CMPE 255 - Data Mining
Trip Prediction using Bay Area Bike Share Data

SAN JOSÉ STATE
UNIVERSITY

Submitted to

Gheorghi Guzun

on

01/12/2022

by

Team 04 (Loud Logic)

Git Repo

(https://github.com/stavan30/cmpe255_final_project)

Anshul Shandilya	016039894
Girish Bisane	016650348
Parth Saraiya	016715608
Stavan Patel	016622398

TABLE OF CONTENTS

1. **Introduction**

1.1. Motivation

1.2. Objective

2. **System Design and Implementation Details**

2.1.Algorithms Selected

2.2.System Design and Architecture

3. **Experiments**

3.1.Dataset

3.2.Feature Creation and Selection

3.3.Creating Training Data Set

3.4.Choices of Algorithms To Train and Evaluate

3.5.Performance Comparison

3.6.Running and Comparing the Algorithms

3.7.Selecting Algorithms for Deployment

4. **Discussions and Conclusions**

4.1.Decisions made

4.2.Difficulties faced

4.3.Things that worked well and what didn’t

4.4.Conclusion

5. **Project Plan/Task Distribution**

6. **Reference**

Chapter 1: Introduction

1. Motivation

With the quick acceptance of a healthy lifestyle, biking has become much more popular in today's society. People all over the world have embraced the practice of using bicycles for daily tasks rather than forking over money for pricey public transportation or getting stuck in congested traffic during bad weather. Over 12% of Americans bike daily, and every sixth person in the world owns a bicycle. By 2022, more than 43 million people could potentially use bicycles to help with the issue of heavy traffic in major cities, according to a WRC report. Many large companies, including Bikelink and Scoot, have suffered significant losses because they were unable to foresee the supply and demand for their bikes in a given place. The Bay Area Bike Ride dataset contained a wide variety of patterns, user behavior, and ideologies that offered great potential for research. We decided to use the data mining concepts that were taught to us in class to find the solution because we thought that this was a good problem statement to work on.

1. Objectives

The regression and projections of bike riders at a specific city for a specific day using the numerous time, weather, and behavior variables that we have access to, are the main goals of our project. We plan to analyze the dataset, and after manipulation and extraction of data, perform pre-processing techniques on the dataset and apply regression algorithms and methods to be able to predict the trips for individual cities using the processed data.

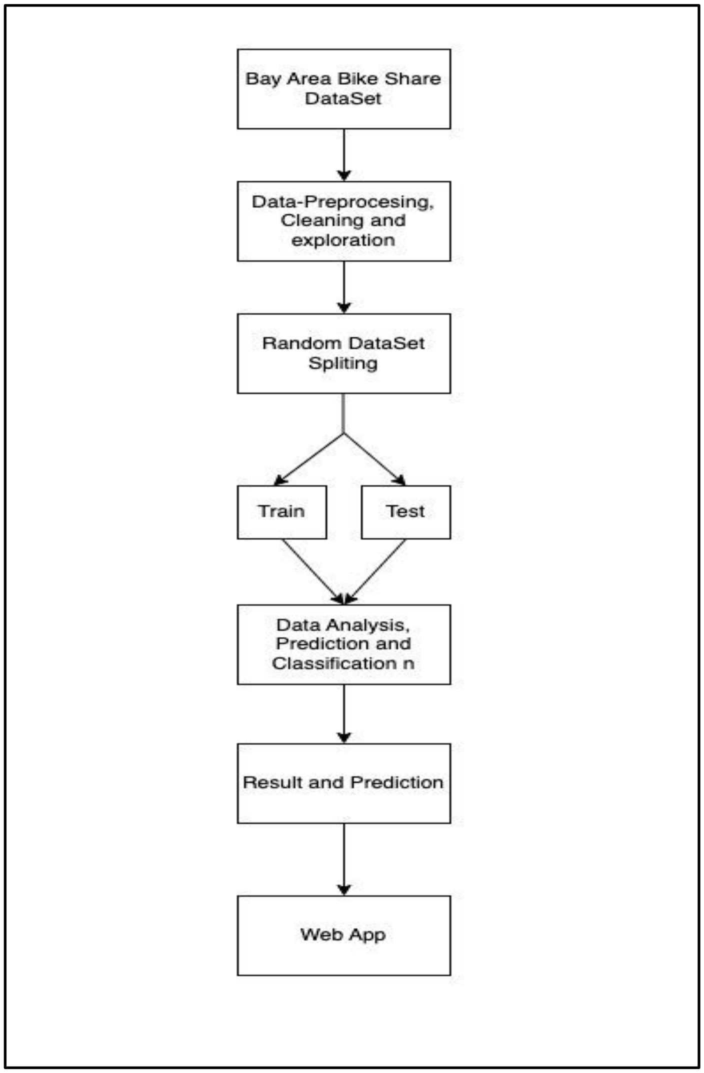
Chapter 2: System Design and Architecture

