BIKE SHARING DATASET ANALYSIS REPORT

(Hourly Basis)

Business Understanding

Abstract: This dataset contains the hourly and daily count of rental bikes between years 2011 and 2012 in Capital bikeshare system with the corresponding weather and seasonal information.

Problem Statement : Using historical usage patterns and weather data, forecast(predict) bike rental demand (number of bike users (‘cnt’)) on hourly basis.

Problem Motivation: Use the provided “Bikes Rental” data set to predict the bike demand. Therefore, sufficient number of bike for the customers can be arranged.

Use of this dataset in publications must be cited to the following publication:

[1] Fanaee-T, Hadi, and Gama, Joao, "Event labeling combining ensemble detectors and background knowledge", Progress in Artificial Intelligence (2013): pp. 1-15, Springer Berlin Heidelberg, doi:10.1007/s13748-013-0040-3.

Data Exploration

* All coding done in Python 3
* Extensive use of pandas , numpy , matplotlib , seaborn, sweetviz and sklearn packages.
* This dataset contains 17 different features on 17379 observation.
* There have categorical and numerical features.
* Target variables was ‘cnt’,which was count of total rental bikes including both casual and registered.
* Pandas package was imported and a dataframe was created.
* Numerical variables were visualized by matplotlib,seaborn and sweetviz package to analyse its distribution and its relation with the total rental bikes including both casual and registered.
* Sweetviz package also used to check kurtosis and skewness of each features. The data is asymmetric.
* The relationship between the dependent and independent variables look fairly linear. Thus, our linearity assumption is satisfied
* Categorical variables have been converted into numerical variables and visualized by matplotlib and seaborn package to analyse its distribution.

Chart, diagram

Description automatically generated

* The relation between bike users and season across a year.Season 1,2,3,4 represents springer, summer, fall, winter respectively.

Chart, line chart

Description automatically generated

* There have not missing values in categorical and numerical attributes.
* From the box plot, we can observed that few outliers are present in ‘holiday’,’weathersit’ and ‘hum’.Heavy outliers are presented in the ‘windspeed’,’’casual’,’registered’ and ’cnt’.

A screenshot of a computer

Description automatically generated with medium confidence

* Heatmap using seaborn package was created to show us correlations between the independent variables and the target variable outcome.
* 'temp' is highly correlated with 'atemp',whereas 'cnt' and 'registered' also. 'atemp',’casual’ and ‘registered’ will be dropped as they might have carried same information.
* ‘mnth’ and ‘season’ have correlated with each other.’yr’ and ‘instant’ same as them.

Data Preprocessing:

* For this pipeline at first we need to check missing values for the dataset features. In this dataset we are not having any missing values to handle.
* Then we need to convert categorical data to numerical equivalent, in this dataset we are having one categorical data(‘dteday’) for conversion, here we have use labelencoder to covert.
* As this a regression type of problem, so outliers handling is really very important. So, we have checked for outliers if there is any.
* In this data set ‘weathersit’,’hum’ required outlier treatment. So, we impute outliers using IQR method.

Feature Selection:

* In this pipeline at first we check for highly correlated feature using heatmap
* From the heatmap we can see that ‘instant’ and ‘dteday’ are highly correlated to each other, we need to drop one of them.
* ‘Temp’ and ‘atemp’ are highly correlated to each other, we need to drop one of them.
* ‘registered’ and ‘cnt’ are highly correlated to each other, need to drop one of them.

Chart, treemap chart

Description automatically generated

* Then we need to remove constant features, that means those features with standard deviation=0, in this dataset we do not have any constant feature to drop off.
* In this dataset we have one unique ID feature that is ‘instant’, we need to drop that one.

Chart, treemap chart

Description automatically generated

**Model Building:**

The Bike Rental Program is a regressive problem involves the prediction of count of total rental bikes including both casual and registered users.

The data is observed to be much neater this time. There seems to be no missing entries and other unknown values are found. The data is standardised to remove the skewness, kurtosis and to improve the performance of the model.

The Regression models used here are the following.

1) Linear Regression

2) Decision Tree Regressor

3) Random Forest Regressor

4) KNN Regressor

5) Gradient Boosting Regressor

6) XG Boost Regressor

7) Bagging Regressor

The major parameter used for evaluating the model are r2 score and adjusted r2score. R2 **shows how well terms** (data points) fit a curve or line. Adjusted R2 also indicates how well terms fit a curve or line, but adjusts for the number of terms in a model. R2 score is taken on both training and testing data. For a good model adjusted r2 score is less than r2 score. Different models and their performance are tabulated below.

**HOUR**

|  |  |  |
| --- | --- | --- |
| **MODEL** | **R2 Score** | **Adjusted R2 score** |
| Linear Regression | 0.60 | 0.59 |
| Decision Tree Regressor | 0.92 | 0.91 |
| Random Forest Regressor | 0.96 | 0.95 |
| KNN Regressor | 0.74 | 0.73 |
| Gradient Boosting  Regressor | 0.91 | 0.90 |
| XG Boost Regressor | 0.96 | 0.96 |
| Bagging Regressor | 0.97 | 0.96 |

**RESULT**

The final model chosen is Bagging Regressor as it shows the best performance of adjusted r2 score above 96%. Hyper parameter tuning will be performed on this model. However due to limited computation capabilities the hyper parameter tuning takes marginally higher times including hours and was unable to process it.

**Future Work**

Here are some ideas of future work to improve the performance of the data model further:

* Distribution adjustment of the target variable: Some predictive models assume a normal distribution of the target variable - a transformation in the data preprocessing could improve the performance of such methods.
* Large scale dataset implementation of random forests. For large scale datasets (> 10 Mio. samples) the used sklearn python implementation of random forests will extremely slow down if it is unable to hold all samples in the working memory or can run into serious memory problems. A solution could be the [woody](https://github.com/gieseke/woody) implementation with top trees for pre-classification and flat random forests implemented in C at the leaves of the top trees.