**Architecture Design**

**STORE SALES PREDICTION**

|  |  |
| --- | --- |
| Written By | Mohammad Salman |
| Document Version | 1.0 |
| Last Revised Date | 22-11-2022 |

**Document Control**

Change Record:

|  |  |  |  |
| --- | --- | --- | --- |
| **Version** | **Date** | **Author** | **Comments** |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

**Approval Status:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Version** | **Review date** | **Reviewed By** | **Approved By** | **Comments** |
|  |  |  |  |  |

**Index**

|  |  |
| --- | --- |
| **Content** | **Page No** |
| Abstract | 4 |
| 1. Introduction | 4 |
| 1.1 What is Architecture Design? | 4 |
| 1.2 Scope | 4 |
| 1.3 Constraints | 4 |
| 2. Technical Specification | 5 |
| 2.1 Dataset | 5 |
| 2.2 Logging | 7 |
| 2.3 DataBase | 7 |
| 2.4 Deployment | 7 |
| 3. Technology Stack | 8 |
| 4. Proposed Solution | 8 |
| 5. Architecture | 8 |
| 5.1 Architecture Description | 9 |
| 6. Model Training/Validation Workflow | 11 |
| 7. User Input/Output Workflow | 12 |

# Abstract

Currently Rental bikes are introduced in many urban cities for the enhancement of mobility comfort. It is important to make the rental bike available and accessible to the public at the right time as it lessens the waiting time. Eventually, providing the city with a stable supply of rental bikes becomes a major concern. The crucial part is the prediction of bike count required at each hour for the stable supply of rental bikes.

# 1. Introduction

**1.1 What is Architecture Design?**

The goal of Architecture Design (AD) or a low-level design document is to give the internal design of the actual program code for the `Bike Share Prediction System`. AD describes the class diagrams with the methods and relation between classes and program specification. It describes the modules so that the programmer can directly code the program from the document.

## 1.2 Scope

Architecture Design(AD) is a component-level design process that follows a step-by-step refinement process. This process can be used for designing data structures, required software, architecture, source code, and ultimately, performance algorithms. Overall, the data organization may be defined during requirement analysis and then refined during data design work. And the complete workflow.

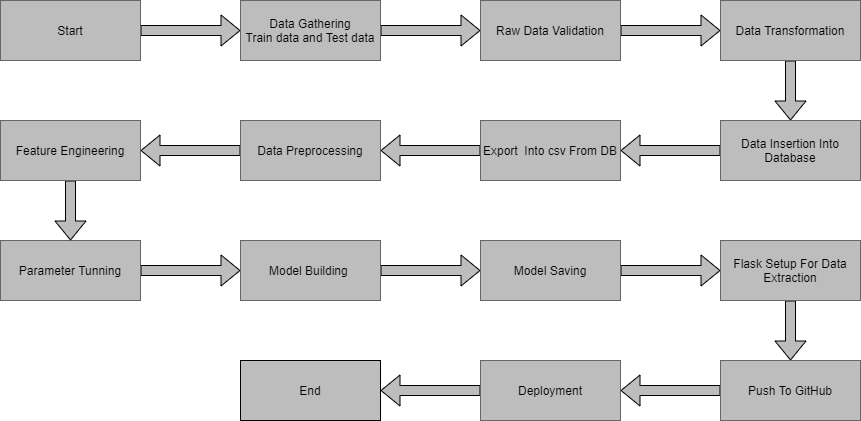
## 1.3 Constraints

We only predict the expected casual and registered customers based on the weather condition and date information.

