_2	weathersit_3	weathersit_4
0	False	False	False	False
1	False	True	False	False
2	False	True	False	False
3	False	True	False	False
4	False	True	False	False

[5 rows x 25 columns]

Machine Learning Life Cycle



Model Selection and Training

To find the most effective model for predicting hourly bike-sharing demand, multiple regression models were evaluated.

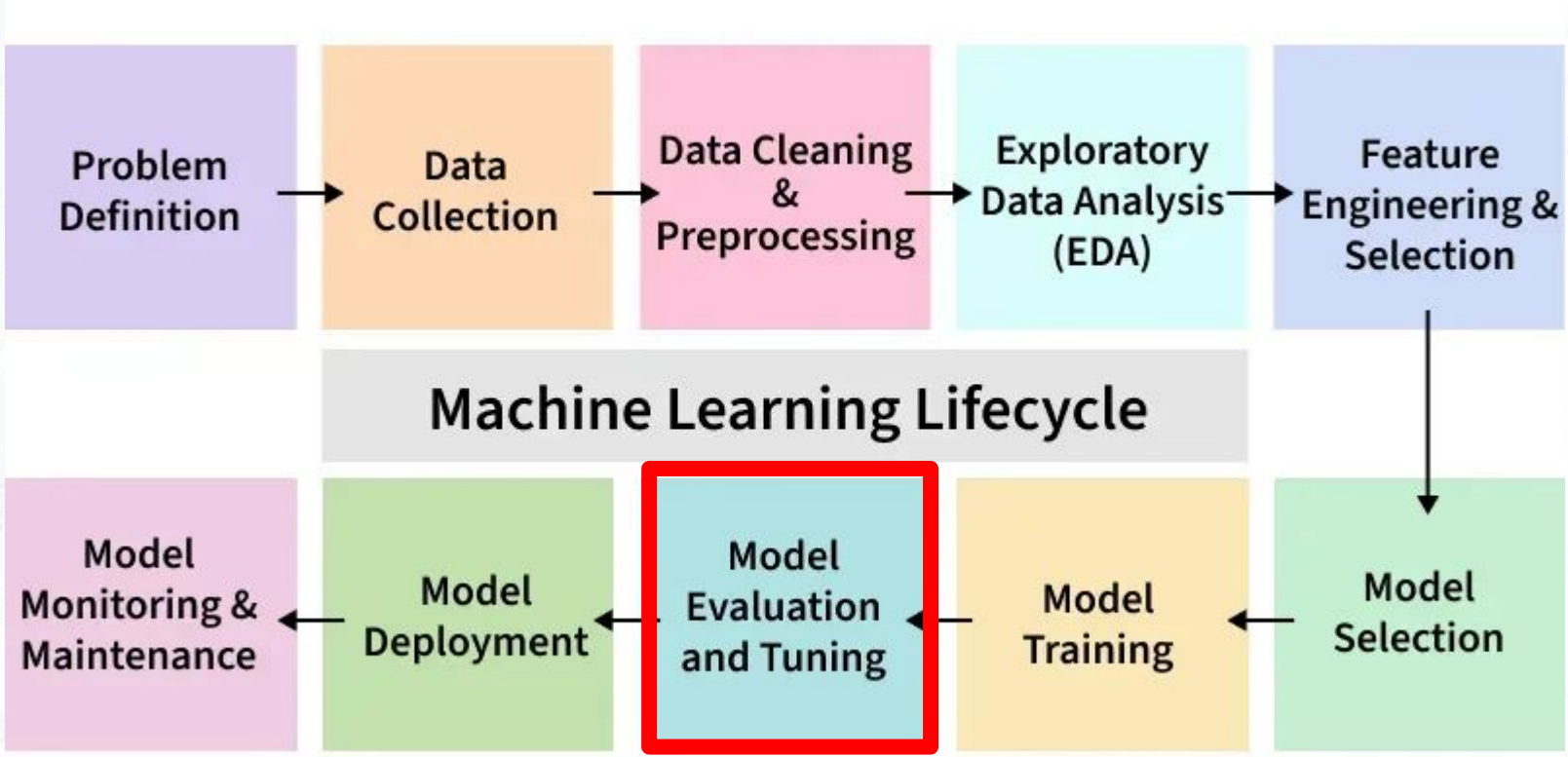
Both **linear** and **ensemble-based** approaches were compared for accuracy, robustness, and generalization.

