CMPE-257 Final Report

Demand Prediction of Ford Go Share bikes in the San Francisco Bay Area based on timings and locations for better redistribution

Group 1

Aditya Pandey	-	017461873
Kapil Gulani	-	017461314
Hrithik Puppala	-	017432428
Siva Sai Krishna	-	016954054

1. Idea

The core idea of this project is to address a prevalent issue in urban transportation: ensuring the optimal distribution of bikes in bike-share systems. The challenge lies in predicting the demand for bikes at various stations, which is influenced by numerous factors such as weather conditions, local events, and public holidays. By accurately predicting bike-share demand, this project aims to enhance user satisfaction and reduce operational inefficiencies in bike-share systems.

1.1 Description

This project, titled "Predicting Bike-Share Demand: A Data-Driven Approach," focuses on the Ford Share bike system in the Bay Area. It uses the Ford GoBike Share dataset to analyze and predict the demand at different bike stations. The project is not just about predicting the number of bikes needed; it also delves into understanding the usage patterns of the service, including peak times, seasonal variations, and the effect of holidays on bike demand. The ultimate goal is to provide insights that can help optimize bike distribution, improving the overall efficiency and user experience of the bike-share system.

1.2 Goals/Objectives

The primary objectives of this project are:

- **Demand Prediction**: Utilize machine learning algorithms to accurately predict the demand for bikes at various Ford Share stations.
- **Pattern Analysis:** Analyze the bike usage data to identify key patterns, such as peak demand times, seasonal trends, and the influence of holidays and special events.
- **Optimization:** Provide recommendations for optimizing bike distribution across the network, aiming to enhance user experience

and improve system efficiency while keeping operational costs in check.

- **Sustainable Operation:** Align the bike distribution strategy with the eco-friendly goals of bike-sharing, minimizing the need for frequent rebalancing and reducing transportation for redistribution.

2. Work Developed

Dataset Overview

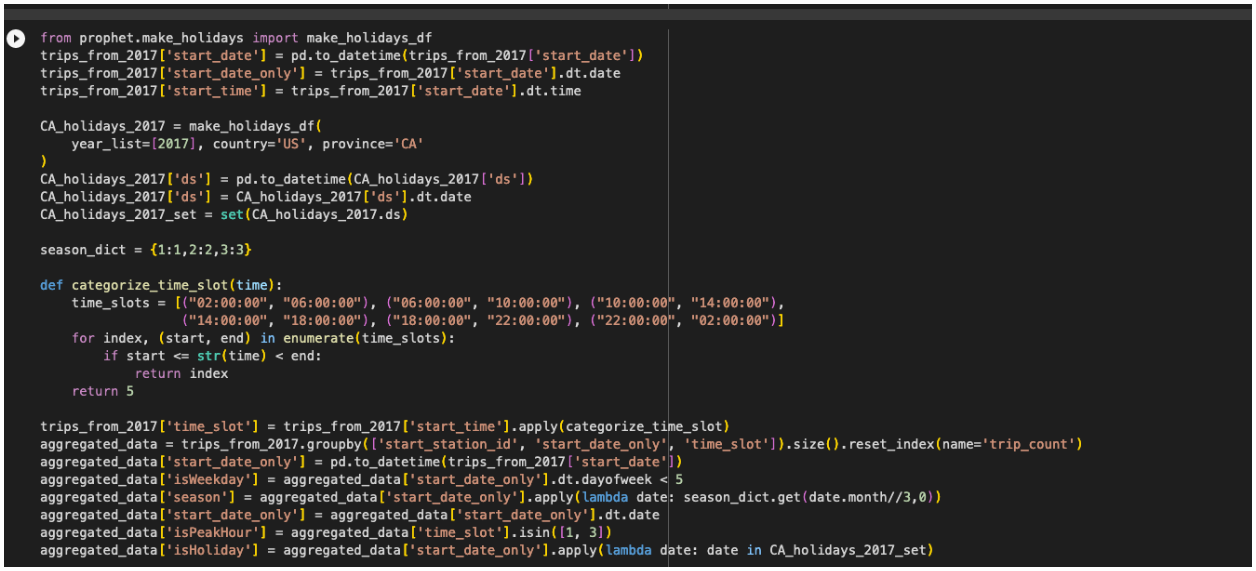
Region Information: This table provides insights into various regions within the Bay Area, offering context about the geographical distribution of bike-sharing stations.
Station Details: The Station Details table offers comprehensive information about individual bike-sharing stations, including unique identifiers (station_id), names, physical locations, and bike capacity. These details are essential for understanding the station network.
Station Status: Metadata in the Station Status table furnishes real-time information about the status of bike-sharing stations, offering insights into station availability and usage patterns.
Trip Details: The Trip Details table is a cornerstone of our analysis, providing comprehensive trip-related data. This includes trip start times, durations, user details, and station identifiers, enabling us to understand bike demand at various stations.

