Technical Document

Seoul Bike Sharing Demand Prediction

Prepared By

Nitin Solanki
(solankinitin1210@gmail.com)

Contents

1. Objective

A bike-sharing system provides people with a sustainable mode of transportation and has beneficial effects on both the environment and the user. Nowadays when prices are breaking its all-time record, bike-sharing service has exponentially increased its popularity. Nonetheless, the increased usage has led to creating issues like the unavailability of bikes and docks at bike stations. The objective of this analysis is to predict the number of bikes at the station level for bike-sharing systems using machine learning models.

2. Introduction

Currently, Rental bikes are introduced in many urban cities for the enhancement of mobility comfort. Bike-sharing systems allow anyone to hire bicycles from one of the city's numerous automated rental stations. 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 the bike count required at each hour for the stable supply of rental bikes.

For this analysis dataset of bike-sharing demand in Seoul is given which is contained the 13 Independent variables and our target variable "Rented bike count"
In this analysis, we'll try to find the answers to the questions listed below.

1. To what extent we can predict the bike count required for the stable supply of rental bikes using machine learning?
2. Which machine learning algorithm gives the highest prediction accuracy?
3. Which features are the most important for bike count prediction?

3. Dataset

The dataset given for this analysis is Seoul-Bike-Sharing Demand, which is constitutes of Rented bike data with **8760** records of rental bike count per hour from 1st December 2017 till 31st December 2018, The features utilised from the data sets for the research are listed in Table

Sr.	Column	Column Description	Data type
1	Date	Year-Month-Day	Object
2	Rented Bike count	Count of bikes rented at each hour	Integer
3	Hour	Hour of the day (0-23)	Integer
4	Temperature(°C)	Temperature in Celsius for Hour of the day	Float
5	Humidity(%)	Humidity in percentage for Hour of the day	Integer
6	Windspeed(m/s)	Humidity in m/s for an Hour of the day	Float
7	Visibility(10m)	Humidity in metres for Hour of the day	Integer
8	Dew point temperature(°C)	Dew Point Temperature in Celsius for Hour of the day	Float
9	Solar radiation(MJ/m2)	Humidity in MJ/m2 for Hour of the day	Float
10	Rainfall(mm)	Humidity in mm for Hour of the day	Float
11	Snowfall(cm)	Humidity in cm for Hour of the day	Float
12	Seasons	Winter, Spring, Summer, Autumn	Object
13	Holiday	Holiday/No holiday	Object
14	Functional Day	NoFunc(Non Functional Hours), Fun(Functional hours)	Object

4. Data Modification and Exploration

In this section, Data modification, exploration and pre-processing will be explained, which is required before we input our data set to any machine learning algorithm in order to improve prediction for machine learning models.

Data exploration and pre-processing part will be done using pandas and NumPy packages. All the graphs and figures will use Matplotlib and seaborn package.
The first step is to deal with the date variable, Datatype of the date variable is Object type First it needs to convert it into Date format and the extract the week number for the give date and added a weekly column in the data set this way dates are grouped to gather in the week and then date columns is removed.

The next step is to handle categorical data, in the data having 3 categorical features, namely Seasons, Holidays, and Functioning Days. Machine learning can seldom deal with categorical data. Hence, statistical models must be converted into numerical ones. Many methods can be used for this convergence, here Label encoding (for holiday and Functioning days) and domification (for season) are used. A separate column is created for all four seasons and Holidays & function days contain binary numbers that range from 0 to 1.

The next step is to check the correlation of features and for that correlation matrix can be used, from the correlation matrix of the features in the dataset it is clear that temperature and time of day have the highest correlation. When comparing other features, the features "temperature (°C)" and "dew point temperature(°C)", are highly correlated, which causes multicollinearity. This can cause problems when fitting the models, so the dew point temperature is excluded from the dataset.

The next step is to verify that the data is normal. Depending on the distribution of the data, parameters will be decided. After plotting the distribution chart, it can be seen that the target variable distribution does not follow a normal distribution. Thus, so it requires to transform closely to resemble a normal distribution. Using this transformation can improve the predictive power of machine learning algorithms.

5. Implementation and Evaluation Metrics

In this Analysis, the data is split with a ratio of 80:20. Each model will be tested against the baseline regression model which is Linear Regression. All models will be optimized for better accuracy using Grid Search. Grid-search is used to find the model’s optimal hyperparameters that provide the most ’accurate’ predictions. The function is fed with different hyperparameters, model takes each parameter and calculates accuracy for it and selects that hyper-parameter which gives the best result.To generate the models ScKit-Learn will be used.

Models will be evaluated based on its accuracy using RMSE, R2 and Adjusted R2
adjusted R-squared can provide a more precise view of the correlation by also taking into account how many independent variables are added to a particular model against which the stock index is measured.

6. Different Regression Model

6.1 Linear Regression

When the dependent variable is continuous (e.g., bike count), linear regression is a good option. linear regression is one of the simplest and most widely-used models.

Linear regression assumes that the bike counts are linearly correlated to the features in the dataset such as temperature. It also assumes that attributes are independent of one another.

Linear regression fits a linear model with coefficients for each feature to minimize the mean square error in the linear regression approach, outliers can have a significant impact on the regression. Furthermore, linear regression may be prone to overfitting which will give low bias and high variance to overcome this issue we can use the regularisation technique (lasso ne ridge)

6.2 Lasso and Ridge Regression

Lasso and Ridge Regression are popular regularization techniques which are used to prevent overfitting. Ridge regression applies an L2 penalty, which penalized coefficients with higher weights.

