A

REAL TIME RESEARCH PROJECT REPORT ON

SUBWAY PASSENGER FLOW FORECASTING USING MACHINE LEARNING

Submitted in partial fulfilment for the award of the degree of
BACHELOR OF TECHNOLOGY

In
COMPUTER SCIENCE AND ENGINEERING ARTIFICIAL INTELLIGENCE AND MACHINE LEARNING
By

GAJULA NIKHIL	**:**	**22Q91A6685**
HARSH R BAGTHARIA	**:**	**22Q91A6691**
MADHARAPU NITHIN KUMAR	**:**	**22Q91A6697**
NANGNURI NIKHITHA	**:**	**22Q91A66A2**

Under the guidance of

DR. K. MADAN MOHAN
Assistant Professor, Dept. of CSE(AIML)

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING ARTIFICIAL INTELLIGENCE AND MACHINE LEARNING
MALLA REDDY COLLEGE OF ENGINEERING

(Approved by AICTE-Permanently Affiliated to JNTU-Hyderabad) Accredited by NBA & NAAC, Recognized section 2(f) & 12(B) of UGC New Delhi ISO 9001:2015 certified Institution
Maisammaguda, Dhulapally (Post via Kompally), Secunderabad500100

2023-2024

MALLA REDDY COLLEGE OF ENGINEERING

(Approved by AICTE-Permanently Affiliated to JNTU-Hyderabad)

Accredited by NBA & NAAC, Recognized section 2(f) & 12(B) of UGC New Delhi ISO 9001:2015 certified Institution

Maisammaguda, Dhulapally (Post via Kompally), Secunderabad500100

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING ARTIFICIAL INTELLIGENCE AND MACHINE LEARNING

CERTIFICATE

This is to certify that the Real Tile Research Project report on "SUBWAY PASSENGER FLOW FORECASTING USING MACHINE LEARNING" is successfully done by the following students of Department of Computer Science & Engineering (Artificial Intelligence and Machine Learning) of our college in partial fulfilment of the requirement for the award of B. Tech degree in the year 2023-2024. The results embodied in this report have not been submitted to any other University for the award of any diploma or degree.

GAJULA NIKHIL	**: 22Q91A6685**
HARSH R BAGTHARIA	**: 22Q91A6691**
MADHARAPU NITHIN KUMAR	**: 22Q91A6697**
NANGNURI NIKHITHA	**: 22Q91A66A2**

Submitted for the viva-voce examination held on: _____

INTERNAL GUIDE
Mr. M. Sakthivel
Assistant Professor

HOD
Mr. Vignesh Janarthanan
Assistant Professor

Internal Examiner

External Examiner

DECLARATION

We, Gajula Nikhil, Harsh R Bagtharia, Madharapu Nithin Kumar, R Nangnuri Nikhitha with Regd.no. 22Q91A6685, 22Q91A6691, 22Q91A6697, 22Q91A66A2 are hereby declaring that the real time research project report entitled **"SUBWAY PASSENGER FLOW FORECASTING USING MACHINE LEARNING"** has done by us under the guidance of **Dr. K Madan Mohan,** Assistant Professor, Department of Computer Science Engineering (Artificial Intelligence and Machine Learning) is submitted in the partial fulfilment of the requirements for the award of degree of **BACHELOR OF TECHNOLOGY** in **COMPUTER SCIENCE AND ENGINEERING ARTIFICIAL INTELLIGENCE AND MACHINE LEARNING.**

The Results embedded in this project report have not been submitted to any other University or institute for the award of any degree or diploma.

Signature of the Candidate

GAJULA NIKHIL	**: 22Q91A6685**
HARSH R BAGTHARIA	**: 22Q91A6691**
MADHARAPU NITHIN KUMAR	**: 22Q91A6697**
NANGNURI NIKHITHA	**: 22Q91A66A2**

DATE:

PLACE: Maisammaguda

ACKNOWLEDGEMENT

First and foremost, we would like to express our immense gratitude towards our institution Malla Reddy College of Engineering, which helped us to attain profound technical skills in the field of Computer Science & Engineering, there by fulfilling our most cherished goal.

We are pleased to thank **Sri Ch. Malla Reddy**, our Founder, Chairman **MRGI**, **Sri Ch. Mahender Reddy**, Secretary, **MRGI** for providing this opportunity and support throughout the course.

It gives us immense pleasure to acknowledge the perennial inspiration of **Mr. Vignesh Janarthanan** HOD, **Mr. C. Dinesh** Project Coordinator of Computer Science Engineering (Artificial Intelligence and Machine Learning) Department for their inspiration adroit guidance and constructive criticism for successful completion of our degree.

We would like to thank **Dr. K. Madan Mohan** Assistant Professor our internal guide, for his valuable suggestions and guidance during the exhibition and completion of this project.

Finally, we avail this opportunity to express our deep gratitude to all staff who have contribute their valuable assistance and support making our project success.

GAJULA NIKHIL	**: 22Q91A6685**
HARSH R BAGTHARIA	**: 22Q91A6691**
MADHARAPU NITHIN KUMAR	**: 22Q91A6697**
NANGNURI NIKHITHA	**: 22Q91A66A2**

PROPOSED SYSTEM

The proposed subway passenger flow forecasting system aims to leverage advanced machine learning algorithms and modern data processing techniques to overcome the limitations of the existing system. By using Scikit-learn and XGBoost, the system can train robust predictive models on comprehensive datasets of historical passenger flow data. This approach allows for the integration of multiple variables such as station-specific attributes, time of day, day of the week, and external factors, enhancing the accuracy and granularity of predictions. Data preprocessing tasks, handled efficiently with Pandas, ensure data quality and relevance for model training.

The deployment of trained AI models within a Django web application environment, facilitated by Joblib, enables real-time predictions based on user inputs. This provides commuters and transit authorities with immediate access to forecasted passenger counts, visualized through interactive charts and graphs using Chart.js. These visualizations offer insights into hourly and daily passenger flow trends, empowering stakeholders to make informed decisions regarding resource allocation, service planning, and operational adjustments. Real-time data integration further enhances the system's adaptability to changing transit conditions, ensuring that forecasts remain responsive and accurate in dynamic urban environments.

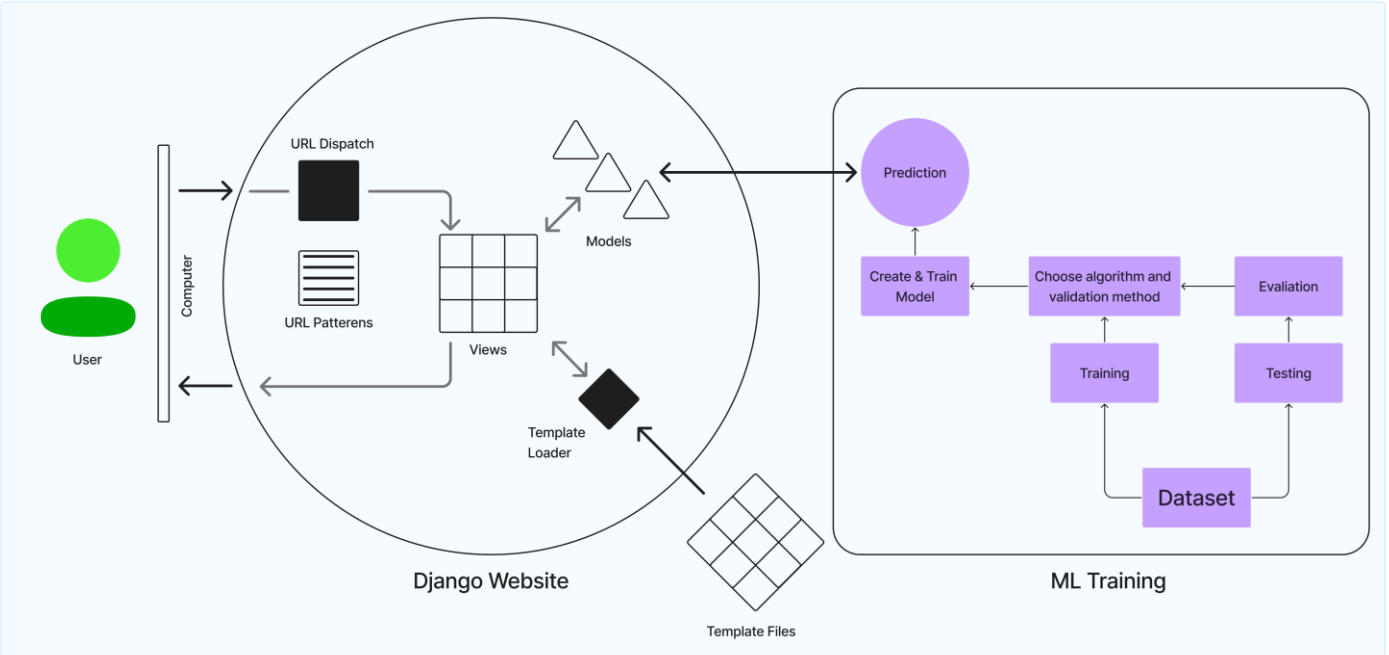

Figure 2: Proposed System

Key Features of the Proposed System

1. Advanced Machine Learning Model:

Dynamic Prediction: The system uses a pre-trained machine learning model capable of adapting to new data and trends, providing more accurate and up-to-date predictions.

Complex Data Integration: By incorporating diverse datasets including historical passenger counts, weather conditions, event schedules, and temporal factors, the model can capture complex interactions and patterns in passenger flow.