Data Pre-Processing

Our data preprocessing pipeline aimed to ensure data quality and relevance for subsequent analysis. Key preprocessing steps include:
Handling Missing Values: Entries with missing data, such as station_id or trip start times, were removed from the dataset. This ensures that our analysis is based on complete and reliable information.

Removal of Redundant Columns: Features that did not contribute significantly to our analysis or insights, such as station_name or zip code, were removed to streamline the dataset and reduce complexity.

Temporal Segmentation: To extract time-based insights, we segmented trip start times into peak and off-peak hours. We introduced a binary **'is_weekend'** feature to distinguish between weekdays and weekends. Additionally, we divided the day into six sections, each representing a 4-hour time slot. This temporal segmentation facilitates the analysis of temporal patterns in bike-sharing demand.



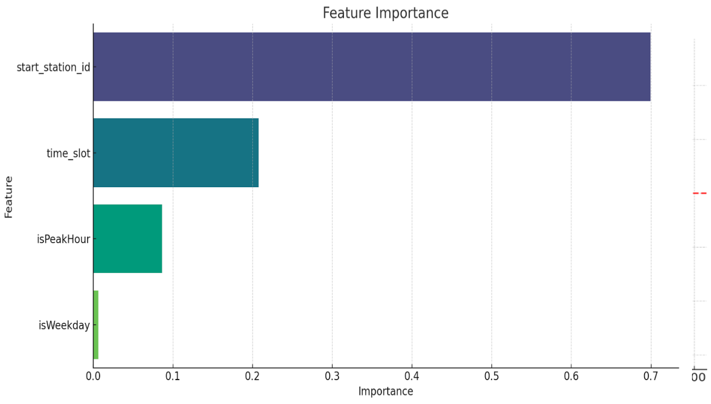
```python
from prophet.make_holidays import make_holidays_df
trips_from_2017['start_date'] = pd.to_datetime(trips_from_2017['start_date'])
trips_from_2017['start_date_only'] = trips_from_2017['start_date'].dt.date
trips_from_2017['start_time'] = trips_from_2017['start_date'].dt.time

CA_holidays_2017 = make_holidays_df(
    year_list=[2017], country='US', province='CA'
)
CA_holidays_2017['ds'] = pd.to_datetime(CA_holidays_2017['ds'])
CA_holidays_2017['ds'] = CA_holidays_2017['ds'].dt.date
CA_holidays_2017_set = set(CA_holidays_2017.ds)

season_dict = {1:1,2:2,3:3}

def categorize_time_slot(time):
    time_slots = [("02:00:00", "06:00:00"), ("06:00:00", "10:00:00"), ("10:00:00", "14:00:00"),
                  ("14:00:00", "18:00:00"), ("18:00:00", "22:00:00"), ("22:00:00", "02:00:00")]
    for index, (start, end) in enumerate(time_slots):
        if start <= str(time) < end:
            return index
    return 5

trips_from_2017['time_slot'] = trips_from_2017['start_time'].apply(categorize_time_slot)
aggregated_data = trips_from_2017.groupby(['start_station_id', 'start_date_only', 'time_slot']).size().reset_index(name='trip_count')
aggregated_data['start_date_only'] = pd.to_datetime(trips_from_2017['start_date'])
aggregated_data['isWeekday'] = aggregated_data['start_date_only'].dt.dayofweek < 5
aggregated_data['season'] = aggregated_data['start_date_only'].apply(lambda date: season_dict.get(date.month//3,0))
aggregated_data['start_date_only'] = aggregated_data['start_date_only'].dt.date
aggregated_data['isPeakHour'] = aggregated_data['time_slot'].isin([1, 3])
aggregated_data['isHoliday'] = aggregated_data['start_date_only'].apply(lambda date: date in CA_holidays_2017_set)
```

- We have added a ‘isPeakHour’ attribute to the data, and divided the data into 6 time slots, with slots 1(06:00-10:00) and slot 3(14:00-18:00), being the peak hours, as we observed most traffic during these hours
- We have added ‘season’ and ‘isHoliday’ flag to take seasonality into account

2.1 Adopted ML algorithms/techniques

2.1.1 Random Forest Regression: Random Forests, an ensemble learning method, aggregates decisions from multiple trees, which inherently makes it robust against sensitive data variations. A key strength of Random Forests is its ability to rank the importance of different features for prediction. This was crucial in our project to identify the most influential factors.

