Architecture Design
Bike Share Prediction

Revision Number: 1.1

Last Date of Revision: 18/3/2024

Sameer Singh

Contents

1. 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 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.

The most important problem from a business point of view for bike-sharing systems like Capital Bikeshare (one of the U.S.A.'s largest bicycle sharing systems) is to predict the bike demand on any particular day. There is a possibility that bike stations can be full or empty when a traveler comes to the station. While having excess bikes results in wastage of resources (bike maintenance and land/bike stand required for parking and security), having fewer bikes leads to revenue loss (ranging from a short term loss due to missing out on immediate customers to potential longer term loss due to loss in future customer base). Thus having an estimate on the demands would enable efficient functioning of this company Capital Bikeshare. And to predict the use of such a system can be helpful for the users to plan their travels and also for the Capital Bikeshare entrepreneurs to set up the system properly.

2. Introduction

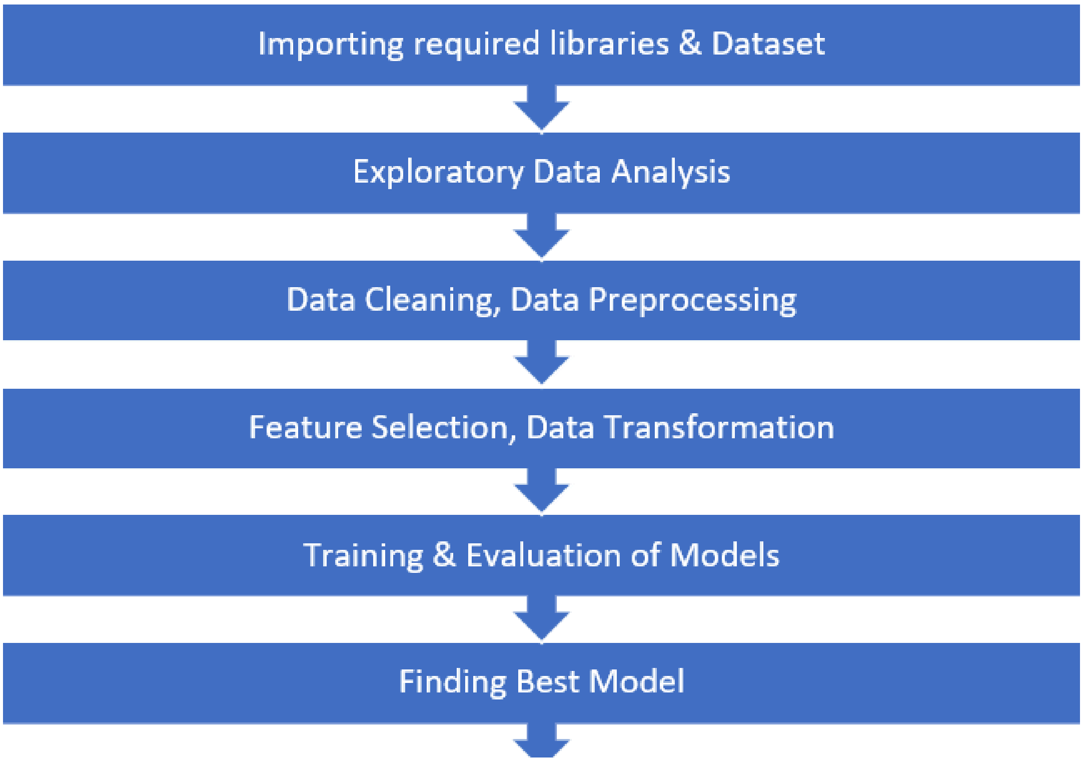
2.1 why this architecture design document?

The main objective of the Architecture design documentation is to provide the internal logic understanding of the Rental Bike share demand 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.

2.2 Scope

Architecture Design document is an architecture design process that follows a step-by-step refinement process. The process can be used for designing data structures, required software architecture, source code and ultimately, performance algorithms. Overall the design principles may be defined during requirement analysis and then refined during architectural design work.