SOFTWARE ARCHITECTURE OVERVIEW

The architecture diagram provided illustrates the interaction and data flow between different components in a subway passenger flow forcasting system. Below is a detailed explanation, similar to what would be found in a technical documentation:

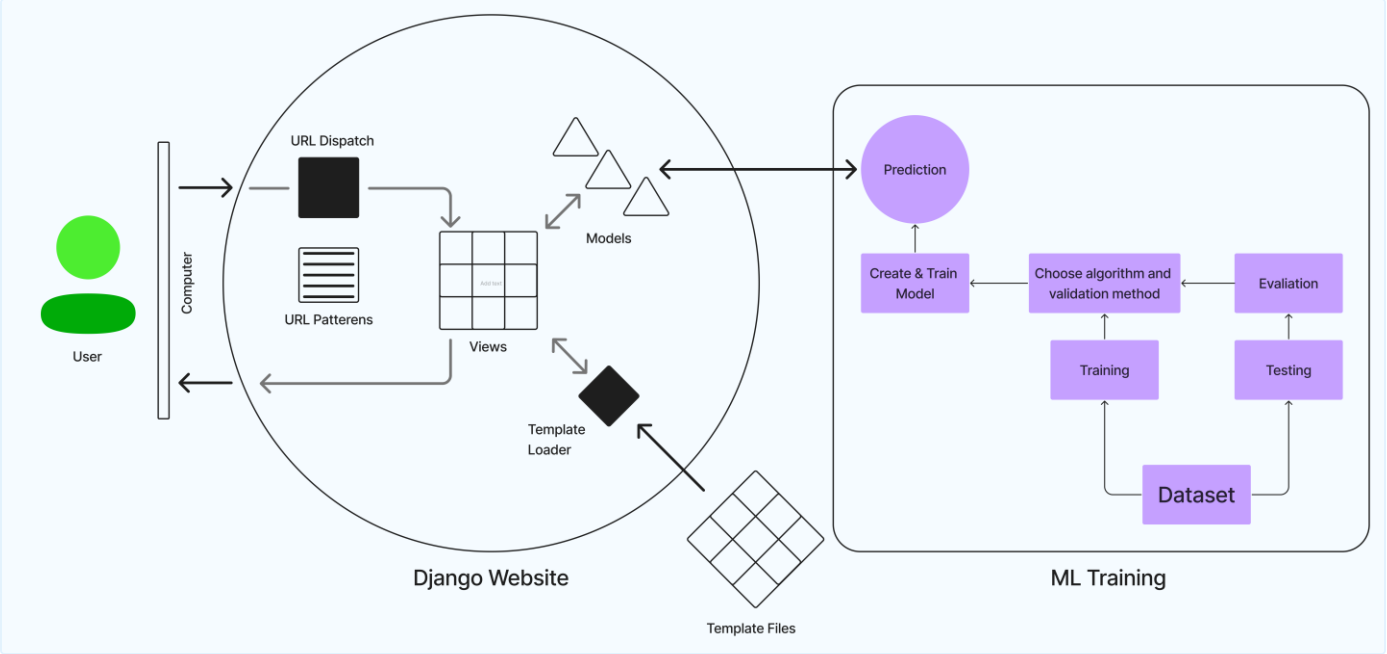

Figure 3: Software Architecture

Components of the System:

1. **Models**:
   - **TrafficPrediction Model**: Defines the structure for storing subway passenger flow predictions. It includes fields for stop name, line number, hour, day, and predicted counts for entering and leaving passengers.

```
class TrafficPrediction(models.Model):
    stop_name = models.CharField(max_length=100)
    line = models.IntegerField()
    hour = models.IntegerField()
    day = models.IntegerField()
    enter_count = models.FloatField()
    leave_count = models.FloatField()

    def __str__(self):
        return f"{self.stop_name} {self.hour}:{self.day}"
```

2. **Views**:
   - **predict_traffic View**: Handles user requests for predicting subway passenger flows. It loads a pre-trained machine learning model, processes user input, performs predictions, and stores results in the database.

```
def predict_traffic(request):
    if request.method == 'POST':
        # Process user input, load model, perform predictions, and
store results
        ...
        return render(request, 'traffic/prediction_result.html',
{'prediction': prediction, 'week_results':week_results,
'day_results': day_results})
    return render(request, 'traffic/predict_traffic.html')
```

3. **Templates**:
   - **predict_traffic.html**: Provides a form for users to input details such as stop name, line, hour, and day.
   - **prediction_result.html**: Displays prediction results using bar graphs and pie charts generated with Chart.js to visualize passenger flows.
4. **Machine Learning Model**:
   - **sklearn and XGBoost**: Used for training and deploying the machine learning model. Predictions are made based on input data (stop name, line, hour, day) provided by users via the Django application.

This software architecture overview highlights the structured approach of using Django's MVT pattern along with machine learning integration and visualization techniques to deliver a robust and user-friendly subway passenger flow forecasting system.

Class Diagram

Class Diagram Overview: The class diagram depicts the static structure of the system, showing classes, their attributes, methods, and relationships. Here's an overview based on your project:

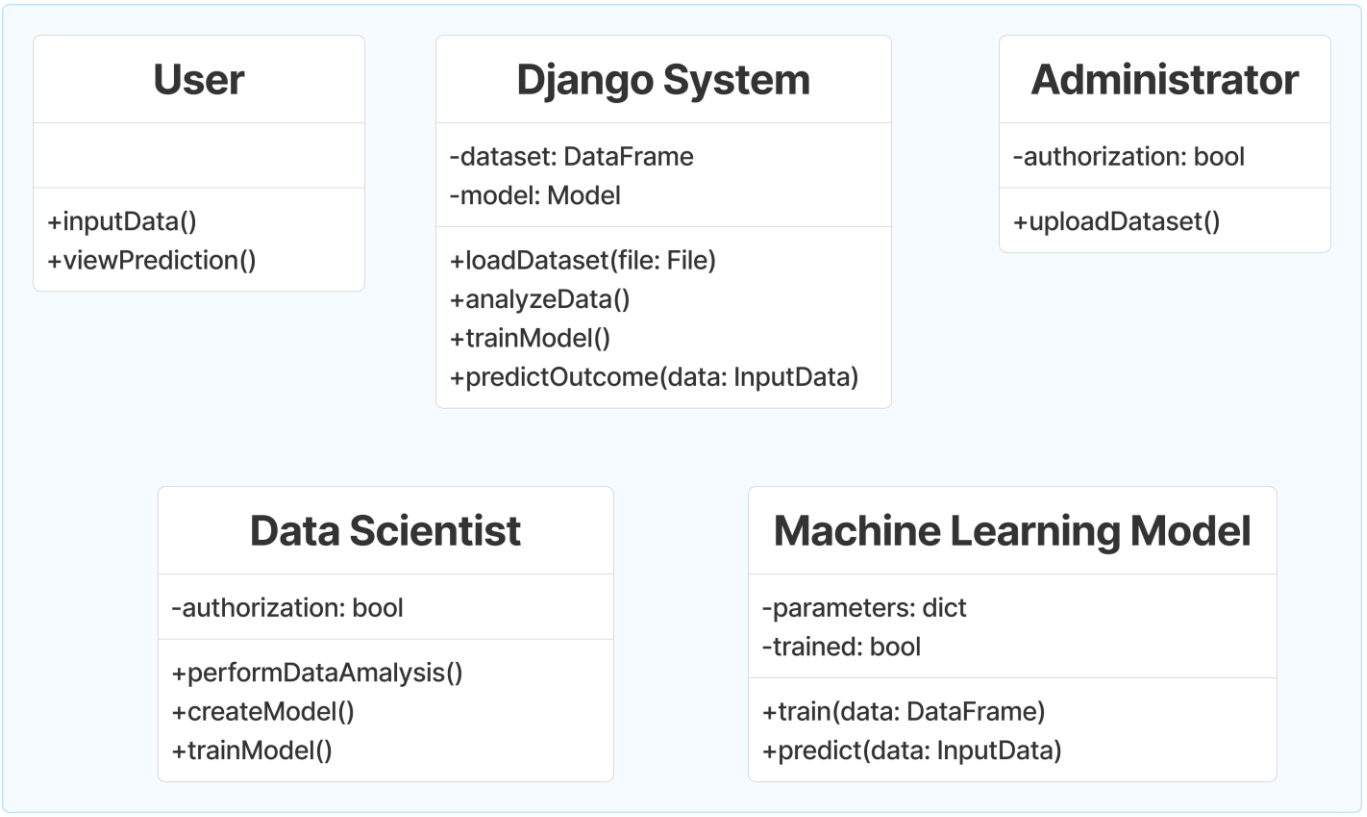

Figure 4: Class Diagram

- **Entities:**
  - **User**: Represents individuals interacting with the system.
  - **Administrator**: Manages system administration tasks such as dataset management.
  - **Data Scientist**: Responsible for developing and managing machine learning models.
- **Attributes and Methods:**
  - **Dataset**: Represents the dataset used for analysis and training.
  - **Model**: Represents the machine learning model used for prediction.
  - **InputData**: A class or structure defining the format of input data for predictions.
  - **ViewPrediction()**: Method allowing users to view prediction results.
  - **UploadDataset()**: Method enabling administrators to upload datasets into the system.
  - **LoadDataset(file)**: Method for loading datasets into the system for analysis and training.
  - **TrainModel()**: Method for training the machine learning model using the dataset.
  - **PredictOutcome(data)**: Method for predicting outcomes based on input data.