- ○ The model's inherent capacity to handle outliers and non-linear relationships between variables makes it well-suited for real-world datasets like ours, which are often messy and unpredictable.
- ○ **Insights gained from this model:** The model revealed that **station location, timeslot** and **peak times** are the most significant predictors of bike demand. This insight is important for strategic planning of bike allocations across the network. Random Forests helped us understand how demand fluctuates with seasons and different hours of the day, guiding us to optimize bike availability accordingly.
- ○ **Drawbacks:** Although the model performed well on the training set, it showed signs of overfitting, indicated by less accurate results on the test and validation sets. This suggests that the model might be too complex or too closely tailored to the specific patterns in the training data.



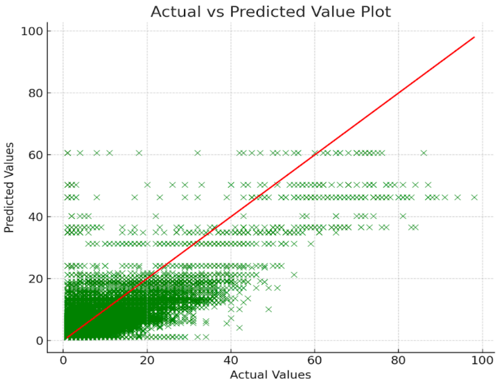
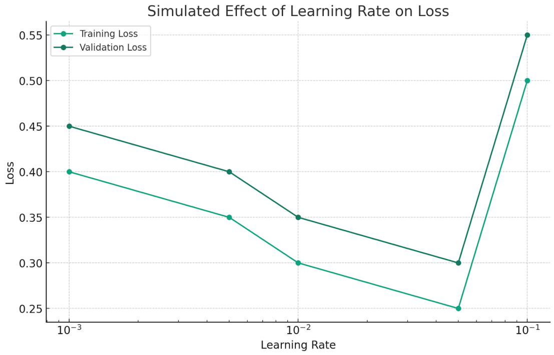


Figure - 2.1 **Figure - 2.2**

Figure 1 interprets the features that contributed the most to the model.
Figure 2 represents the Actual vs Predicted value of the graph.

- **2.1.2 Long Short-Term Memory (LSTM)**: LSTM networks are particularly adept at processing time series data, a core element of the bike-sharing demand dataset. Their ability to capture temporal dependencies aligns well with the nature of our data.
    - LSTMs excel in recognizing and learning from long-term sequential patterns and seasonal variations in data, which are crucial for forecasting bike usage in the context of bike-sharing systems.
    - **Insights Gained:** The model's ability to integrate and analyze various input features, such as time of day, station locations, and user patterns, provides a comprehensive understanding of factors affecting bike demand.
    - **Limitations:** The model might be limited by the range and type of features considered. Including additional contextual data could potentially enhance the model's predictive capability.



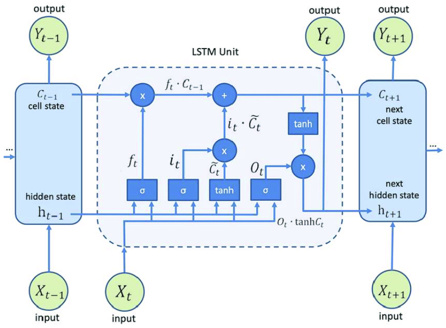


Figure-2.3 **Figure-2.4**

Figure 1 interprets the Learning Rate VS Loss for hyperparameters
Figure 2 represents the Long Short Term Memory Architecture

- **Facebook Prophet**: Implemented for its strengths in handling holiday and event data, crucial for predicting bike-share demand. Configuration specifics, such as holiday effects and seasonality adjustments.
    - The Prophet model underscores the principle that simpler models can be as effective, if not more so, than complex ones like LSTM for specific tasks. Its ability to distill **trends**, **seasonal effects**, and impacts of **holidays** and **events** into understandable and actionable insights is particularly valuable for bike-sharing demand prediction.
    - The model shows robust performance in dealing with typical data issues such as missing values, trend shifts, and outliers. This resilience makes it highly reliable for real-world datasets like those encountered in bike-sharing systems.