1. Algorithms Selected

Algorithms that we considered for our regression problem were:

1. Random Forest Regression
2. Extra Trees Regression
3. XGBoost Regressor
4. KNN Regression
5. Lasso Regression
6. Gradient Boosting Regressor

2. System Design and Architecture


Bay Area Bike Share DataSet
Data-Preprocesing, Cleaning and exploration
Random DataSet Spliting
Train
Test
Data Analysis, Prediction and Classification n
Result and Prediction
Web App

Chapter 3: Experiments

1. Dataset

(Link for the dataset) - Bay Area Bike Ride Dataset

The Dataset includes 4 CSV files:

- **station.csv** - Contains data that represents a station where users can pick up or return the bikes.
- **status.csv** - data about the number of bikes and docks available for a given station and minute.
- **trips.csv** - Data about individual bike trips.
- **weather.csv** - Data about the weather on a specific day for certain zip codes.

Description

There are a total of 4 CSV files, with each file having data of a particular day. The data all together consists of holidays, business day, month, weekday, temperature, dew points, winds speed, gust speed, precipitation inches, cloud cover, fog, rain and thunderstorm to name a few features.

2. Feature Selection and Creation

Since we had 4 different csv files, we had to perform data analysis and exploration to determine which of the subsets of these 4 files we can combine to create our training dataset. The process to do this was to import the csv files to their respective dataframes and determine how much data we have to work with. After we obtained the dataframes for each csv file, we proceeded to work on one dataframe at a time.

Trips.csv

For the trips dataframe, we saw that the data had information about trips which included trip duration, start date end date, stations name and ID etc. The first step was to determine if the dataframe had any null values. We found a significant amount of null values in the 'zip code' column. Now, for this 'zip code' column, we tried to determine what the zip code is about and it was found that majority of this column had data which were not zip codes, for example ado346. This, added with the number of null values in the column motivated us to drop the column entirely. We also decided to drop the subscriber type column which had information about if a user is a subscriber or not, since our problem was to predict the bike trips for a city, and not a user and a lot of other features and at the end. After analyzing trips dataframe further, we decided to extract features from it. We decided to count the number of entries for each date in the dataset, and using this information, we created a new dataframe which only consisted of dates and the number of trips for that date. We later on use the count for each date as the labels for our training data.

For feature creation, we needed more data for each date to be able to effectively perform regression. So, we created features 'holiday', 'business day', 'business day'. This was done by using USFederalHolidayCalendar from 'pandas.tseries.holiday', which provided us with data on holidays and then we proceeded to label dates that were not in 'USFederalHolidayCalendar'. Weekday and Weekend features were created similarly. All these features were made to be Boolean, for example, if a date is 12/01/2022, then that date is a Thursday, weekday and not a holiday. This would mean that 'holiday' will be set to 0, 'weekday' will be set to 1 and 'weekend' would be set to 0. Furthermore, a 'month' feature was also created that labeled each date in the range of 1-12, where each number denoted a month.

Weather.csv

The weather dataframe had information about weather for a respective date. This could be used to train on for each date so that the model would be able to predict the trips more effectively. The first step was to determine if any null values were present. The ‘events’ column had very high amount of null values. The other entries were ‘fog’, ‘rain’, ‘fog rain’ and ‘thunderstorm’. This meant that the null values in this column denoted that the day was nither ‘fog’, ‘rain’ and ‘thunderstorm’ or ‘fog rain’. So we proceeded to replace the null values for this column with the value ‘clear_day’. Also, the ‘max_gust_speed’ column seemed to have some null values. Since they were not many in amount, we replaced those values with corresponding values in ‘max_wind_speed’ column. After this, we tried to determine the datatypes for each column. The column precipitation_inches’ was of type string, so we converted it to float, but that lead to new null values due to type conversion loss. So we handled that by simply replacing the null values with ‘0.0’.

The weather column also had respective ‘zip codes’ for the weather data. Since our problem was to predict the bike trips for cities, we split the weather dataframe for each unique value in ‘zip code’ column. The zip codes were 94107 for San Francisco, 94063 for Redwood, 94301 for Palo Alto, 94041 for Mountain View and 95113 for San Jose. We also saw that the weather data for Palo Alto had a lot of null values, so we decided to not predict for the city ‘Palo Alto‘.

Station.csv and Status.csv

We did not use data from these as the features in them were not relevant to our problem.