Sequence Diagram

Sequence Diagram Overview: A sequence diagram illustrates the interactions between objects in a sequential order, showing the flow of messages and actions between different components of the system.

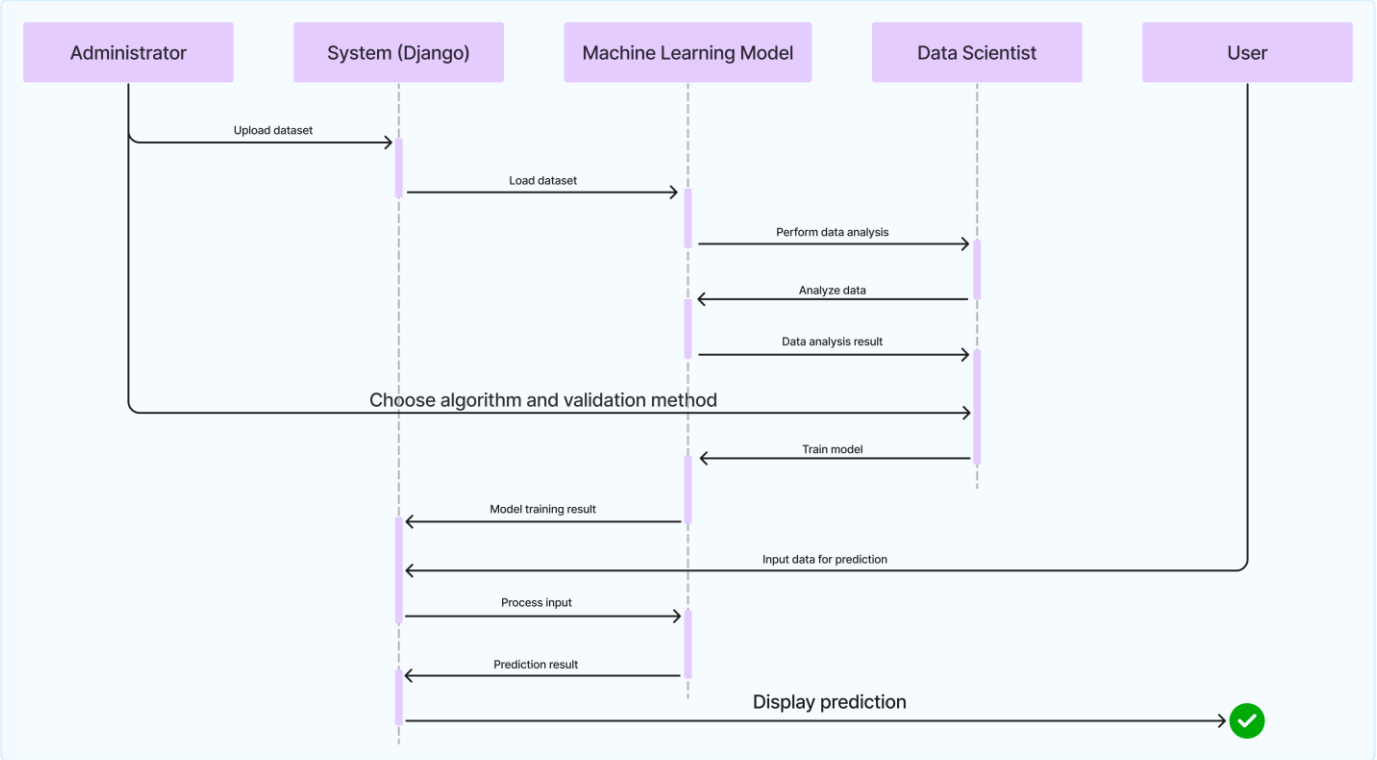

Figure 5: Sequence Diagram

- **Administrator** uploads a dataset.
- **System** loads and analyzes the dataset.
- **Data Scientist** performs data analysis.
- **Data Scientist** chooses algorithms and validation methods.
- **Data Scientist** trains the model.
- **User** inputs data for prediction.
- **Machine Learning Model** processes the input and provides prediction results.

Activity Diagram

Activity Diagram Overview: An activity diagram shows the flow of activities or actions within the system, depicting the sequence and conditions of activities.

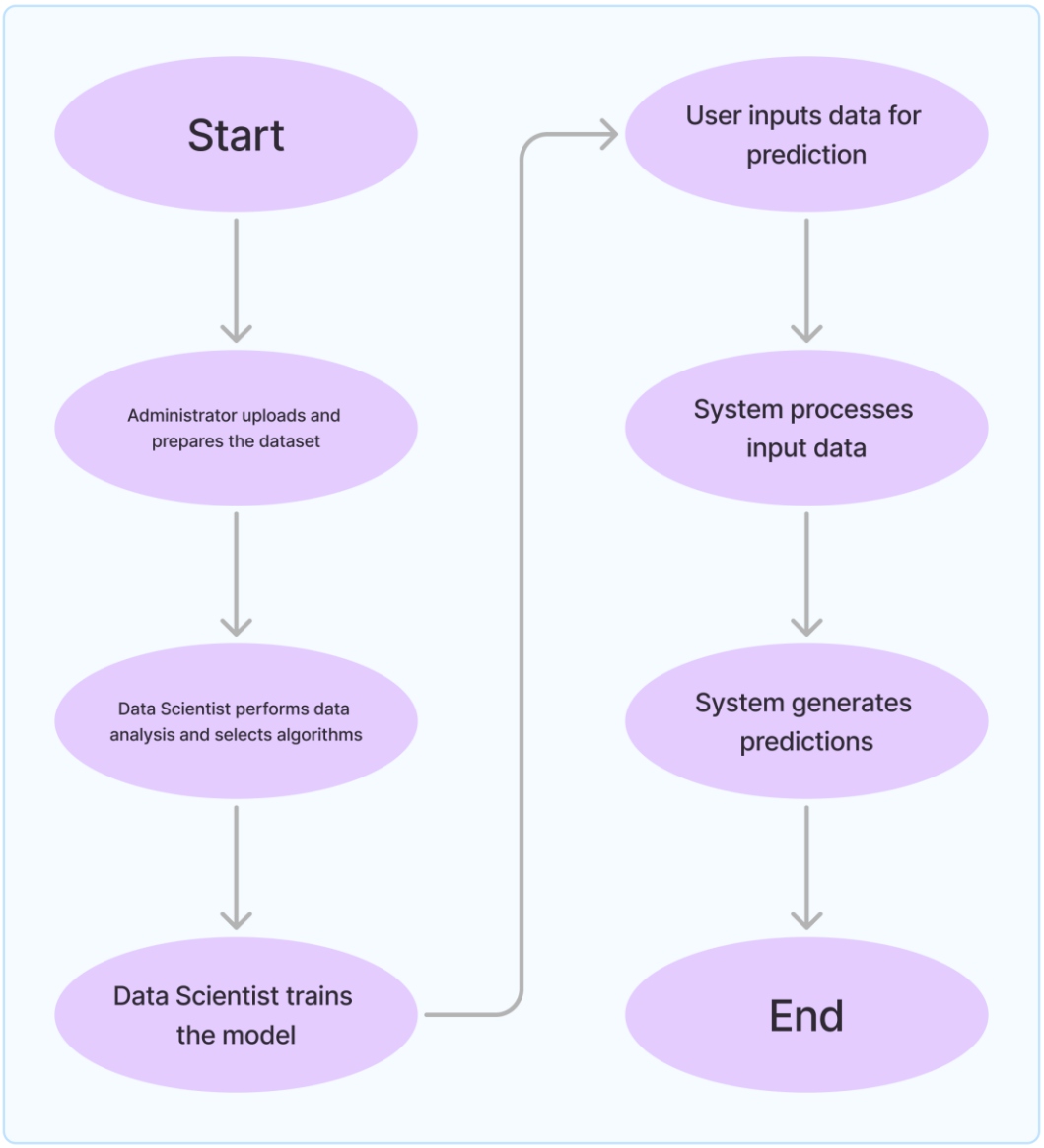

Figure 6: Activity Diagram

- **Administrator** uploads and prepares the dataset.
- **Data Scientist** performs data analysis and selects algorithms.
- **Data Scientist** trains the model.
- **User** inputs data for prediction.
- **System** processes input data and generates predictions.

Use Case Diagram

Use Case Diagram Overview: A use case diagram identifies the interactions between actors (users) and the system, showing how users interact with the system to achieve specific goals.