Models Tested:

- **Linear Regression** – baseline model for simple trend fitting
- **Ridge / Lasso Regression** – to handle multicollinearity
- **Decision Tree Regressor** – captures non-linear relationships
- **Random Forest Regressor** – ensemble averaging for stability
- **Extra Trees Regressor** – faster and less overfitting-prone variant
- **XGBoost Regressor** – optimized gradient boosting for best performance

The dataset was split into **80% Train** and **20% Test** sets to ensure unbiased evaluation.

Machine Learning Life Cycle



Model Evaluation

Mean Absolute Error (MAE):
Measures the **average magnitude of prediction errors**.
Lower MAE indicates that the model's predictions are closer to the actual values.

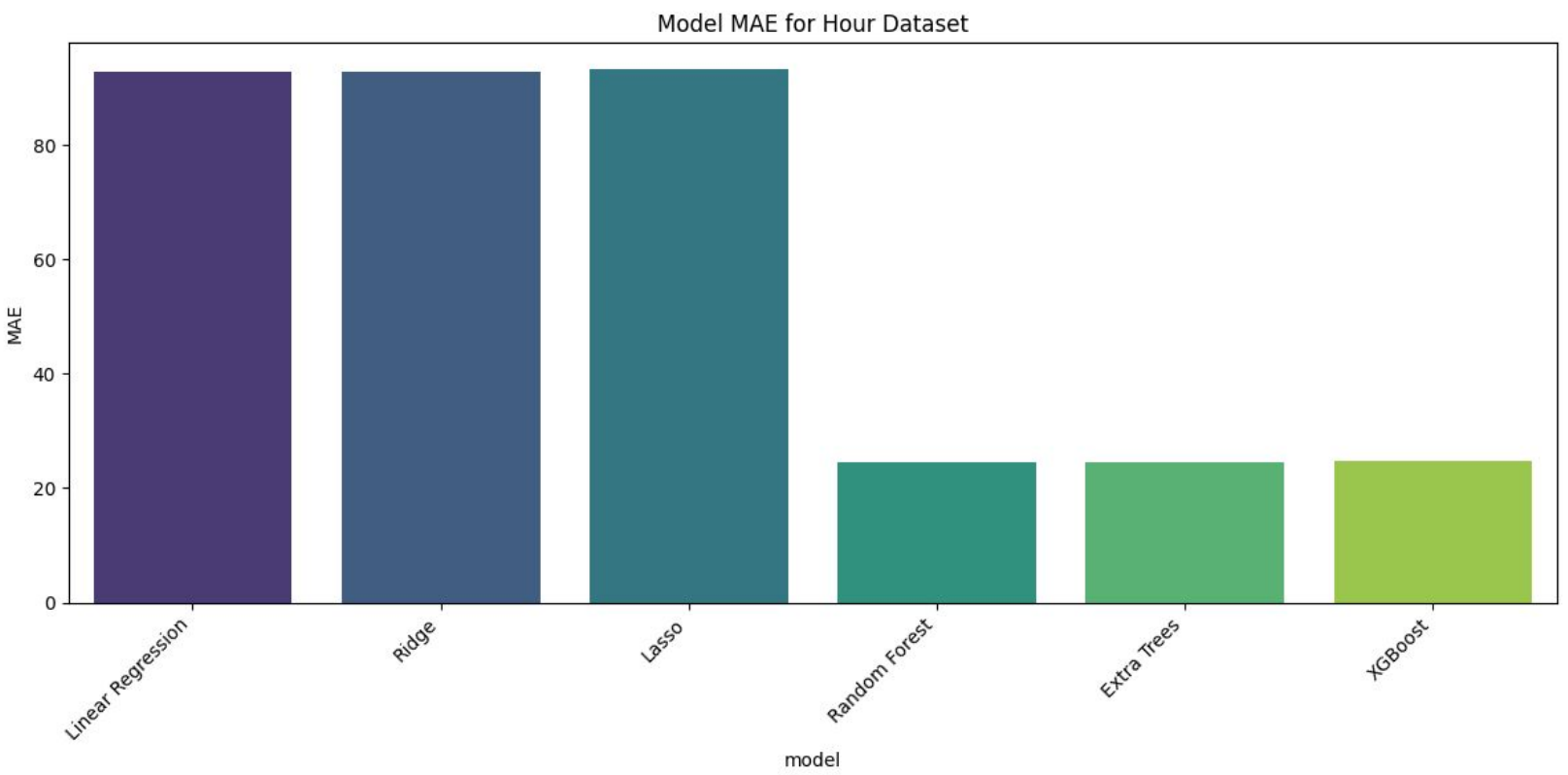
Root Mean Squared Error (RMSE):
Emphasizes **larger errors** more than MAE by squaring them before averaging.
A lower RMSE means the model is making fewer large mistakes.