3. Creating The Training Dataset

For the creation of our training dataset, we merged the two obtained dataframe from weather and trips into one new training dataframe. This was done individually for each city that we were predicting for.

4. Choices of algorithms to train and evaluate

The Following models were trained for comparison:

- Random Forest Regressor
- Extra Trees Regressor
- Gradient Boosting Regressor
- KNN regressor
- XGBoost Regressor
- Lasso Regressor

5. Performance Comparison

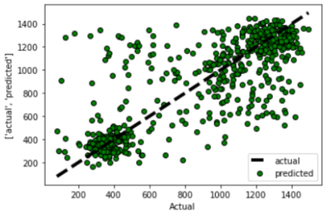
To evaluate the performance of the selected algorithms, RMSE evaluation score was used, along with scatter plots for each of the algorithms. Then the scores were compared and the best performing algorithm was selected for our problem. The training set for this was split into 80% train and 20% test for validation purposes.

6. Running and comparing the algorithms

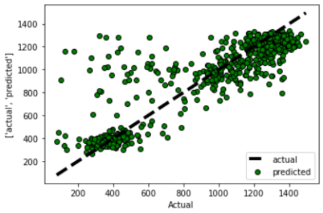
All the algorithms were run on the default parameters on the training set created above. After this, the performance for each model was evaluated on the validation set.

Model Used	RMSE score	Comment
RandomForestRegressor	Mean: 244.9306	Relatively poor performance
ExtraTreesRegessor	Mean:190.8538	Good Performance
XGBoostRegressor	Mean: 190.0768	Good Performance, better than EXT
KNN Regressor	Mean: 452.8310	Worst Performance
LassoRegression	Mean: 228.4479	Relatively Poor Performance
GradientBoosting	Mean: 190.0746	Good Performance