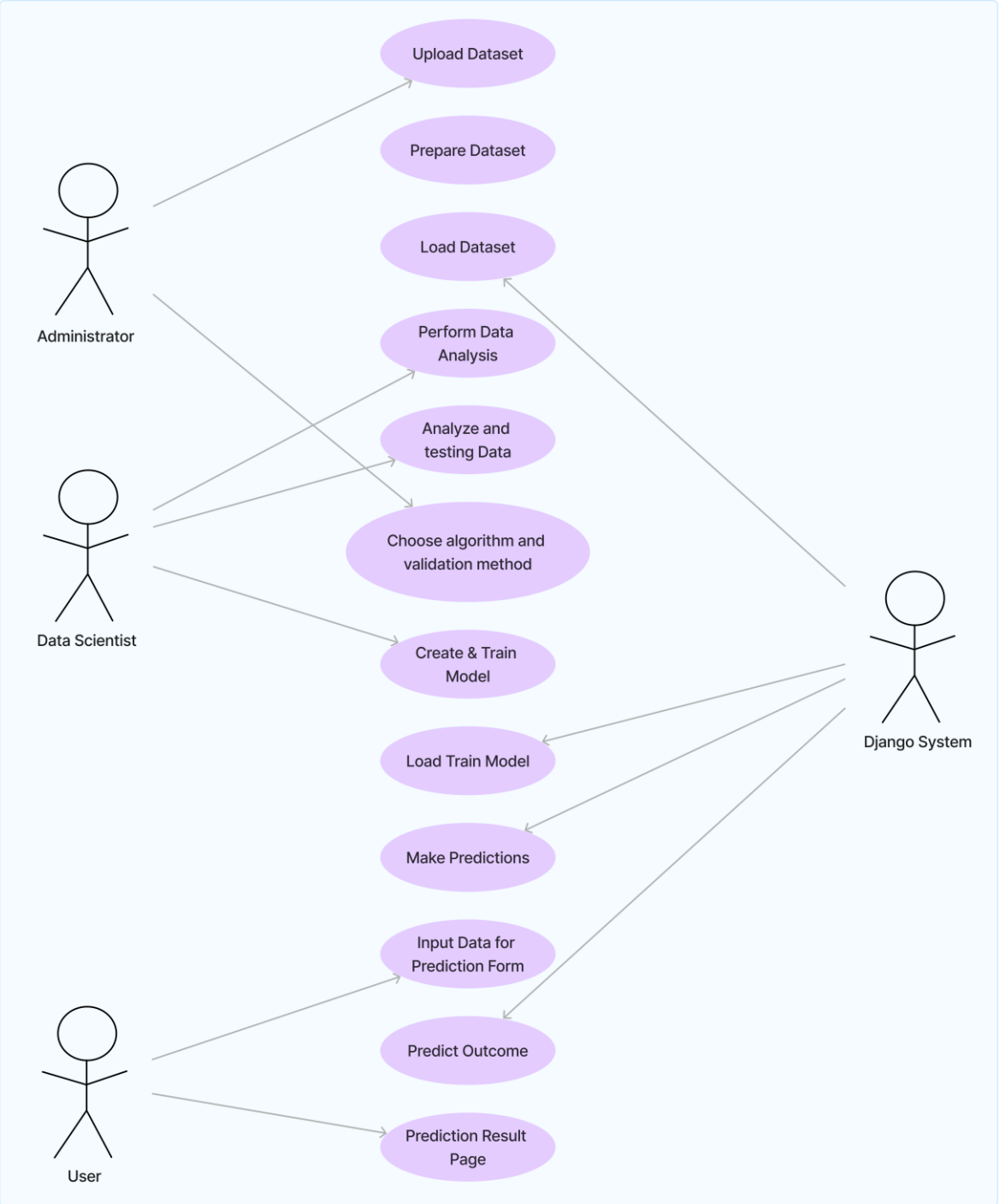

Figure 7: Use Case Diagram

- **Administrator**: Uploads and manages datasets.
- **Data Scientist**: Analyzes data, trains models, and deploys them.
- **User**: Interacts with the system to receive predictions.

Data Flow Diagram

Data Flow Diagram Overview: A data flow diagram illustrates the flow of data within the system, showing how data moves between processes and storage.

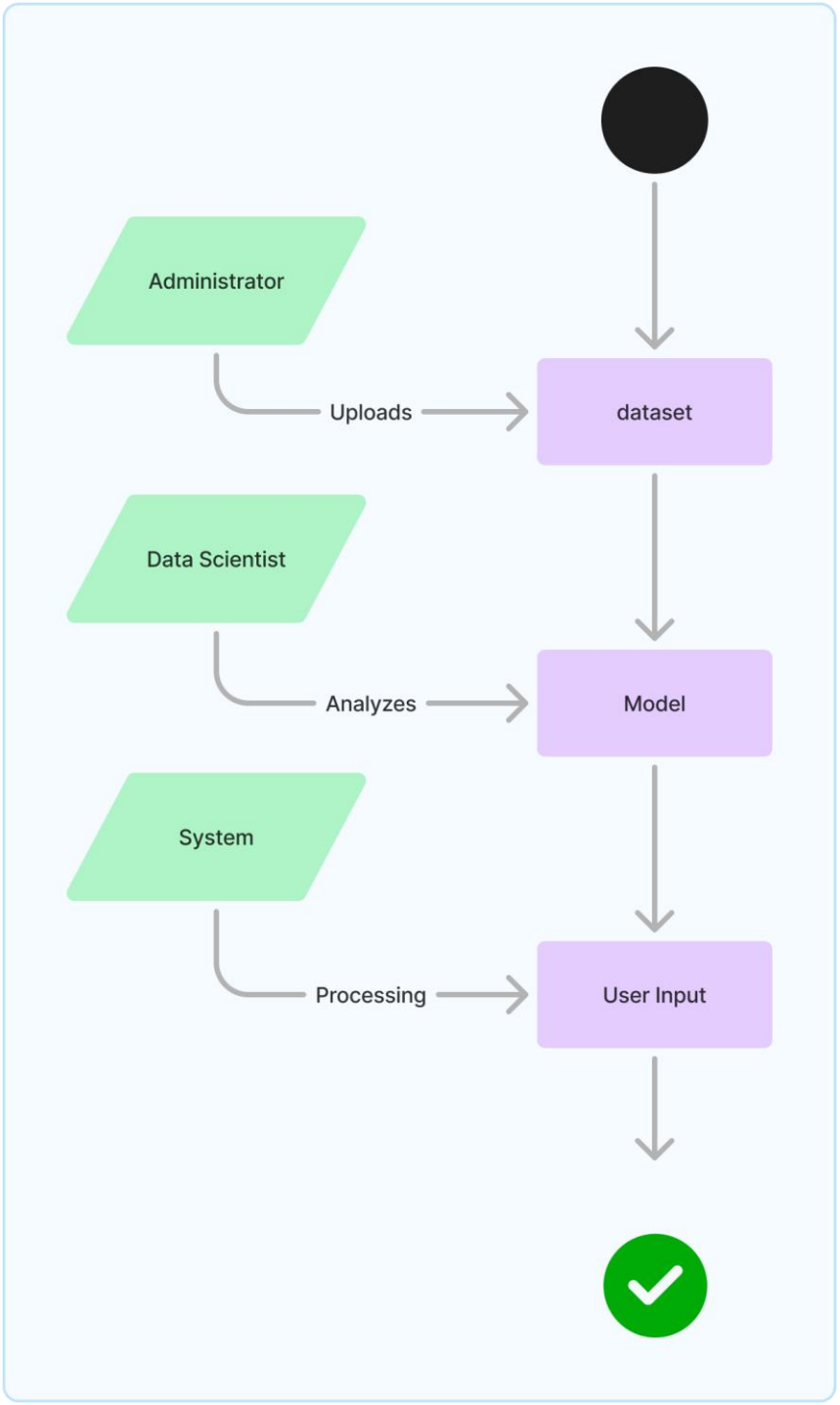

Figure 8: Data Flow Diagram

- **Dataset**: Flows from **Administrator** to **Data Scientist** for analysis.
- **Model**: Flow from **Data Scientist** to **System** for prediction.

State Diagram

State Diagram Overview: A state diagram shows the states of an object and how it transitions between states based on events.

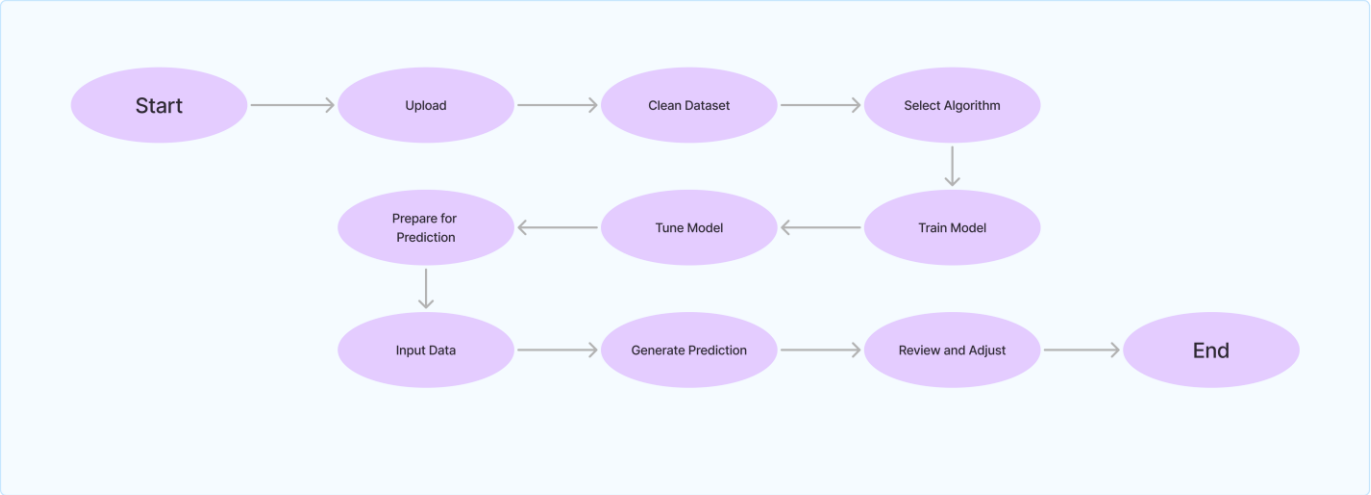

Figure 9: State Diagram

- **Dataset Loaded**: Initial state after dataset upload.
- **Model Trained**: State after model training.
- **Ready for Prediction**: State when the system is ready to accept input for predictions.