R^2 Score (Coefficient of Determination):
Represents the **proportion of variance explained by the model**.
Higher R^2 (closer to 1) indicates better predictive accuracy and model fit.

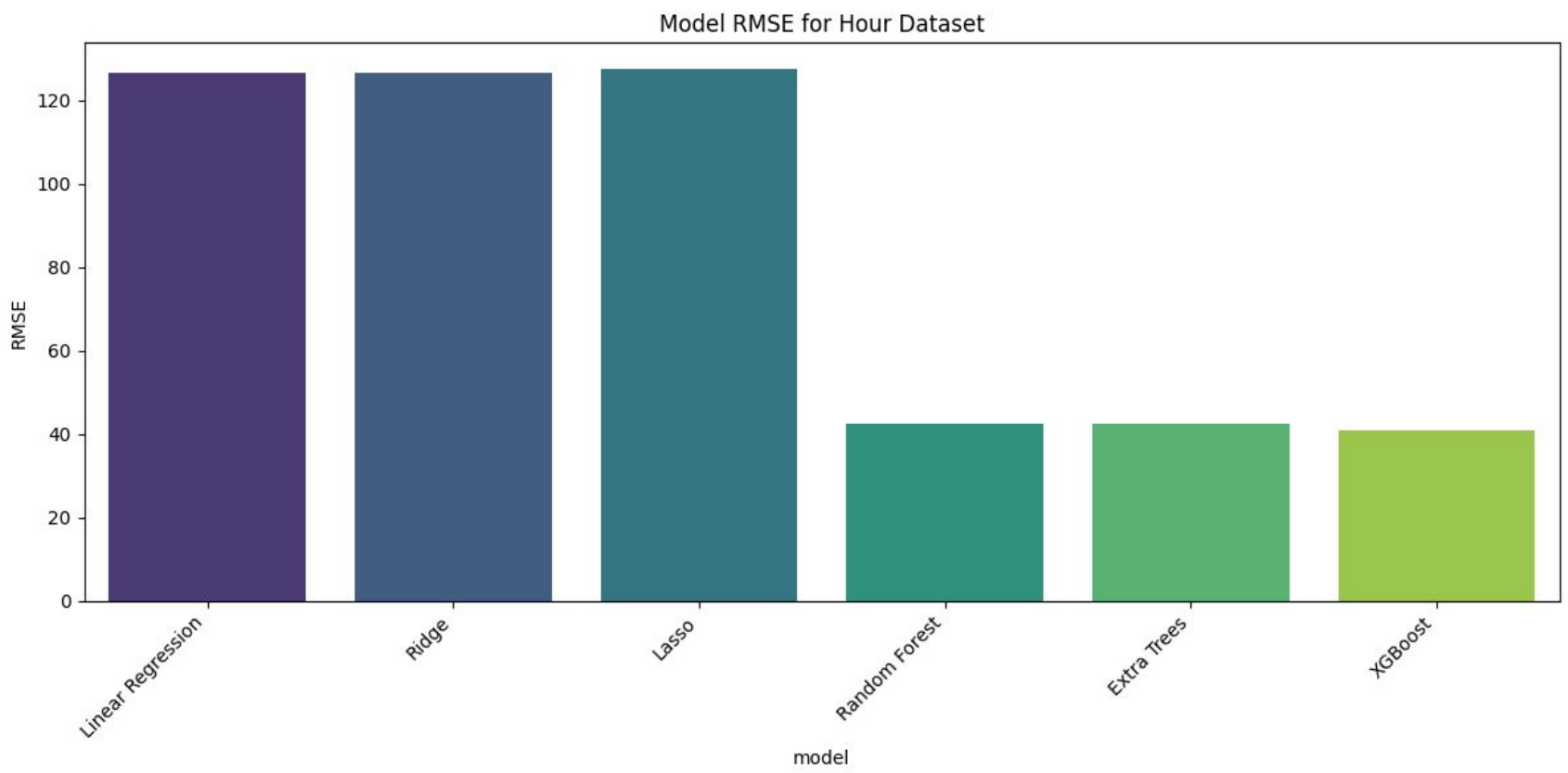
Model Evaluation

Model	MAE	RMSE	R^2
Linear Regression	92.82	126.55	0.5077
Ridge	92.82	126.55	0.5077
Lasso	93.17	127.34	0.5015
Random Forest	24.56	42.41	0.9447
Extra Trees	24.42	42.34	0.9449
XGBoost	24.77	40.67	0.9491

