# Configuration Manual

MSc Research Project
Programme Name

Pragya Jha
Student ID: x22211047

School of Computing
National College of Ireland

Supervisor: Prof Furqan Rustam

# National College of Ireland
# Project Submission Sheet
# School of Computing

| | |
|---|---|
| **Student Name:** | Pragya Jha |
| **Student ID:** | x22211047 |
| **Programme:** | Programme Name |
| **Year:** | 2024 |
| **Module:** | MSc Research Project |
| **Supervisor:** | Prof Furqan Rustam |
| **Submission Due Date:** | 12/08/2024 |
| **Project Title:** | Configuration Manual |
| **Word Count:** | 956 |
| **Page Count:** | 5 |

I hereby certify that the information contained in this (my submission) is information pertaining to research I conducted for this project. All information other than my own contribution will be fully referenced and listed in the relevant bibliography section at the rear of the project.

**ALL** internet material must be referenced in the bibliography section. Students are required to use the Referencing Standard specified in the report template. To use other author's written or electronic work is illegal (plagiarism) and may result in disciplinary action.

| | |
|---|---|
| **Signature:** | |
| **Date:** | 12th August 2024 |

## PLEASE READ THE FOLLOWING INSTRUCTIONS AND CHECKLIST:

| | |
|---|---|
| Attach a completed copy of this sheet to each project (including multiple copies). | ☐ |
| **Attach a Moodle submission receipt of the online project submission**, to each project (including multiple copies). | ☐ |
| **You must ensure that you retain a HARD COPY of the project**, both for your own reference and in case a project is lost or mislaid. It is not sufficient to keep a copy on computer. | ☐ |

Assignments that are submitted to the Programme Coordinator office must be placed into the assignment box located outside the office.

| **Office Use Only** | |
|---|---|
| Signature: | |
| Date: | |
| Penalty Applied (if applicable): | |

# Configuration Manual

Pragya Jha
x22211047

## 1 Abstract

The primary goal of this research will be to improve the Dublin bike-sharing system to make it a more reliable means of public transport specially to prepare it for future pandemics by applying statistical and machine-learning approaches. Therefore, to compare the results of the modern transfer learning models like Random Forest Regressor, Linear Regression, ARIMA (AutoRegressive Integrated Moving Average), LSTM (Long Short-Term Memory), and the Prophet model and evaluate by using Root Mean Squared Error (RMSE), Mean Absolute Error (MAE), R-squared ($R^2$) and Adjusted R-squared. Chicco et al. (2021)

## 2 System Requirements

- Operating System: Windows, Mac, or Linux.
- Processor: Intel Core i5 8th Gen
- RAM: At least 8GB (16GB preferred).
- Disk Space: Free space that is retrievable and easily accessible, and this should be a minimum of 5GB.
- Internet Connection: During downloading of pre-trained models and
- datasets.

## 3 Software Requirements

- Python Version: Python 3.8 or higher.
- Jupyter Notebook or Google colab.

placeins

## 4 Installation Guide

### 4.1 Python Installation and Version

Install Python and check its version to ensure it is higher than 3.8.

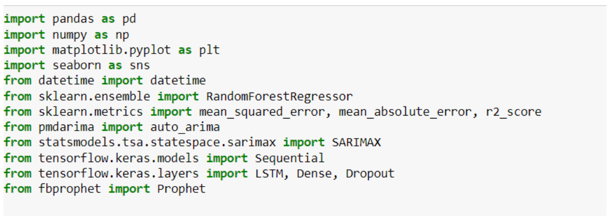
```
!python --version
```

## 4.2 Install Required Packages

To install necessary packages, use pip to download and install them. Example commands:

```
from sklearn.ensemble import RandomForestClassifier
from sklearn.metrics import accuracy_score, classification_report
```