Component Diagram

Component Diagram Overview: A component diagram shows the physical components of the system and their relationships, including dependencies and interfaces.

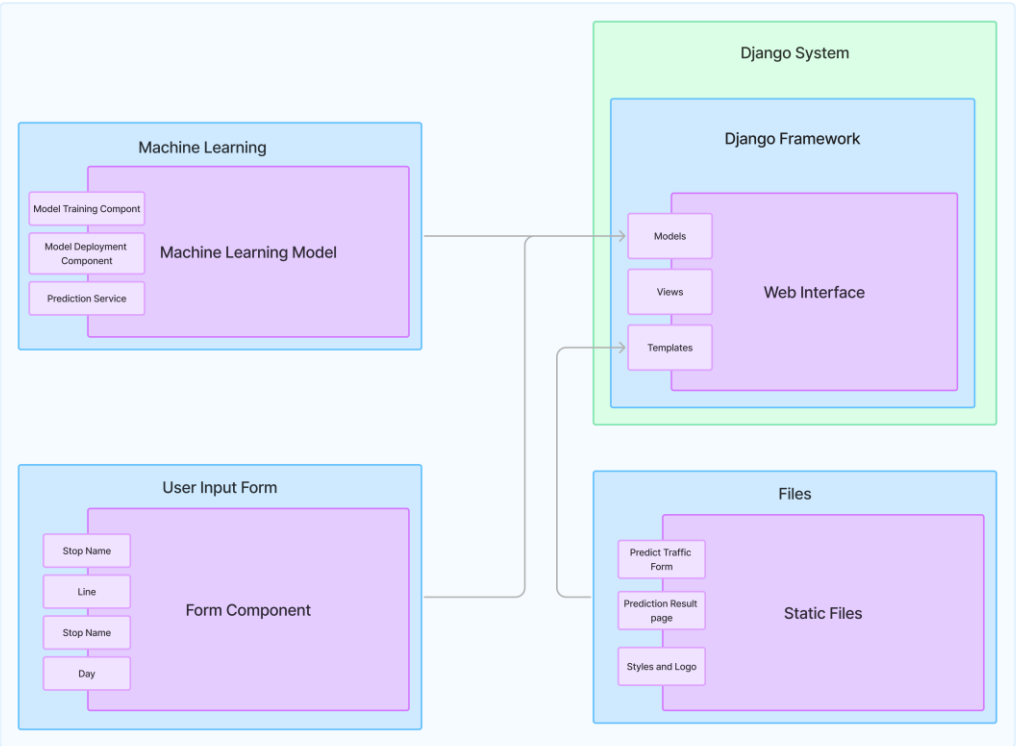

Figure 10: Component Diagram

- **Django System**: Includes Django framework components.
- **Machine Learning Model**: Represents components related to model development and deployment.
- **Database**: Stores datasets and system configurations.

Flowchart Diagram

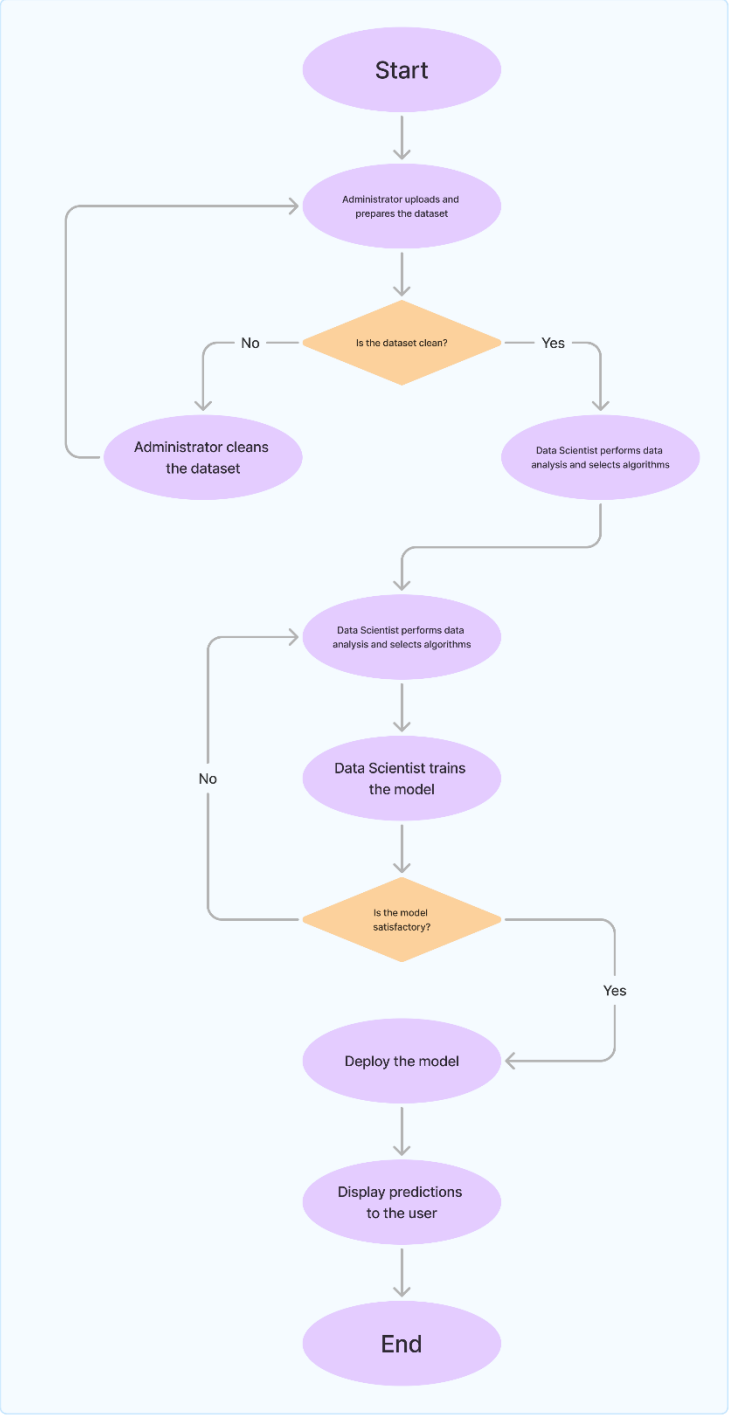

Figure 11: Flowchart Diagram

Flowchart Diagram Overview: A flowchart provides a visual representation of a process, showing steps and decision points in a sequential manner.

- **Dataset Upload**: Step for uploading datasets.
- **Data Analysis**: Step for analyzing dataset characteristics.
- **Model Training**: Step for training machine learning models.
- **Prediction Generation**: Step for generating predictions based on user input.

Deployment Diagram

Deployment Diagram Overview: A deployment diagram illustrates the physical deployment of components in a system, showing how software and hardware interact.

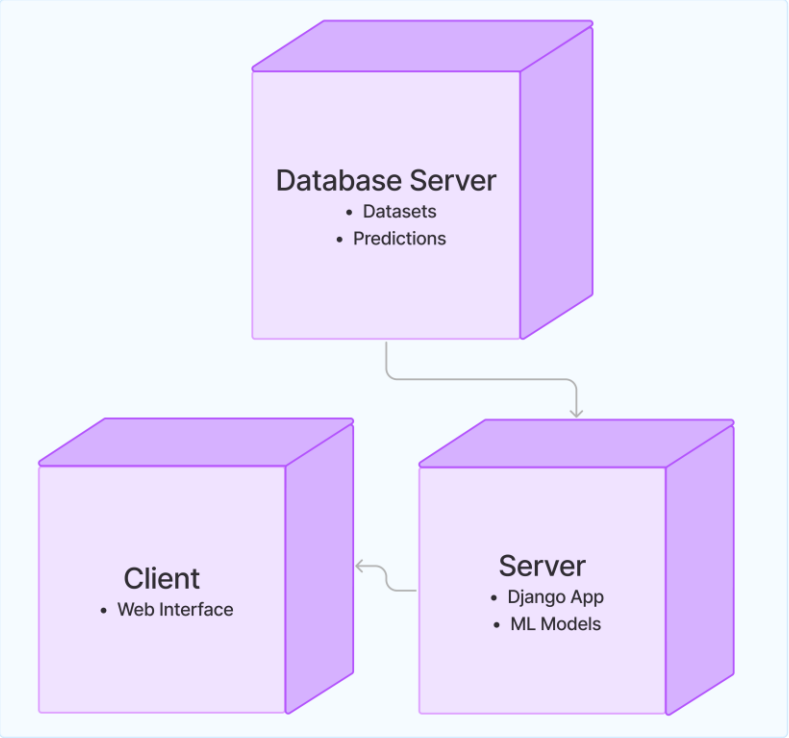

Figure 12: Deployment Diagram

- **Server**: Hosts Django application and machine learning models.
- **Database Server**: Stores datasets and prediction results.
- **Client**: Accesses the system through a web interface.

SCREENSHOTS & RESULTS

This section provides a detailed overview of the final results of the project, showcasing how the system works in practice. Users can see step-by-step screenshots of the process, from entering details to checking the subway forecast.