3. Architecture



Importing required libraries & Dataset

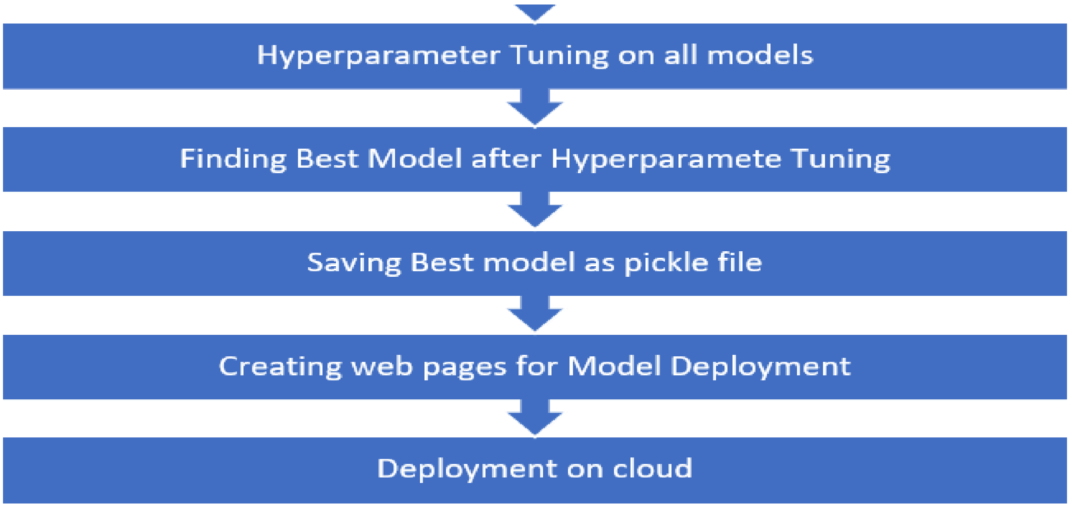
Exploratory Data Analysis

Data Cleaning, Data Preprocessing

Feature Selection, Data Transformation

Training & Evaluation of Models

Finding Best Model


Hyperparameter Tuning on all models
Finding Best Model after Hyperparamete Tuning
Saving Best model as pickle file
Creating web pages for Model Deployment
Deployment on cloud

4. Architecture Design

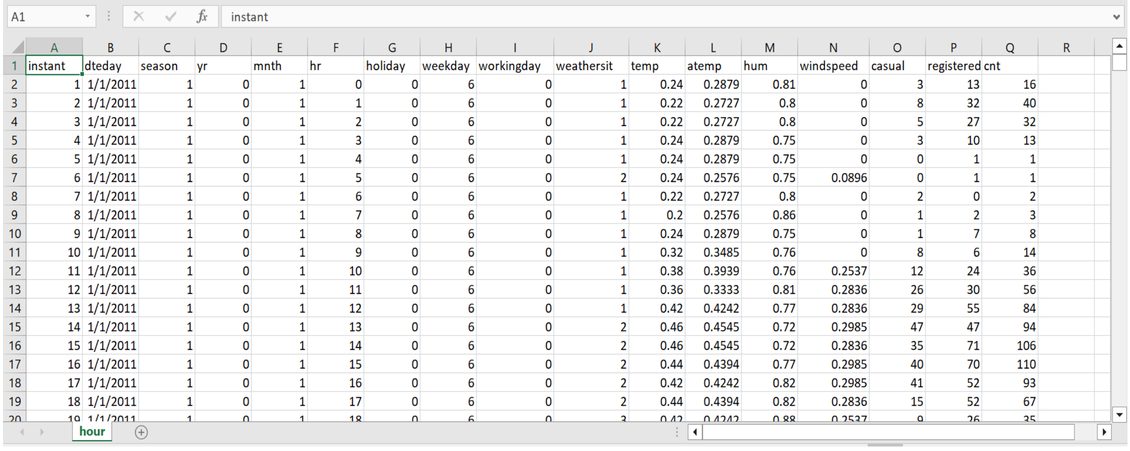
4.1 Data Collection

The dataset was taken from the UCI Machine Learning Repository. Dataset hour.csv is available in this link https://archive.ics.uci.edu/dataset/275/bike+sharing+dataset.

4.2 Data Description

Rental Bikeshare Demand Prediction is 17K+ dataset publicly available on UCI Repository. The information in the dataset is present in a csv file named hour.csv. Dataset contains 17379 rows which shows the information such as index instant, Dateday, season, year, month, hour, holiday, weekday, workingday, weathersit, temperature, atemperature, humidity, windspeed, casual, registered, count.