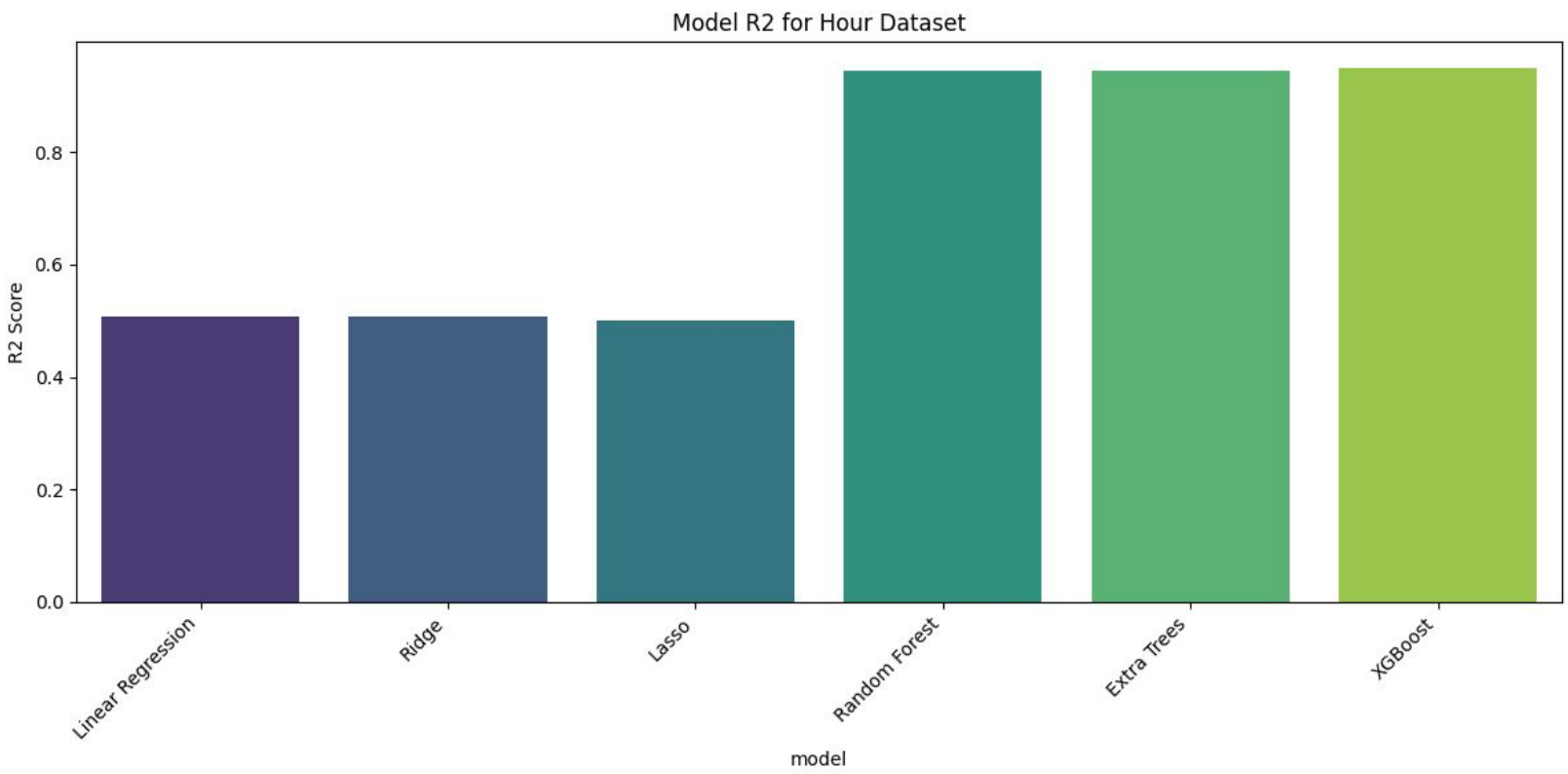
MAE for Hour Dataset



RMSE for Hour Dataset



R^2 for Hour Dataset



Hyper Parameter Tuning

Randomized Search — Broad Parameter Exploration

- Used to **quickly explore a wide range of hyperparameter values** for the XGBoost model.
- Randomly samples a fixed number of combinations instead of testing every possible one.
- Evaluated each combination using **3-fold cross-validation** to ensure stable performance.
- Helps identify **promising regions** in the parameter space efficiently.

Key Hyperparameters Explored:

- n_estimators (number of trees)
- learning_rate (step size in optimization)
- max_depth (tree depth)
- subsample & colsample_bytree (random sampling)
- reg_alpha, reg_lambda (regularization to reduce overfitting)

Hyper Parameter Tuning