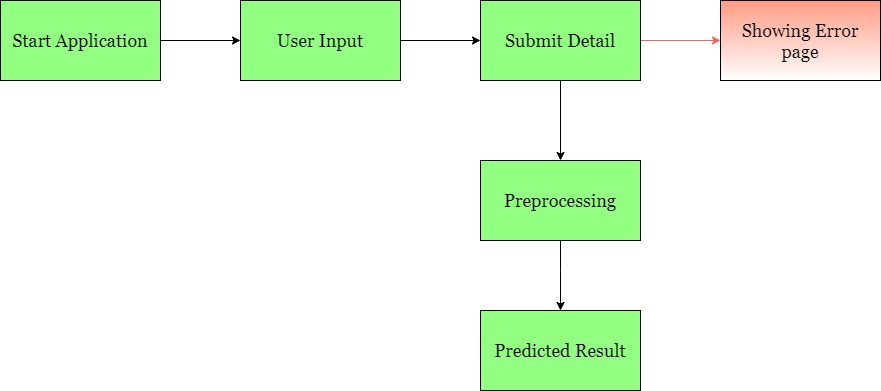
# 2. Technical Specification

## 2.1 Process Flow

We will be using following process flow for this project. The process will be based on modular coding i.e. use of oops concepts to build the entire project from start to end.



## 2.2 Deployment Process



## 2.3 Error Handling

If any error occurred in the processing way then the error message should be shown to the user in a completely non-technical way that can be understandable by any person. And Meaningful error message should be shown, so the user can spot his mistake and rerun the process with improvement. All the errors that are will occur should be handled properly. And we have to log every error for our application and have to manage the same.

## 2.4. Performance

The Rental bike Prediction is dependent on machine-learning algorithms. We will train various ml algorithms and will find the best fitting algorithm for predicting the target. Our system performance will be based on the data we are going to feed to the algorithms. And the performance will depend on the finalized model. and the web application and the deployment server. With all of these components, our program should run properly.

### 2.5 Reusability

The code and the module are created during the time of building the project should maintain all coding guidelines and full project code is written in a Modular fashion. Our system should have the flexibility to work properly from any location. And it should handle any improper input value from the user and should give a meaningful error message so the user can correct his/her mistake and enter valid input to get the result. And the system should be reusable in every manner with different types of inputs values that are all are it has been trained.

### 2.6 Application Compatibility

The different libraries and python programming languages are used to build the system. Every library has its own functionality and it should work properly with our fluctuate system. Flask will be used for making the web APIs and HTML/CSS will be used to make the web application. All the components of the application should work properly and it should produce a result without any interpretation.

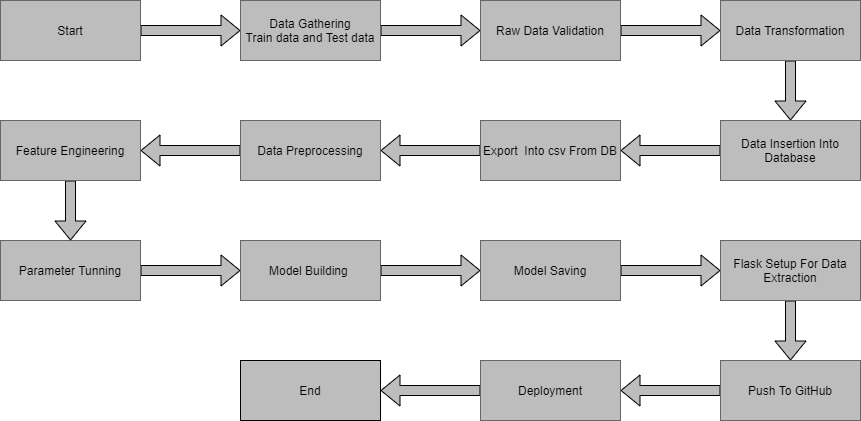
### 2.7 Resource Utilization

Our application should utilize the given resource properly and it should use a minimal amount of internet to work and call the APIs on the Web page. Our system should not use much amount of computational resources hence it will make the application slow. Our application will be deployed cloud platform and it should utilize the resource given on the cloud and work properly.

## 2.8. Deployment

For the deployment process, we will using Heroku cloud platforms for hosting our application. The cloud platform will run the system and it will give the flexibility to use our application globally.