The glance of the Dataset is :



	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q
1	instant	dteday	season	yr	mnth	hr	holiday	weekday	workingday	weathersit	temp	atemp	hum	windspeed	casual	registered	cnt
2	1	1/1/2011	1	0	1	0	0	6	0	1	0.24	0.2879	0.81	0	3	13	16
3	2	1/1/2011	1	0	1	1	0	6	0	1	0.22	0.2727	0.8	0	8	32	40
4	3	1/1/2011	1	0	1	2	0	6	0	1	0.22	0.2727	0.8	0	5	27	32
5	4	1/1/2011	1	0	1	3	0	6	0	1	0.24	0.2879	0.75	0	3	10	13
6	5	1/1/2011	1	0	1	4	0	6	0	1	0.24	0.2879	0.75	0	0	1	1
7	6	1/1/2011	1	0	1	5	0	6	0	2	0.24	0.2576	0.75	0.0896	0	1	1
8	7	1/1/2011	1	0	1	6	0	6	0	1	0.22	0.2727	0.8	0	2	0	2
9	8	1/1/2011	1	0	1	7	0	6	0	1	0.2	0.2576	0.86	0	1	2	3
10	9	1/1/2011	1	0	1	8	0	6	0	1	0.24	0.2879	0.75	0	1	7	8
11	10	1/1/2011	1	0	1	9	0	6	0	1	0.32	0.3485	0.76	0	8	6	14
12	11	1/1/2011	1	0	1	10	0	6	0	1	0.38	0.3939	0.76	0.2537	12	24	36
13	12	1/1/2011	1	0	1	11	0	6	0	1	0.36	0.3333	0.81	0.2836	26	30	56
14	13	1/1/2011	1	0	1	12	0	6	0	1	0.42	0.4242	0.77	0.2836	29	55	84
15	14	1/1/2011	1	0	1	13	0	6	0	2	0.46	0.4545	0.72	0.2985	47	47	94
16	15	1/1/2011	1	0	1	14	0	6	0	2	0.46	0.4545	0.72	0.2836	35	71	106
17	16	1/1/2011	1	0	1	15	0	6	0	2	0.44	0.4394	0.77	0.2985	40	70	110
18	17	1/1/2011	1	0	1	16	0	6	0	2	0.42	0.4242	0.82	0.2985	41	52	93
19	18	1/1/2011	1	0	1	17	0	6	0	2	0.44	0.4394	0.82	0.2836	15	52	67
20	19	1/1/2011	1	0	1	18	0	6	0	3	0.42	0.4242	0.88	0.2537	9	26	35

4.3 Loading dataset

Local file system was used for loading the dataset (hour.csv) using read_csv function of Pandas Library.

4.4 Data Preprocessing

• All the necessary libraries were imported first such as Numpy, Pandas, Matplotlib, Seaborn, and sk-learn.

• Checking the basic profile of the dataset. To get a better understanding of the dataset.

  ○ Using Info method
  ○ Using Describe method
  ○ Checking for unique values of each column.

• Checking for info of the Dataset, to verify the correct datatype of the Columns.

• Checking for Null values, because the null values can affect the accuracy of the model.

• Doing Feature selection and dropping those columns which are not needed.

• Converting the integer columns [‘season’, ‘yr’, ‘mnth’, ‘hr’, ‘holiday’, ‘weekday’, ‘workingday’, ‘weathersit’] into categorical columns.

• Performing standard scaling on numerical and categorical columns. And, One Hot Encoding on Categorical columns.

Now, the info is prepared to train a Machine Learning Model.

3.5 Modeling Process

● After preprocessing the data, the data will be split into 2 sets X and y. X contains all the columns except the target column in our case (Count), y contains only the Target column.

● Using train test split we first split the dataset into X_train,X_test, y_train, y_test .

● Then use following Regression Algorithms like: CatBoost, Random Forest, Bagging, Decision Tree, Gradient Boost, KNeighbors, Linear Regression to predict the count of Rental Bikes required based on certain conditions.

● After creating the following normal without any hyper tuned models, find the best performing model out of all.
● Then use Grid Search CV to find the best suiting parameters for all the models and again find out the best performing model after Hyperparameter tuning.

3.6 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.

3.7 Data from User

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

3.8 Data Validation

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

3.9 Rendering the Results

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

3.10 Deployment

The tested model is then deployed to railway.app (a deployment platform like Heroku). So, users can access the project from any internet device.