User Interface Description

Homepage

URL: http://127.0.0.1:8000/

Title: Subway Passenger Flow Forecasting

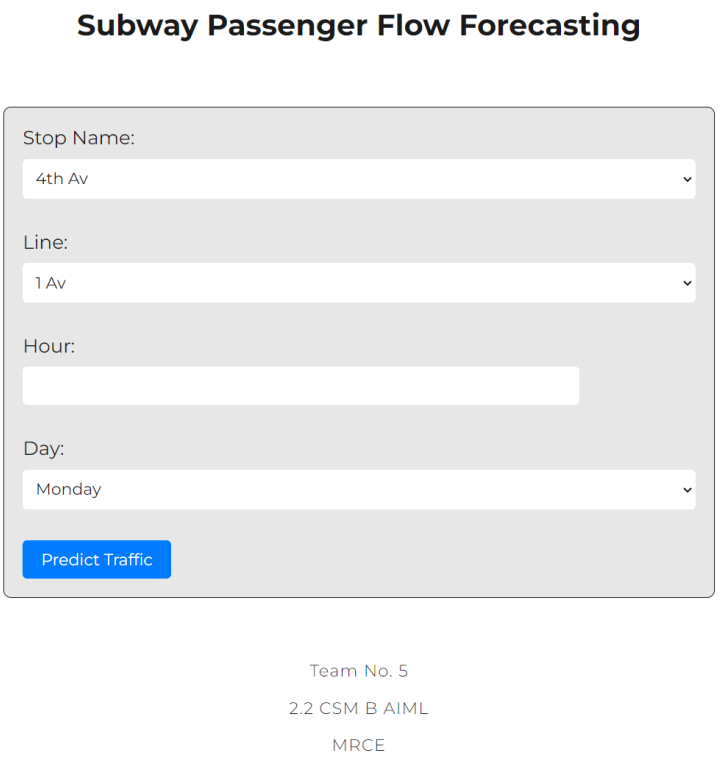

Figure 13: Homepage

Form Inputs

- **Stop Name:** Dropdown list where users can select the subway station name (e.g., Astoria).
- **Line:** Dropdown list where users can select the subway line number (e.g., 125 St).
- **Hour:** Input field where users can enter the hour (must be between 0 and 23).
- **Day:** Dropdown list where users can select the day of the week (e.g., Saturday).

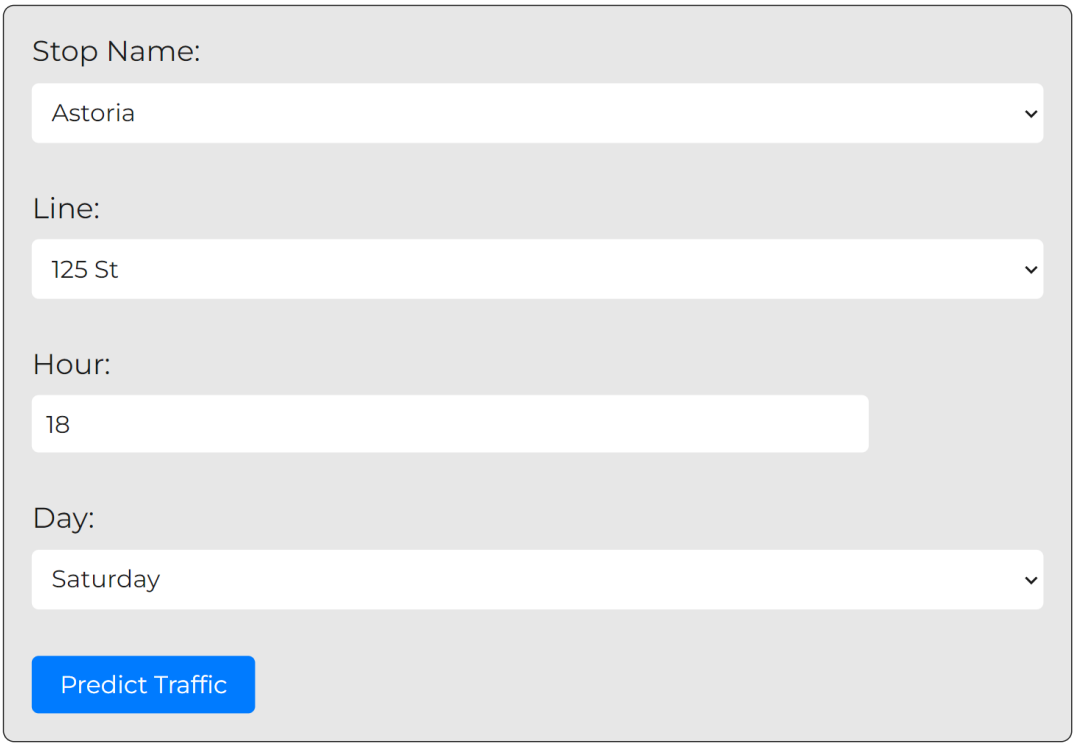

Figure 14: Form Inputs

Submit Button

- **Predict Traffic:** Button that users click to submit their inputs and trigger the prediction process.

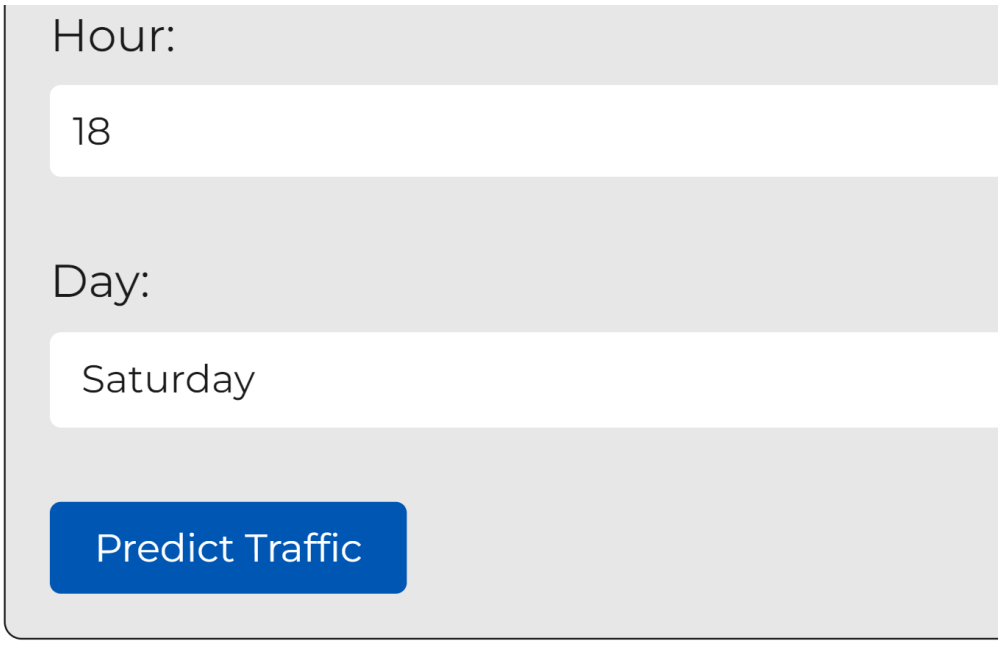

Figure 15: Submit Button

Expected Results

After submitting the form with valid inputs, the system will display the following forecasted traffic results:

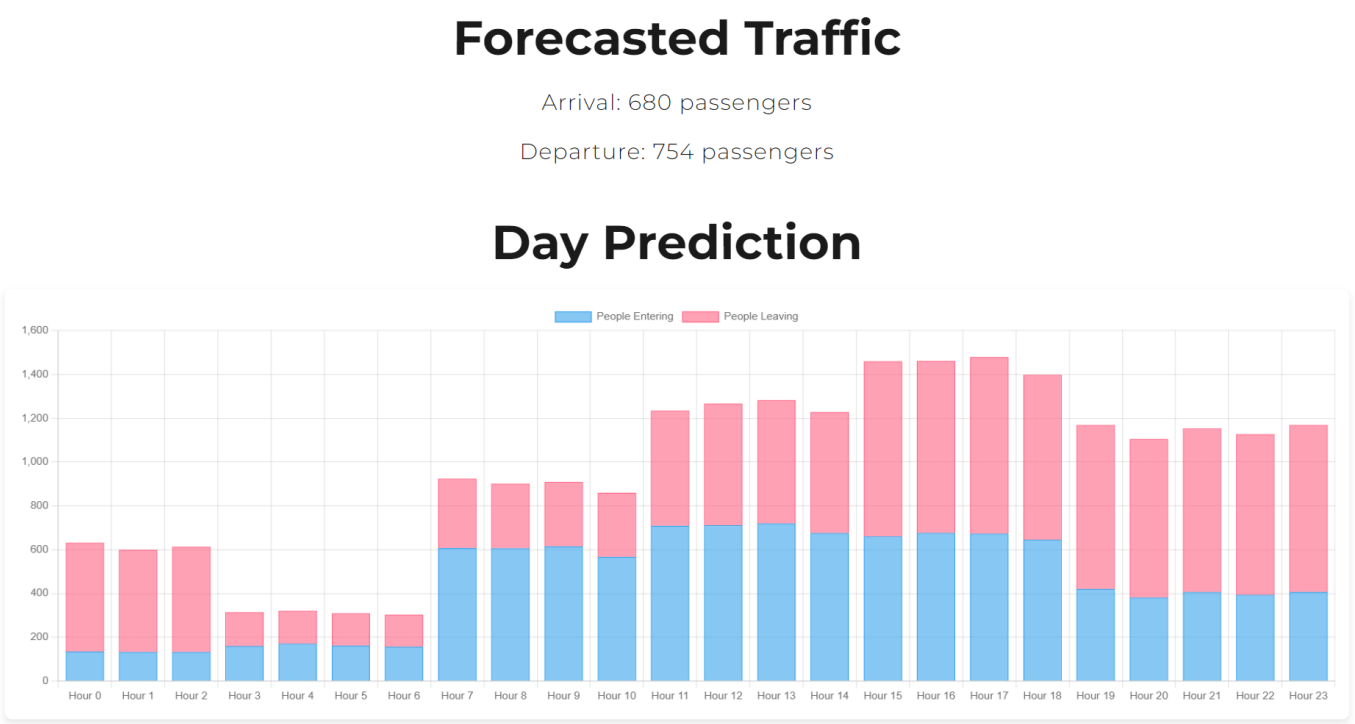

Figure 16: Result Page

Forecasted Traffic

- **Arrival:** Displayed number of passengers forecasted to arrive at the selected subway station during the specified hour and day (e.g., 706 passengers).
- **Departure:** Displayed number of passengers forecasted to depart from the selected subway station during the specified hour and day (e.g., 542 passengers).