### 5.1 Data Gathering

Data source:  <https://www.google.com/url?q=https://archive.ics.uci.edu/ml/datasets/Bike%2BSharing%2BDataset&sa=D&source=apps-viewer-frontend&ust=1669523611018571&usg=AOvVaw2w2CNen9aLPM9FfBoI2J16&hl=en>Train and Test data are stored in .csv format.

### 5.2 Raw Data Validation

After data is loaded, various types of validation are required before we proceed further with any operation. Validations like checking for zero standard deviation for all the columns, checking for complete missing values in any columns, etc. These are required because The attributes which contain these are of no use. It will not play role in contributing to the sales of an item from respective outlets.

Like if any attribute is having zero standard deviation, it means that’s all the values are the same, its mean is zero. This indicates that either the sale is increasing or decrease that attribute will remain the same. Similarly, if any attribute is having full missing values, then there is no use in taking that attribute into an account for operation. It’s unnecessary increasing the chances of dimensionality curse.

### 5.3 Data Transformation

Before sending the data into the database, data transformation is required so that data are converted into such form with which it can easily insert into the database.

### 5.5 Export as `CSV` from Database

From the database both the train and test data set are exported into the local system and stored into CSV files. Now this CSV file will have proceeded for further processing.

### 5.6 Data Preprocessing

* Season: Month column has a direct mapping with season (Winter: January to March, Summer: April to June, Fall: July to September and Spring: October to December). Hence we will drop season column
* Holiday and 'day': workingday = weekday and not a holiday. Since we noticed that there were two kinds of bike rental behavoirs - during working days and not a working day, we will retain only the workingday column and drop 'day' and 'holiday' column
* Workingday: After observing the bike rental trend, we propose to build 2 separate models for 1. if it is a working day, and 2. if it is a non-working day. Hence, we can separate out the data based on this column and drop the column
* Weather: Split weather column to weather\_1, weather\_2 and weather 3 (recall that we had relabelled all the weather = 4 data points to weather = 3 due to its sparseness). Drop weather\_3 since it is a function of the rest of the weather columns
* Temp: temp and atemp are highly correlated. Hence retain only the temp column
* Windspeed: Very poorly correlated with count. Hence drop this column
* Casual and registered: These are individual components of our to be predicted column (count). Hence drop these columns
* Month: Split month column to month\_1, month\_2, ..., month\_12. Drop month\_12 since it is a function of the rest of the month columns
* Date: Intuitively, there is should be no dependency on date. Hence drop this column
* Hour: Split hour column to hour\_0, hour\_1, ..., hour\_23. Drop hour\_23 since it is a function of the rest of the hour columns

### 5.7 Feature Engineering

After preprocessing it was found that some of the attributes are not important to the item sales for the particular outlet. So those attributes are removed. Even one hot encoding is also performed to convert the categorical features into numerical features.

### 5.8 Parameter Tuning

Parameters are tuned using Randomized searchCV. Four algorithms are used in this problem, Linear Regression, Gradient boost, Random Forest, and XGBoost regressor. The parameters of all these 4 algorithms are tunned and passed into the model.

### 5.9 Model Building

After doing all kinds of preprocessing operations mention above and performing scaling and hyperparameter tuning, the data set is passed into all four models, Linear Regression, Gradient boost, Random Forest, and XGBoost regressor. It was found that Gradient boost performs best with the smallest RMSE value i.e. 587.0 and the highest R2 score equals 0.55. So ‘Gradient boost’ performed well in this problem.

**5.10 Model Saving**

Model is saved using pickle library in `.pkl` format.

### 5.11 Flask Setup for Data Extraction

After saving the model, the API building process started using Flask. Web application creation was created here. Whatever the data user will enter and then that data will be extracted by the model to predict the prediction of sales, this is performed in this stage.

**5.12 GitHub**

The whole project directory will be pushed into the GitHub repository.

### 5.13 Deployment

The cloud environment was set up and the project was deployed from GitHub into the Heroku cloud platform.

App link- [https://bike-new-share.herokuapp.com](https://bike-new-share.herokuapp.com/)

**7. User Input / Output Workflow.**