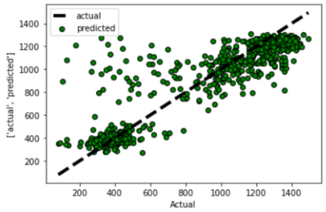
Random Forest Regressor



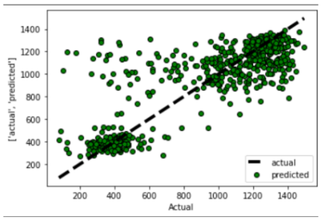
Extra Trees



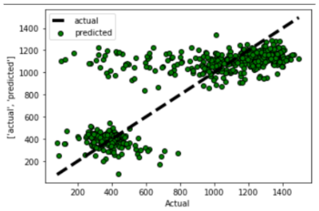
XG Boost



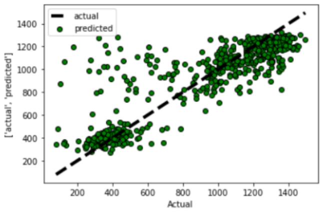
KNN



Lasso



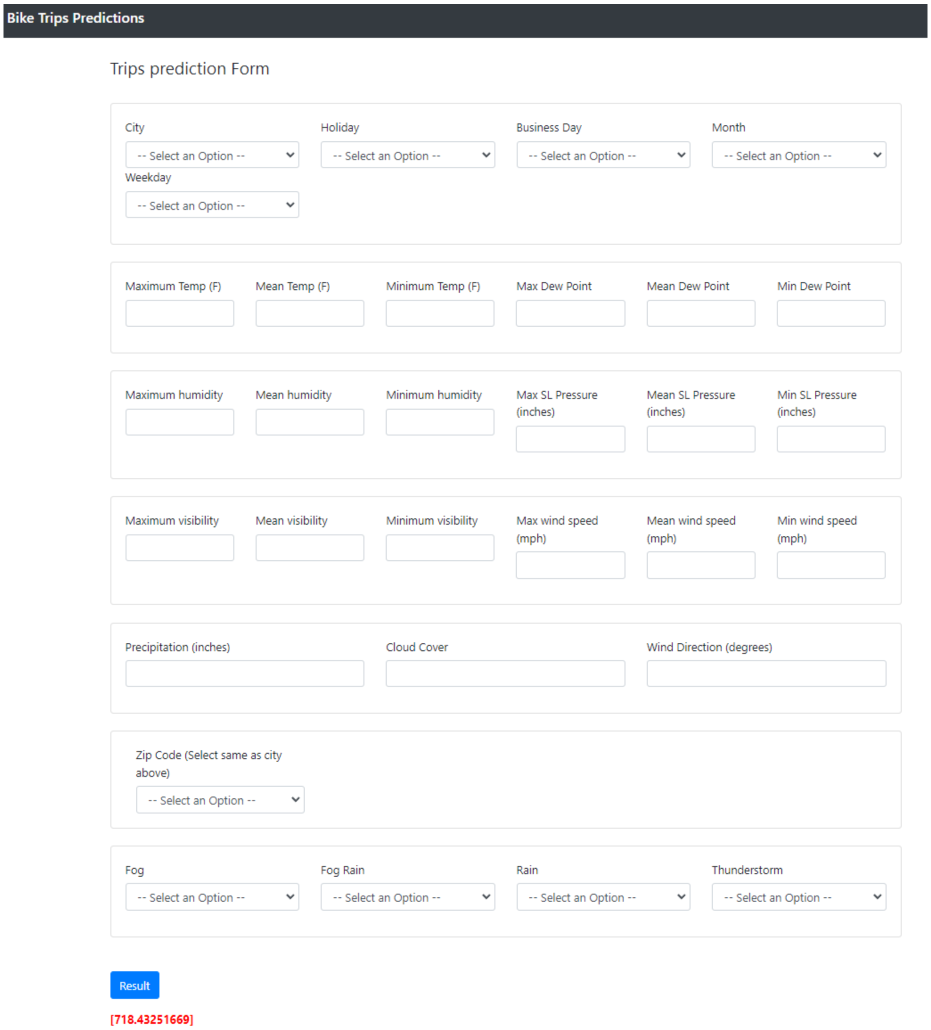
Gradient Boosting



7. Selecting Algorithm for deployment

Gradient Boosting was selected for our problem since it had the best score.

Four different models were trained based on the respective city training data created above, and then integrated into a flask application for predicting of bike trips based on various given data for a day.



Bike Trips Predictions

Trips prediction Form

City: -- Select an Option --
Holiday: -- Select an Option --
Business Day: -- Select an Option --
Month: -- Select an Option --
Weekday: -- Select an Option --

Maximum Temp (F) | Mean Temp (F) | Minimum Temp (F) | Max Dew Point | Mean Dew Point | Min Dew Point

Maximum humidity | Mean humidity | Minimum humidity | Max SL Pressure (inches) | Mean SL Pressure (inches) | Min SL Pressure (inches)

Maximum visibility | Mean visibility | Minimum visibility | Max wind speed (mph) | Mean wind speed (mph) | Min wind speed (mph)

Precipitation (inches) | Cloud Cover | Wind Direction (degrees)

Zip Code (Select same as city above): -- Select an Option --

Fog: -- Select an Option --
Fog Rain: -- Select an Option --
Rain: -- Select an Option --
Thunderstorm: -- Select an Option --

Result

[718.43251669]

Chapter 4: Discussions and Conclusions

1. Decisions Made

- The team discussed and researched about selecting the dataset that met all the criteria necessary for the project implementation.
- After finalizing the dataset, the team decided and narrowed down various tasks that were to be implemented during the project lifecycle.
- The team also made decisions regarding various preprocessing steps and data cleaning strategies to be used such as different data selection techniques, hyper parameter tuning etc. in order to achieve the best possible results.
- Finally, the team discussed and made decisions about various algorithms to be used for each task ranging from various types of regressors to be used for category prediction to achieve better accuracy.

2. Difficulties Faced

- The dataset that the team selected was in multiple files, so we had to extract feature from each file or create new ones in order to create the training dataset.

- The team also faced difficulties in implementing algorithms and obtaining the best possible results considering all the factors such as the complexity of data, where datatypes were different for many columns.

3. Things that worked well and what didn’t

- We were able to successfully create the flask app for bike prediction
- Selection of data extraction and cleaning tasks worked well for us as we initially thought that extracting features and creation of features would vie a daunting task.
- The data for the city “Palo Alto” contained too many null values, so we had to drop the prediction for that city.
- We also wanted to predict the count for each individual bike stations, but after splitting the dataset for each individual bike station, there was not enough data to train and predict the values for stations.

5. Conclusion

- While implementing this project, we learned about a wide array of techniques, algorithms, and different preprocessing tasks involved in data analysis and prediction and how they affect the performance of the algorithms as a whole.
- We learned how to handle multiple files of data and integrate them into one.
- We learned how to apply various algorithms and use predictions models.
- We learned that the output of a model varies based on the requirement of the predictions.

Chapter 5: Project Plan/Task Distribution

Task	**Responsibility**
Dataset Selection	Stavan Patel
Data Exploration and Cleaning	Girish Bisane and
Data Preprocessing	Stavan Patel
Research on Algorithms	Anshul Shandilya
Classification Comparisons	Anshul Shandilya
Creating Flask python app for prediction	Parth Saraiya and Stavan Patel
Documentation and Report	All
PPT	All