Architecture

Architecture

BIKE SHARE PREDICTION

Revision Number – 2.0

Last Date of Revision – 28/03/2022

Biswajit Rajaguru Mohapatra

1

Architecture

Document Version Control

|  |  |  |  |
| --- | --- | --- | --- |
| Data Issued | Version | Description | Author |
| 28/02/2022 | 01 | Abstract, Introduction, Problem Statement | Biswajit |
| 28/03/2022 | 03 | Design Flow | Biswajit |
| 28/03/2022 | 03 | Performance Evaluation | Biswajit |

2

ARCHITECTURE

**Contents**

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| [**Abstract**](https://docs.google.com/document/d/14Wck6wDTC8bscpCxNjrSRu01-UC_tJ5Y/edit#heading=h.gjdgxs) | | | | | | | | | | | | | | | [4](https://docs.google.com/document/d/14Wck6wDTC8bscpCxNjrSRu01-UC_tJ5Y/edit#heading=h.gjdgxs) | | |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| [**Introduction**](https://docs.google.com/document/d/14Wck6wDTC8bscpCxNjrSRu01-UC_tJ5Y/edit#heading=h.30j0zll) | | |  | | | | | | | | | | | | [4](https://docs.google.com/document/d/14Wck6wDTC8bscpCxNjrSRu01-UC_tJ5Y/edit#heading=h.30j0zll) | | | |
|  |  | |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  | [**Why this Architecture Design documentation?**](https://docs.google.com/document/d/14Wck6wDTC8bscpCxNjrSRu01-UC_tJ5Y/edit#heading=h.1fob9te) | | | | | | | | | | | | | |  | | [4](https://docs.google.com/document/d/14Wck6wDTC8bscpCxNjrSRu01-UC_tJ5Y/edit#heading=h.1fob9te) |  |
| [**1 Architecture**](https://docs.google.com/document/d/14Wck6wDTC8bscpCxNjrSRu01-UC_tJ5Y/edit#heading=h.3znysh7) | | | |  | | | | | | | | | | | [4](https://docs.google.com/document/d/14Wck6wDTC8bscpCxNjrSRu01-UC_tJ5Y/edit#heading=h.3znysh7) | | | |
|  | | | |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| [**2 Architecture design**](https://docs.google.com/document/d/14Wck6wDTC8bscpCxNjrSRu01-UC_tJ5Y/edit#heading=h.3dy6vkm) | | | | | | | |  | | | | | | | [5](https://docs.google.com/document/d/14Wck6wDTC8bscpCxNjrSRu01-UC_tJ5Y/edit#heading=h.3dy6vkm) | | |  |
|  |  | | | |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  | [**2.1 Data gathering from main source**](https://docs.google.com/document/d/14Wck6wDTC8bscpCxNjrSRu01-UC_tJ5Y/edit#heading=h.1t3h5sf) | | | | | | | | | | | | | | [5](https://docs.google.com/document/d/14Wck6wDTC8bscpCxNjrSRu01-UC_tJ5Y/edit#heading=h.1t3h5sf) | | |  |
|  |  | | | |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  | [**2.2 Data description**](https://docs.google.com/document/d/14Wck6wDTC8bscpCxNjrSRu01-UC_tJ5Y/edit#heading=h.4d34og8) | | | | | | | | | | | | |  | [5](https://docs.google.com/document/d/14Wck6wDTC8bscpCxNjrSRu01-UC_tJ5Y/edit#heading=h.4d34og8) | | |  |
|  |  | | | |  |  |  |  |  |  |  |  |  | |  |  |  |  |
|  |  | | | |  |  |  |  | |  |  |  | | |  |  |  |  |
|  | [**2.3 Data pre-processing**](https://docs.google.com/document/d/14Wck6wDTC8bscpCxNjrSRu01-UC_tJ5Y/edit#heading=h.3rdcrjn) | | | | | | | | | | |  | | |  | [5](https://docs.google.com/document/d/14Wck6wDTC8bscpCxNjrSRu01-UC_tJ5Y/edit#heading=h.3rdcrjn) | |  |
|  |  | | | |  |  |  |  | |  |  | | | |  | |  |  |
|  | [**2.4 Modelling**](https://docs.google.com/document/d/14Wck6wDTC8bscpCxNjrSRu01-UC_tJ5Y/edit#heading=h.26in1rg) | | | |  | | | | |  | | | | |  | | [6](https://docs.google.com/document/d/14Wck6wDTC8bscpCxNjrSRu01-UC_tJ5Y/edit#heading=h.26in1rg) | |
|  |  | | | |  |  |  |  | | |  | | | |  | |  |  |
|  | [**2.5 UI integration**](https://docs.google.com/document/d/14Wck6wDTC8bscpCxNjrSRu01-UC_tJ5Y/edit#heading=h.lnxbz9) | | | | |  | | | | | | | | | [6](https://docs.google.com/document/d/14Wck6wDTC8bscpCxNjrSRu01-UC_tJ5Y/edit#heading=h.lnxbz9) | | |  |
|  |  | | | | |  |  |  | | |  | | | |  | |  |  |
|  | [**2.6Data from user**](https://docs.google.com/document/d/14Wck6wDTC8bscpCxNjrSRu01-UC_tJ5Y/edit#heading=h.35nkun2) | | | | | | | | | | | | | | [6](https://docs.google.com/document/d/14Wck6wDTC8bscpCxNjrSRu01-UC_tJ5Y/edit#heading=h.35nkun2) | | |  |
|  |  | | | | |  |  |  | | |  | | | |  | |  |  |
|  | [**2.7 Data validation**](https://docs.google.com/document/d/14Wck6wDTC8bscpCxNjrSRu01-UC_tJ5Y/edit#heading=h.1ksv4uv) | | | | | | |  | | | | | | | [6](https://docs.google.com/document/d/14Wck6wDTC8bscpCxNjrSRu01-UC_tJ5Y/edit#heading=h.1ksv4uv) | | |  |
|  |  | | | | |  | |  | | |  | | | |  | |  |  |
|  | [**2.8 Rendering the results**](https://docs.google.com/document/d/14Wck6wDTC8bscpCxNjrSRu01-UC_tJ5Y/edit#heading=h.44sinio) | | | | | | | | | |  | | | | [6](https://docs.google.com/document/d/14Wck6wDTC8bscpCxNjrSRu01-UC_tJ5Y/edit#heading=h.44sinio) | | |  |
|  | [**2.9 Deployment**](https://docs.google.com/document/d/14Wck6wDTC8bscpCxNjrSRu01-UC_tJ5Y/edit#heading=h.2jxsxqh) | | | | | | | | | |  | | | |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |

ARCHITECTURE

Abstract

Bike sharing systems are a new generation of traditional bike rentals where the whole

process from membership, rental and return back has become automatic. Through these

systems, users are able to easily rent a bike from a particular position and return back at

another position. Currently, there are about over 500 bike-sharing programs around the

world which is composed of over 500 thousand bicycles. Today, there exists great interest in

these systems due to their important role in traffic, environmental and health issues. Apart

from interesting real-world applications of bike sharing systems, the characteristics of data

being generated by these systems make them attractive for the research.

Introduction

Why this Architecture Design documentation?

The main objective of the Architecture design documentation is to provide the internal logic understanding of the flight fare prediction code. The Architecture design documentation is designed in such a way that the programmer can directly code after reading each module description in the documentation.

1 Architecture

**START**

**DATA COLLECTION**

**DATA VALIDATION**

**DATA PREPROCESSING**

**Model Building**

**TEST MODEL**

**CLOUD SETUP**

**PUSHING APP TO CLOUD**

**START APPLICATION**

**RECIVE DATA FROM USER/CLIENT**

**DATA VALIDATION**

**MODEL CALL**

**PREDICT THE OUTCOME**

**SAVE THE PRDICTION File**

**END**

2 Architecture design

This project is to create an interface for the user to know their approximate flight ticket price, in addition to this, in need of getting the real time project experience we are importing the gathered data into our own database and then start the project from the scratch.

2.1 Data gathering from main source

The data for the current project is being gathered from Kaggle dataset, the link to the data is: <https://archive.ics.uci.edu/ml/datasets/Bike+Sharing+Dataset>

2.2 Data description

We have two sets of datasets; one is for hourly basis and another is for day basis. Both of the datasets contain the weather conditions such as humidity, temperature, air temperature and windspeed essential factor to determine and conclude the weather outside of the particular place. These variables in the datasets are our prime data for prediction. The datasets also contain some variables columns such as instant, year, dteday, casual and registered which doesn’t contribute to our prediction. The datasets also hold the information about month, season, holiday, weekday, working day and weather outside.

2.3 Data pre-processing

Steps performed in pre-processing are:

* First the data types are being checked and found only the price column is of type integer.
* Checked for null values as there are few null values, those rows are dropped.
* Converted all the required column into the date time format.
* Performed one-hot encoding for the required columns.
* Scaling is performed for required data.
* And, the data is ready for passing to the machine learning algorithm.

2.4 Modelling

The pre-processed data is then visualized and all the required insights are being drawn. Although from the drawn insights, the data is randomly spread but still modelling is performed with different machine learning algorithms to make sure we cover all the possibilities. And finally, as expected random forest regression performed well and further hyperparameter tuning is done to increase the model’s accuracy.

2.5 UI integration

Both CSS and HTML files are being created and are being integrated with the created machine learning model. All the required files are then integrated to the app.py file and tested locally.

2.6 Data from user

The data from the user is retrieved from the created HTML web page.

2.7 Data validation

The data provided by the user is then being processed by app.py file and validated. The validated data is then sent for the prediction.

2.8 Rendering the results

The data sent for the prediction is then rendered to the web page.

2.9 Deployment

The tested model is then deployed to Heroku. So, users can access the project from any internet devices.