## 4.3 All related libraries can be installed using:



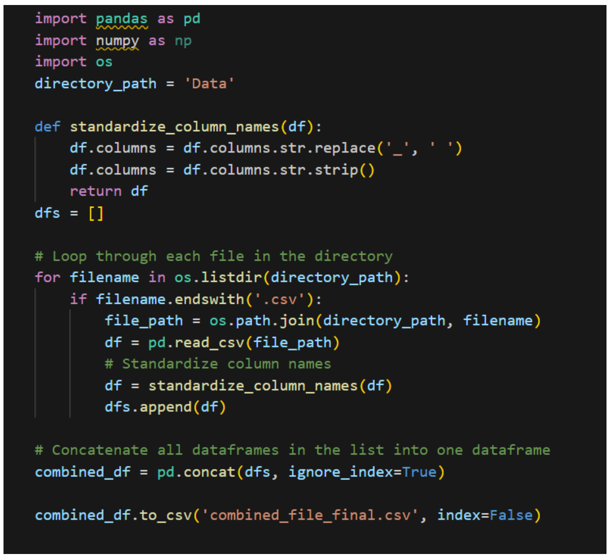
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
from datetime import datetime
from sklearn.ensemble import RandomForestRegressor
from sklearn.metrics import mean_squared_error, mean_absolute_error, r2_score
from pmdarima import auto_arima
from statsmodels.tsa.statespace.sarimax import SARIMAX
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import LSTM, Dense, Dropout
from fbprophet import Prophet
```

Figure 1: Import Commands

# 5 Data Preparation

## 5.1 Data Collection

The next step is to create the Data pipeline. First collect the data , store, and manage the data effectively. This is done by accessing the Dublin Bikes API. This is now used to download historical data covering the period from 2018 to 2024. Store the data in a CSV format within a designated folder structure such as data/raw/ for unprocessed data and data/processed/ for cleaned data.

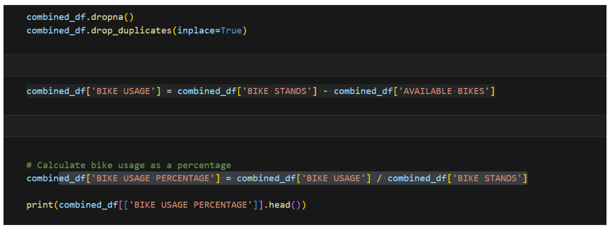


```
import pandas as pd
import numpy as np
import os
directory_path = 'Data'

def standardize_column_names(df):
    df.columns = df.columns.str.replace('_', ' ')
    df.columns = df.columns.str.strip()
    return df
dfs = []

# Loop through each file in the directory
for filename in os.listdir(directory_path):
    if filename.endswith('.csv'):
        file_path = os.path.join(directory_path, filename)
        df = pd.read_csv(file_path)
        # Standardize column names
        df = standardize_column_names(df)
        dfs.append(df)

# Concatenate all dataframes in the list into one dataframe
combined_df = pd.concat(dfs, ignore_index=True)

combined_df.to_csv('combined_file_final.csv', index=False)
```

Figure 2: Data Storing

## 5.2 Exploratory Data Analysis (EDA)

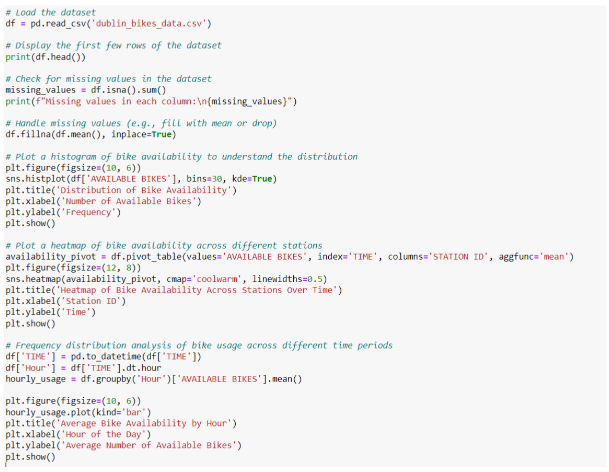
The first step of analyzing the Dublin bike-sharing system dataset would be to view the head of the headset, give an initial understanding of the format, and ensure that all the



```python
combined_df.dropna()
combined_df.drop_duplicates(inplace=True)


combined_df['BIKE USAGE'] = combined_df['BIKE STANDS'] - combined_df['AVAILABLE BIKES']


# Calculate bike usage as a percentage
combined_df['BIKE USAGE PERCENTAGE'] = combined_df['BIKE USAGE'] / combined_df['BIKE STANDS']

print(combined_df[['BIKE USAGE PERCENTAGE']].head())
```

Figure 4: Data Cleaning and feature Engineering

data are combined properly. Check the key variables such as timestamps, station IDs, and bike availability. It's also crucial to check for any missing (NA) values. Then Handle Missing Values, Plot Distribution of Bike Availability, Heatmap of Bike Availability, and then plot frequency Distribution of Bike Usage.