Grid Search — Fine-Tuning the Best Parameters

- Performed **after Randomized Search** to focus on the **best-performing parameter region**.
- Tests **every combination** in a smaller, defined grid for precise optimization.
- Uses **3-fold cross-validation** to evaluate each combination reliably.
- Ensures the **most optimal hyperparameters** are selected for final model training.

Key Hyperparameters Tuned:

- n_estimators, learning_rate, max_depth
- subsample, colsample_bytree, gamma
- reg_alpha, reg_lambda

Outcome:

✅ Tuned XGBoost model with **lower MAE and RMSE**, and **higher R^2**
✅ Better generalization to unseen data

Final Model Evaluation

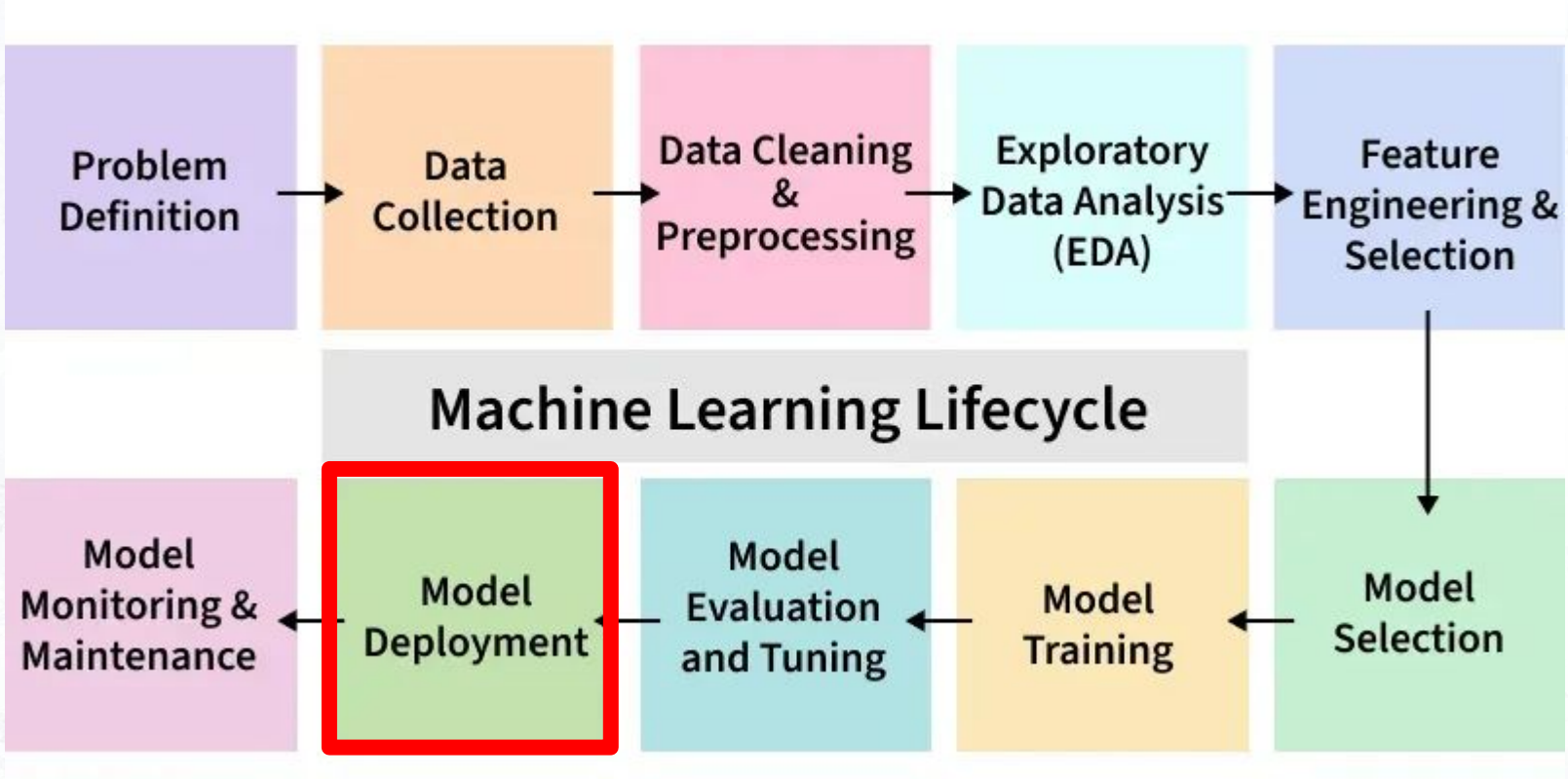
The tuned **XGBoost Regressor** achieved excellent predictive performance on the hourly bike-sharing dataset.