Figure 17: Result Page (Forecasted Traffic)

Day Prediction (Bar Graph)

- **Bar Graph:** Graphical representation showing the predicted traffic (arrivals and departures) for each hour of the specified day. Each bar represents the passenger count for a specific hour.

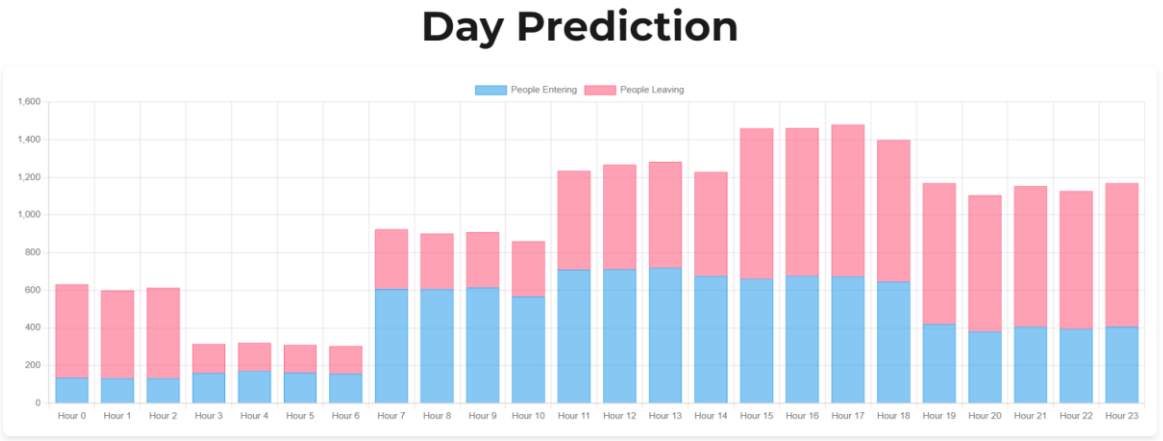

Figure 18: Result Page (Day Prediction Bar Graph)

Week Prediction (Bar Graph)

- **Bar Graph:** Graphical representation showing the predicted traffic (arrivals and departures) summed across all hours for each day of the week. Each bar represents the total passenger count for a specific day.

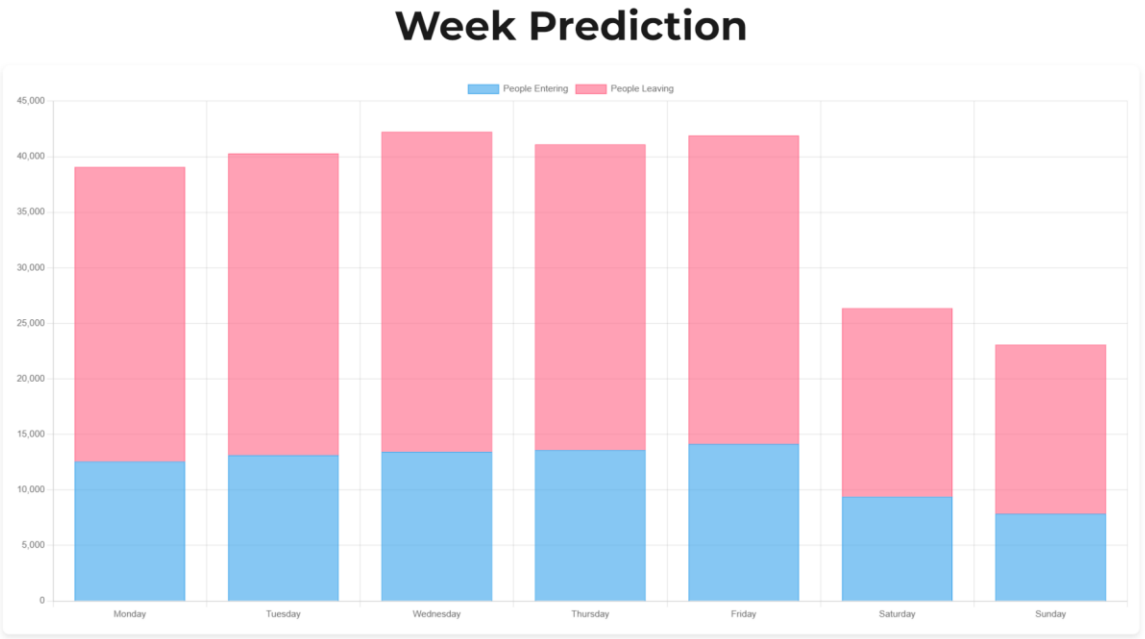

Figure 19 Result Page (Week Prediction Bar Graph)

Daily Prediction by Hours (Pie Chart)

- **Pie Chart:** Visual representation of the predicted traffic distribution (arrivals and departures) across different hours of the specified day. Each segment of the pie chart represents the passenger count for a specific hour.

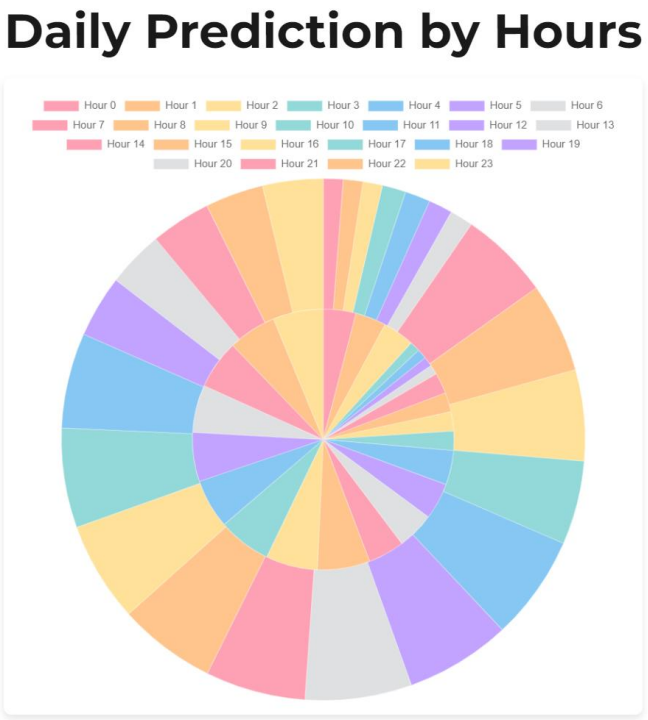

Figure 20: Result Page (Day Prediction Pie Chart)

Weekly Prediction by Days (Pie Chart)

Pie Chart: Visual representation of the predicted traffic distribution (arrivals and departures) summed across all hours for each day of the week. Each segment of the pie chart represents the total passenger count for a specific day.

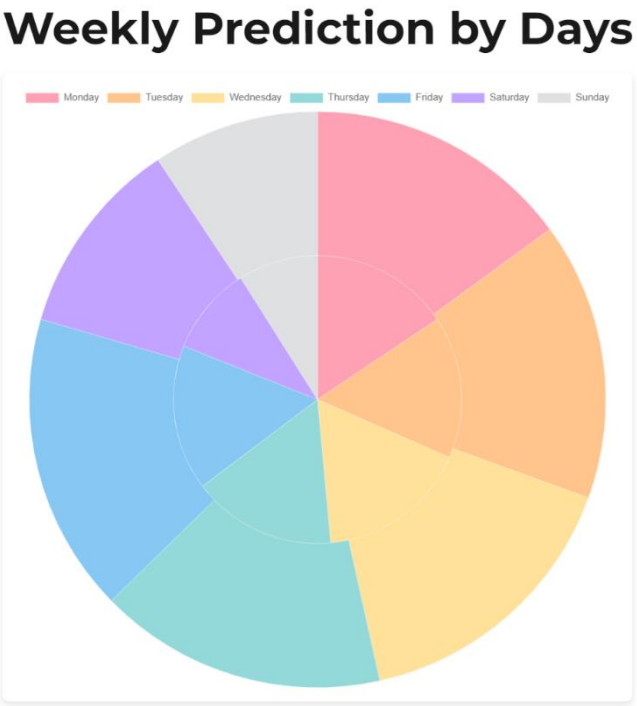

Figure 21: Result Page (Week Prediction Pie Chart)

Example Scenario

Inputs:

