**Capstone Project Submission**

**Instructions:**

i) Please fill in all the required information.

ii) Avoid grammatical errors.

|  |
| --- |
| **Team Member’s Name, Email and Contribution:** |
| **1. Mayur Choudhari (cmayur842@gmail.com)**  1.1 Data Collection and Understanding  1.2 Data Wrangling and Feature Engineering  1.3 Exploratory Data Analysis  1.3.1 Relation of rented bike count with categorical features  1.3.2 Conclusion  1.3.3 Bike Rent Trend according to hour in different scenarios  1.3.4 Distribution of target variable- Bike Rent Count  1.4 Model Selection and Evaluation  1.4.1 Linear Regression  1.4.2 Lasso regression (regularized regression)  1.4.3 Ridge Regression (regularized regression)  1.4.4 Decision Tree regression.  1.4.5 Random forest regression  1.4.6 Gradient Boosting regression.  1.5 Observations.  1.6 Conclusion.  1.7 Preparation of data for model building. |
| **Please paste the GitHub Repo link.** |
| GitHub Link:-   * Mayur Choudhari- https://github.com/Cmayur842/Mayurchoudhari-ML-CapstoneProject |
| **Please write a short summary of your Capstone project and its components. Describe the problem statement, your approaches and your conclusions. (200-400 words)** |
| An Exploratory Data Analysis (EDA) project on Seoul bike sharing prediction aims to analyze the trends and patterns in bike sharing data in the city of Seoul in order to make accurate predictions on the demand for bike sharing services. The project involves several steps, including collecting and cleaning the data, identifying relevant variables and patterns, and using statistical and visualization techniques to understand the data.  The first step in the EDA process is to gather and prepare the data. This involves collecting data on various factors that may influence bike sharing demand, such as weather, seasonality, holidays, and location. The data may be obtained from various sources, such as government agencies, transportation departments, and bike sharing companies. Once the data has been collected, it must be cleaned and processed to ensure that it is accurate and ready for analysis.  We tried to answer the questions such as:  Currently Rental bikes are introduced in many urban cities for the enhancement of mobility comfort. The client is Seoul Bike, which participates in a bike share program in Seoul, South Korea. An accurate prediction of bike count is critical to the success of the Seoul bike share program. 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 dataset contains weather information (Temperature, Humidity, Windspeed, Visibility, Dewpoint, Solar radiation, Snowfall, Rainfall), the number of bikes rented per hour and date information.  We used different libraries to form tables and graphs in order to understand and answer these questions. We go to know that:   * Linear, Lasso, Ridge and Elastic Net: Linear, Lasso, Ridge and Elastic regression models have almost similar R2 scores (61%) on both training and test data. (Even after using GridserachCV we have got similar results as of base models). * Decision Tree Regression: On Decision tree regressor model, without hyper-parameter tuning, we got r2 score as 100% on training data and on test data it was very less. Thus, our model memorized the data. So it was a over fitted model. After hyper-parameter tuning we got r2 score as 88% on training data and 83% on test data which is quite good for us. * Random Forest: On Random Forest regressor model, without hyper-parameter tuning we got r2 score as 98% on training data and 90% on test data. Thus, our model memorized the data. So, it was a over fitted model, as per our assumption After hyper-parameter tuning we got r2 score as 90% on training data and 87% on test data which is very good for us. * Gradient Boosting Regression (Gradient Boosting Machine): On Random Forest regressor model, without hyper-parameter tuning we got r2 score as 86% on training data and 85% on test data. Our model performed well without hyper-parameter tuning. After hyper-parameter tuning we got r2 score as 96% on training data and 91% on test data, thus we improved the model performance by hyper-parameter tuning. |