Key evaluation metrics for the final model are:

- **MAE (Mean Absolute Error):** 22.92 — indicates the average deviation between predicted and actual rentals.
- **RMSE (Root Mean Squared Error):** 38.42 — penalizes larger errors more strongly, showing robust predictions.
- **R^2 Score:** 0.9546 — demonstrates the model explains over 95% of variance in hourly bike demand.

These results confirm that the model can accurately forecast hourly bike rentals, supporting operational planning and decision-making.

Machine Learning Life Cycle



Deployment

- Deployed the XGBoost hourly bike-sharing model using a user-friendly Streamlit web app.
- Users can input temporal (hour, weekday, month, season) and weather features to get real-time demand predictions.
- Visualizations include predicted vs. historical average rentals and feature importance charts for better interpretability.
- Enables data-driven operational decisions, such as optimizing bike availability at peak hours.
- Supports scenario analysis by allowing users to simulate different weather and time conditions.

Future Scope

Real-time Updates: Integrate live weather and city event data to continuously improve predictions.

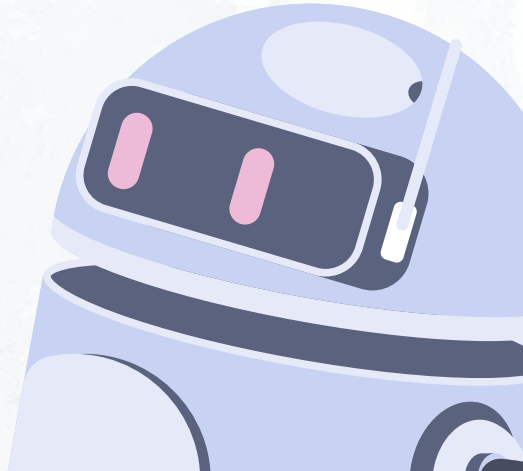
Model Retraining: Periodically retrain the model with new data to maintain accuracy over time.

Enhanced Features: Include additional factors such as traffic, local events, for better demand forecasting.

Multi-city Expansion: Adapt the model for other cities with similar bike-sharing systems.

User Feedback Loop: Incorporate user or operator feedback to refine predictions and improve usability.

Thank You



Sai Raghav Telugu
B.Tech CSE – Final Year