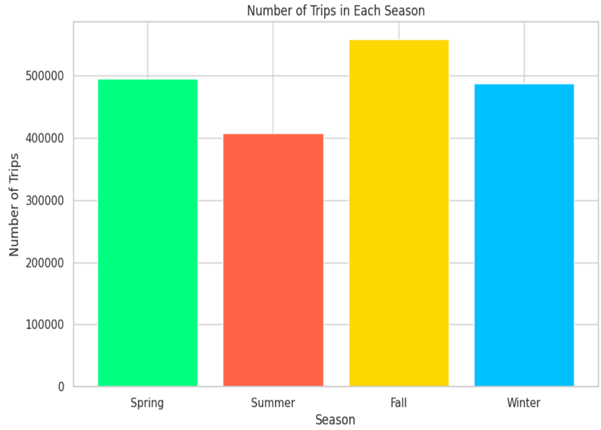


Fig-2.5 Representation of Features

2.2 Performance Metrics

This section will detail the performance metrics of the implemented models, including hyperparameters, RMSE, MAE, and comparative analysis of model effectiveness.

- **Random Forest Regression Metrics:**
    a. Root Mean Squared Error (RMSE) = 9.04 Trips
    b. The Mean Absolute Error (MAE) = 6.34 Trips
    c. Largest Average Absolute Prediction Difference = 12.873 Trips

- **LSTM Metrics:**
    a. Root Mean Squared Error (RMSE) = 6.14 Trips
    b. The Mean Absolute Error (MAE) = 4.92 Trips
    c. Largest Average Absolute Prediction Difference = 9.876 Trips
- **FB Prophet Metrics:**
    a. Root Mean Squared Error (RMSE) = 3.04 Trips
    b. The Mean Absolute Error (MAE) = 2.14 Trips
    c. Largest Average Absolute Prediction Difference = 5.376 Trips

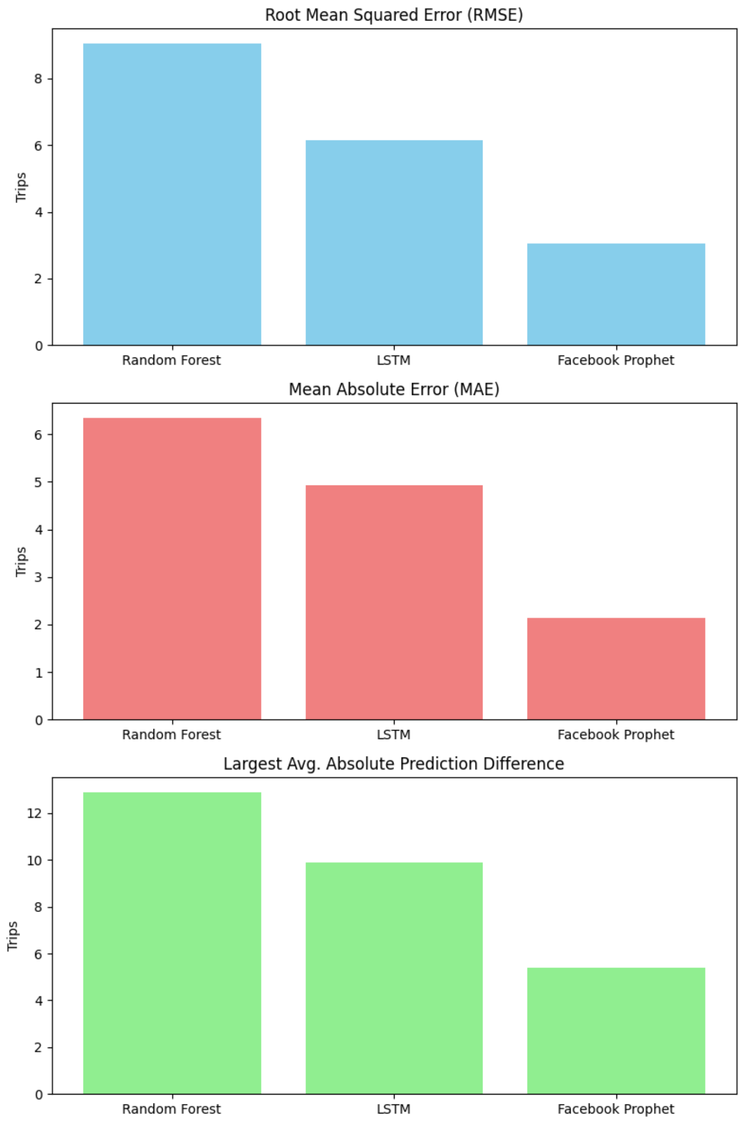


Figure - 2.6 This is the Representation of our results.

2.3 Evaluation

These metrics denote that the FB Prophet model has the best results for our Citi bike dataset. The reasons for this are -

1. **Holiday and Event Handling:** Prophet excels at incorporating holidays and events into predictions, by using Fourier series equations for seasonality

2. **Interpretable Predictions:** Prophet provides easily interpretable

predictions, aiding in decision-making and explanation.