```python
# Load the dataset
df = pd.read_csv('dublin_bikes_data.csv')

# Display the first few rows of the dataset
print(df.head())

# Check for missing values in the dataset
missing_values = df.isna().sum()
print(f"Missing values in each column:\n{missing_values}")

# Handle missing values (e.g., fill with mean or drop)
df.fillna(df.mean(), inplace=True)

# Plot a histogram of bike availability to understand the distribution
plt.figure(figsize=(10, 6))
sns.histplot(df['AVAILABLE BIKES'], bins=30, kde=True)
plt.title('Distribution of Bike Availability')
plt.xlabel('Number of Available Bikes')
plt.ylabel('Frequency')
plt.show()

# Plot a heatmap of bike availability across different stations
availability_pivot = df.pivot_table(values='AVAILABLE BIKES', index='TIME', columns='STATION ID', aggfunc='mean')
plt.figure(figsize=(12, 8))
sns.heatmap(availability_pivot, cmap='coolwarm', linewidths=0.5)
plt.title('Heatmap of Bike Availability Across Stations Over Time')
plt.xlabel('Station ID')
plt.ylabel('Time')
plt.show()

# Frequency distribution analysis of bike usage across different time periods
df['TIME'] = pd.to_datetime(df['TIME'])
df['Hour'] = df['TIME'].dt.hour
hourly_usage = df.groupby('Hour')['AVAILABLE BIKES'].mean()

plt.figure(figsize=(10, 6))
hourly_usage.plot(kind='bar')
plt.title('Average Bike Availability by Hour')
plt.xlabel('Hour of the Day')
plt.ylabel('Average Number of Available Bikes')
plt.show()
```

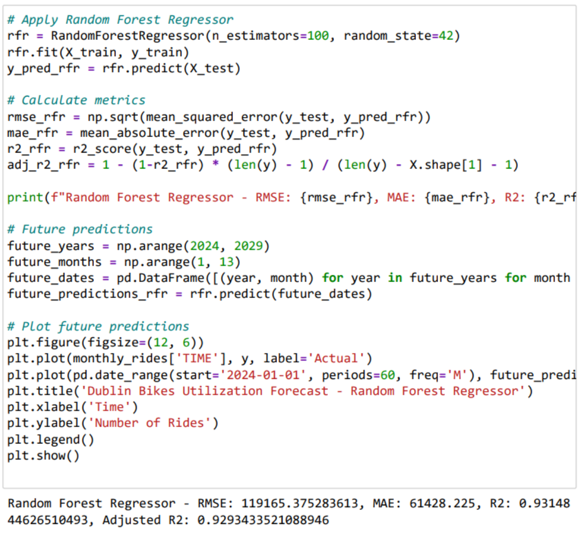
Figure 3: Exploratory Data Analysis

## 5.3 Data Preprocessing and Feature Engineering:

This step ensured that the data was clean, consistent, and ready for analysis. The data is loaded the raw data into a Pandas DataFrame. This will allow for easy manipulation and analysis of the data. During preprocessing the missing values are handled either filling by them using imputation techniques or dropping rows/columns that are extensively incomplete.Feature engineering here involves creating two new variables that enhance the predictive power of the models they are BIKEUSAGE and a BIKEUSAGEPERCENTAGE.

# 6 Model Implementation:

This step involves applying various predictive models to the processed data to forecast bike availability in the Dublin bike-sharing system. The Random Forest model, known for its robustness against overfitting and ability to handle large datasets Falamarzi et al. (2018), was trained on historical data, with feature importance assessed to identify key variables. Linear Regression was used as a baseline model to predict bike availability, assuming a linear relationship between time and availability. ARIMA and its seasonal extension, SARIMA, were employed for time series forecasting, capturing temporal dependencies and seasonality in the data. The LSTM model, leveraging its capacity to learn long-term dependencies, was implemented using TensorFlow/Keras, with normalized data and sequential input features. Lastly, the Prophet model, designed to handle missing data and outliers, was applied to decompose the time series into trend, seasonality, and holiday components. Each model's performance was evaluated using metrics such as RMSE, MAE, and R-squared, ensuring accurate predictions and effective analysis of bike availability patterns.