- Stop Name: Astoria
- Line: 125 St
- Hour: 12
- Day: Saturday

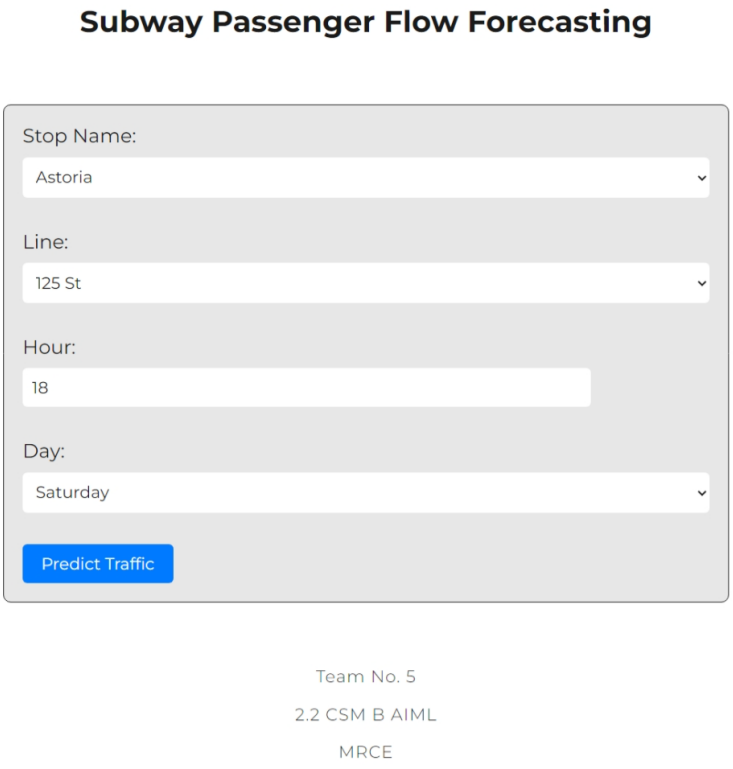

Figure 22: Example Scenario (Home Page)

Result:

- Arrival: 706 passengers
- Departure: 542 passengers

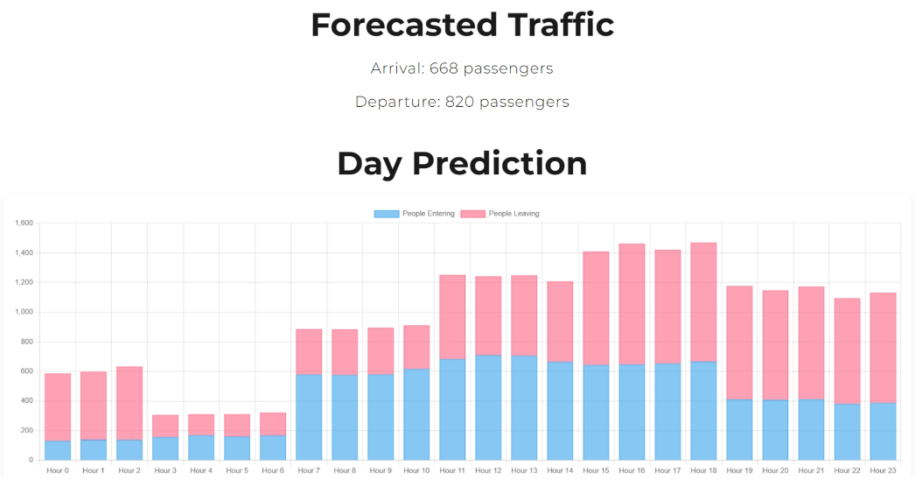

Figure 23: Example Scenario (Complete Result Page)

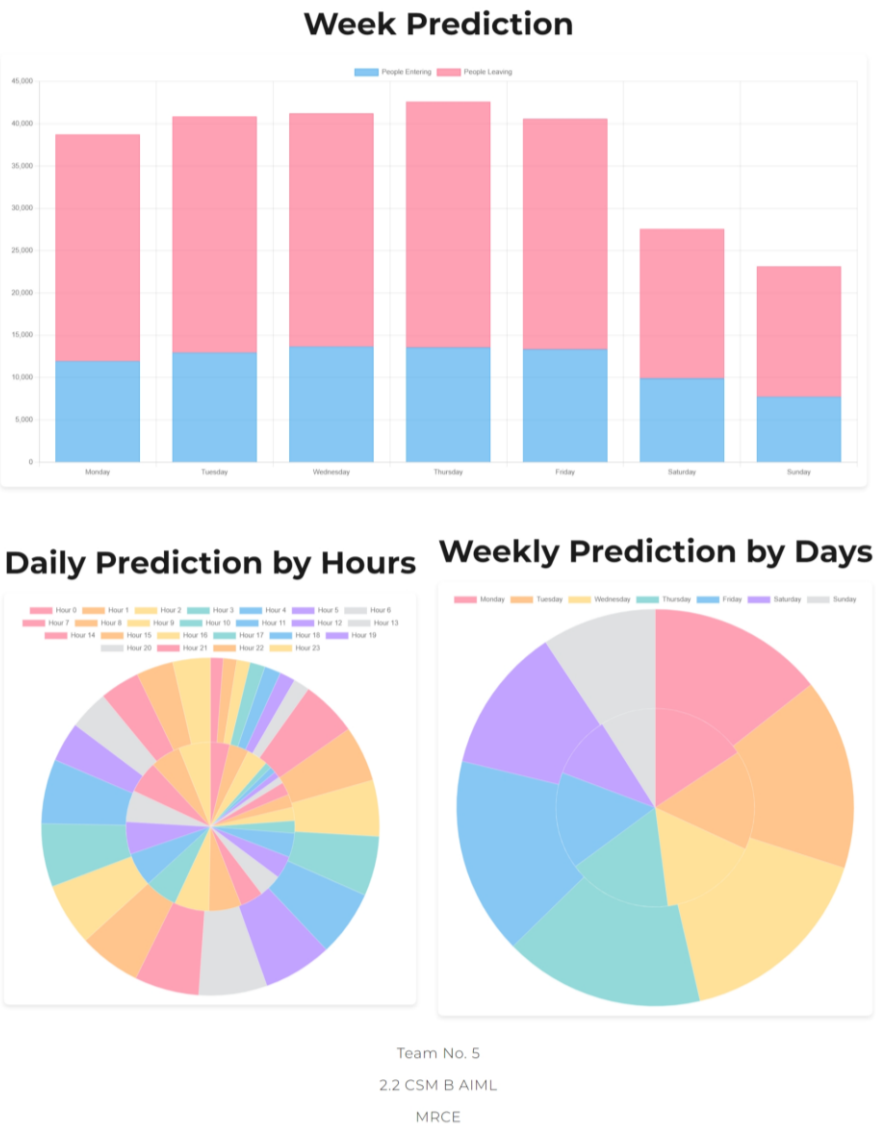

Figure 24: Example Scenario (Complete Result Page)

Graphs:

- **Day Prediction (Bar Graph):** Shows hourly predictions for Saturday.
- **Week Prediction (Bar Graph):** Shows daily predictions summed across all hours for each day of the week.
- **Daily Prediction by Hours (Pie Chart):** Shows hourly distribution of predictions for Saturday.
- **Weekly Prediction by Days (Pie Chart):** Shows daily distribution of predictions summed across all hours for all days of the week.

REFERENCES

References

- *Daniel Sanz Sobrino, Javier Andión, Juan C. Dueñas, José M. Del Álamo, Felix Cuadrado, "Model Generation and Origin Destination(OD) Matrix Forecasting for metro systems", 2023 7th International Young Engineers Forum (YEF-ECE), pp.112-117, 2023.**Show Article** **Google Scholar***
- *Lilan Peng, Xiu Wang, Hongchun Lu, Xiangyu Guo, Tianrui Li, Shenggong Ji, "Dynamic Spatio-Temporal Multi-Scale Representation for Bus Ridership Prediction", 2023 International Joint Conference on Neural Networks (IJCNN), pp.1-9, 2023.**Show Article** **Google Scholar***
- *Peng Xie, Minbo Ma, Tianrui Li, Shenggong Ji, Shengdong Du, Zeng Yu, Junbo Zhang, "Spatio-Temporal Dynamic Graph Relation Learning for Urban Metro Flow Prediction", IEEE Transactions on Knowledge and Data Engineering, vol.35, no.10, pp.9973-9984, 2023.**Show Article** **Google Scholar***
- *Erlong Tan, Haoyang Yan, Kaiqi Zhao, Xiaolei Ma, Zhenliang Ma, Yuchuan Du, "MVOPFAD: Multiview Online Passenger Flow Anomaly Detection", IEEE Sensors Journal, vol.24, no.9, pp.14668-14681, 2024.**Show Article** **Google Scholar***
- *Shuxin Zhang, Jinlei Zhang, Lixing Yang, Chengcheng Wang, Ziyou Gao, "COV-STFormer for Short-Term Passenger Flow Prediction During COVID-19 in Urban Rail Transit Systems", IEEE Transactions on Intelligent Transportation Systems, vol.25, no.5, pp.3793-3811, 2024.**Show Article** **Google Scholar***

Libraries and Frameworks

- **Django:** Web framework for Python. Django Documentation
- **Pandas:** Data manipulation and analysis library. Pandas Documentation
- **Scikit-learn:** Machine learning library. Scikit-learn Documentation
- **Joblib:** Load and Create Models. Joblib Documentation
- **Chart.js:** JavaScript library for charts and graphs. Chart.js Documentation

Dataset

- **NYC Subway Traffic 2017-21:** Available on Kaggle. Dataset Link