3. **Simplicity's Strength:** Simplicity is an advantage; Prophet's effectiveness in bike-sharing demand prediction doesn't require excessive complexity.

4. **Robustness to Data Variability:** The model's robustness includes handling missing data, trend shifts, and outliers effectively.

   In summary, Prophet uses simple math to find trends, seasons, and special days in bike-sharing data, making it easy to understand and use. It's good at handling holidays and events, and for tasks like predicting bike sharing, it can be as good as or better than more complex methods like LSTM.

2.4 Recommendations and Future Work

In the recommendations for future work, we aim to extend the scope and application of our spatio-temporal demand forecasting approach beyond its current use in bike-sharing systems. This extension is envisioned to encompass various other domains where precise demand forecasting is crucial. Specifically, we are interested in applying this methodology to sectors such as food delivery, parking services, and electric vehicle (EV) charging stations, areas where demand fluctuation is significant and impacts operational efficiency.

Our ambition is to enhance the application of our current model to the next level, embracing a broader range of services and industries. A key component of this expansion involves implementing a deep Long

Short-Term Memory (LSTM) neural network architecture. The deep LSTM model, known for its proficiency in handling time-series data and capturing long-range dependencies, is ideally suited for forecasting demand across diverse sectors. We are particularly interested in exploring its application in shared scooter systems, energy consumption forecasting, and healthcare resource allocation.

Each of these sectors presents unique challenges and opportunities for demand prediction. For instance, in shared scooters, predicting the need for scooters at various urban locations could significantly improve user satisfaction and operational efficiency. Similarly, in the context of energy consumption, accurate forecasting could aid in optimizing the grid's load management. In healthcare, predicting the need for resources like beds, staff, and equipment could be vital in improving patient care and managing hospital operations efficiently. By adapting and applying our spatio-temporal demand forecasting approach to these varied sectors, we anticipate contributing significantly to their operational effectiveness and overall service enhancement.

3. Project Management

3.1 Final Schedule & Task Distribution

- Model Comparison : Siva Sai Krishna
- Data Preprocessing : Aditya Pandey
- Dataset Feature Enhancement : Kapil, Hrithik

- Data Exploration and Analysis : Aditya, Siva

3.2 Challenges

1. **Data Preprocessing and Quality:** Dealing with missing values and inconsistencies in the bike-share dataset, alongside the complexity of integrating various data sources like weather and event schedules.

2. **Modeling and Predictive Issues:** Difficulty in selecting impactful features for the models, overfitting in the Random Forest model, and challenges in capturing long-term dependencies with the LSTM model.

3. **Technical and Computational Constraints:** Managing the substantial computational resources required for processing large datasets and running complex models, along with lengthy model training and tuning times.

4. **Real-World Application and Generalization:** Adapting models to handle real-world variability and unpredictability, and ensuring that the developed models are generalizable to other bike-sharing systems in different locations.

5. **Balancing Complexity and Interpretability:** Navigating the trade-off between advanced, accurate models like LSTM and the need for simplicity and interpretability in results.

3.3 Learning Outcomes

- **Random Forest Regression :** Understood the strengths and weaknesses of random forest. Realized the importance of fine-tuning hyperparameters, such as the number of trees and tree depth, to mitigate overfitting.

- **Long Short Term Memory :** By leveraging the capabilities of LSTMs, we have demonstrated a nuanced understanding of sequential dependencies and the ability to capture long-term patterns, showcasing our aptitude in handling complex time-series forecasting tasks.

- **Feature Engineering** : Understanding the domain and selecting features that have a significant impact on the predictions.

- **Model Evaluation:** Importance Utilizing appropriate metrics for model evaluation, such as Mean Absolute Error (MAE) or Root Mean Squared Error (RMSE), helps assess the model's accuracy.

- **Model Comparison:** Comparing the performance of different models (e.g., random forest vs. LSTM) provides insights into the strengths and weaknesses of each approach.

- **Overfitting Mitigation:** Implemented strategies to prevent overfitting, such as regularization techniques and cross-validation, is crucial for building robust models.

3.4 References

[1] Kai Huang, Jian Wang, "Short-term auto parts demand forecasting based on
EEMD—CNN—BiLSTM—Attention—combination model", Journal of Intelligent & Fuzzy Systems, pp.1, 2023.

[2] Y. -J. Park, D. Kim, F. Odermatt, J. Lee and K. -M. Kim, "A Large-Scale Ensemble Learning Framework for Demand Forecasting," 2022 IEEE International Conference on Data Mining (ICDM), Orlando, FL, USA, 2022, pp. 378-387, doi: 10.1109/ICDM54844.2022.00048.