```
# Apply Random Forest Regressor
rfr = RandomForestRegressor(n_estimators=100, random_state=42)
rfr.fit(X_train, y_train)
y_pred_rfr = rfr.predict(X_test)

# Calculate metrics
rmse_rfr = np.sqrt(mean_squared_error(y_test, y_pred_rfr))
mae_rfr = mean_absolute_error(y_test, y_pred_rfr)
r2_rfr = r2_score(y_test, y_pred_rfr)
adj_r2_rfr = 1 - (1-r2_rfr) * (len(y) - 1) / (len(y) - X.shape[1] - 1)

print(f"Random Forest Regressor - RMSE: {rmse_rfr}, MAE: {mae_rfr}, R2: {r2_rf

# Future predictions
future_years = np.arange(2024, 2029)
future_months = np.arange(1, 13)
future_dates = pd.DataFrame([(year, month) for year in future_years for month
future_predictions_rfr = rfr.predict(future_dates)

# Plot future predictions
plt.figure(figsize=(12, 6))
plt.plot(monthly_rides['TIME'], y, label='Actual')
plt.plot(pd.date_range(start='2024-01-01', periods=60, freq='M'), future_predi
plt.title('Dublin Bikes Utilization Forecast - Random Forest Regressor')
plt.xlabel('Time')
plt.ylabel('Number of Rides')
plt.legend()
plt.show()
```

```
Random Forest Regressor - RMSE: 119165.375283613, MAE: 61428.225, R2: 0.93148
44626510493, Adjusted R2: 0.9293433521088946
```

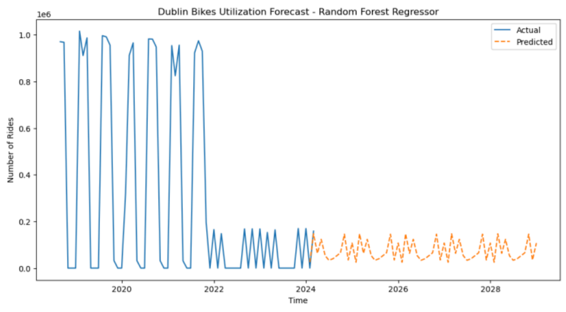
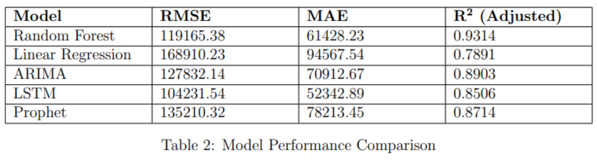


Figure 5: Random Forest Model

## 7 Model Evaluation

The performance of the model is then evaluated using several key metrics, including Root Mean Squared Error (RMSE), Mean Absolute Error (MAE), and R-squared ($R^2$) as mentioned in Chicco et al. (2021). The RMSE helps understand the accuracy of the model by measuring the square root of the average squared differences between the predicted and actual values. Whereas, MAE measures the average magnitude of errors in predictions. This acts as a very straightforward assessment of prediction accuracy. R-squared was also calculated and it determined the proportion of variance in bike availability that could be explained by the model. The model was put through extensive testing by making predictions on unseen test data.The model demonstrated strong performance, evidenced by a low RMSE value, indicating its effectiveness in minimizing large errors.



| Model | RMSE | MAE | $R^2$ (Adjusted) |
|---|---|---|---|
| Random Forest | 119165.38 | 61428.23 | 0.9314 |
| Linear Regression | 168910.23 | 94567.54 | 0.7891 |
| ARIMA | 127832.14 | 70912.67 | 0.8903 |
| LSTM | 104231.54 | 52342.89 | 0.8506 |
| Prophet | 135210.32 | 78213.45 | 0.8714 |

Table 2: Model Performance Comparison

Figure 6: ACCURACY ANALYSIS

## 8 Execution of the Code

- Download the Dataset
- Unzip the Files into a Folder
- Open the Python File in Google Colab or Jupyter Notebook
- Change the Path to Refer to the Dataset Location
- Run the Code

## 9 Conclusion

This study concludes that the Random Forest Regressor outperforms other models in predicting bike availability within Dublin's bike-sharing system. This is especially noted during the volatile periods of the COVID-19 pandemic. The results of this analysis suggest that if we implement these predictions, it can enhance the resilience of the system which would be very important in a time of future pandemic. If we follow this there is a potential that even in case of a public disaster we might never have to go into a full lockdown. The author also proposes future research that could integrate real-time data, expand the scope of variables, and develop a comprehensive multimodal transportation strategy. This could further enhance the system's adaptability and user satisfaction.

## References

Chicco, D., Warrens, M. J. and Jurman, G. (2021). The coefficient of determination r-squared is more informative than smape, mae, mape, mse and rmse in regression analysis evaluation, *PeerJ Computer Science* **7**: e623.
**URL:** *https://doi.org/10.7717/peerj-cs.623*

Falamarzi, A., Moridpour, S., Nazem, M. and Cheraghi, S. (2018). Development of random forests regression model to predict track degradation index: Melbourne case study.