Lasso regression employs the L1 normalization technique which penalizes less relevant features in the dataset by setting coefficients to zero and hence eliminates them. Lasso regression employs the L1 normalization technique which penalizes less relevant features in the dataset by setting coefficients to zero and hence eliminates them. so L2 model can be used for feature selection

It has been found that these algorithms perform better when dealing with lower numbers of features such as our dataset. In our analysis, both models were trained using grid search with different alpha values.

6.3 Decision Tree Regressor

Decision trees are statistical models that measure a target value using a collection of binary rules. To make a decision between 2 nodes, decision trees use **Attribute selection measure techniques** to decide to split a node into more sub-nodes.

There are two popular techniques for ASM

- Information Gain
- Gini Index

Both of the above techniques measure the entropy which is nothing but impurity/randomness in a given attribute and based on the result it chooses a split node and builds the decision tree.

Entropy(s)= -P(yes)log2 P(yes)- P(no) log2 P(no)
Information Gain= Entropy(S)- [(Weighted Avg) *Entropy (each feature)
Gini=1−∑ni=1(pi)2

The advantage of this algorithm is that less pre-processing is required as data scaling or data normalization is not necessary. However, this model can be computationally expensive if the data has a lot of features. Decision trees can completely fit the training data but tend to overfit the test data. An alternative to deal with overfitting in decision trees is to use random forests regressors.

6.4 Random forests regressors

Random forest works by training a large number of decision trees and then calculating the mean prediction of the individual trees. The notion of random implies randomly created decision trees. Random decision trees are created on different subsets of the features and data points. For accurate predictions, random forest regressors can be optimized by hyperparameter tuning to ensure that the model does not depend too heavily on any single feature and that all potentially predictive features are considered equally. Also due to the previously mentioned random creation of decision trees, adding randomness prevents overfitting.

Random forest regression provides is feature importance estimate. Using feature importance, the effort can aid in a deeper understanding of the solved problem and, in some cases, contribute to model improvements.

6.5 Bagging and Boosting Regression

Random forest regression Bagging and Bosting both are The ensembles is a method used in the machine learning algorithm In this method, multiple models or ‘weak learners’ are trained to rectify the same problem and integrated to gain desired results. Weak models combined rightly give accurate models.

Bagging is a homogeneous weak learners’ model that learns from each other independently in parallel and combines them for determining the model average.
Boosting is also a homogeneous weak learners’ model but works differently from Bagging. In this model, learners learn sequentially and adaptively to improve model predictions of a learning algorithm.

The difference between both techniques is Bagging is a technique for reducing prediction variance by producing additional data for training from a dataset by combining repetitions with combinations to create multi-sets of the original data. Boosting is an iterative strategy for adjusting an observation's weight based on the previous classification. It attempts to increase the weight of observation if it was erroneously categorized. Boosting creates good predictive models in general.

Bagging and Boosting reduce variance and provide higher stability with minimizing errors.
In our analysis, we found that bagging is performed better than boosting

7. Feature Importance

Feature importance was extracted from the Random forest regressor importance variable module. The feature importance method uses permutation data to quantify each variable's effect on the established model's overall predictive output. The higher the accuracy of prediction, the more important the element, and vice versa.

Looking back at the feature importance graph, temperature accounts for the highest importance. It showed how different weather parameters correlate to people's mobility Second most important feature is Hour in the day. Considering this information bike, sharing demand companies should put more bikes in more active periods than inactive periods such as lunch break. Therefore, time and temperature are very important factors in the bike-sharing demand market and our model must make the right predictions.

8. Score Board

Sr.	Model	MSE	RMSE	R2	Adj_R2
1	Linear model	118654.856	344.4631417	0.716491154	0.710201991
2	Lasso Model	119117.3406	345.133801	0.715386113	0.709072437
3	Ridge Model	118958.7106	344.903915	0.715765137	0.709459869
4	Decision Tree	93876.17135	306.3921855	0.775696285	0.770720488
5	Random Forest	59814.22126	244.569461	0.857082454	0.853912071
6	Boosting	59858.25091	244.6594591	0.856977252	0.853804534
7	Bagging	55177.40443	234.898711	0.868161466	0.865236852

9. Conclusion

Reflecting on Analysis questions

1. **To what extent we can predict the bike count required for the stable supply of rental bikes using machine learning?**
   Multiple regressors were compared and evaluated with Adj_R2 and RMSE scores. With the best model, Bagging achieved an accuracy of 86% and RMSE of 234, which indicates that machine learning indeed can predict bike counts required for a stable supply of rental bikes

2. **Which machine learning algorithm gives the highest prediction accuracy?**
   The Bagging regressor performed the best out of 7 traditional models on the data set. Which has produced the highest accuracy of Adj_R2 of 86%. The reason why this algorithm produced good results is that Bagging decreases variance, not bias, and solves over-fitting issues in a model.

3. **Which features are the most important for bike count prediction?**
   The feature temperature accounts for the highest importance therefore people of Seoul are more likely to stay home during colder days than on warmer ones. Depending on the feature importance of temperature and study on people activity associated with whether the temperature is the most important factor when it comes to sharing bikes.

10. Library and packages are used for analysis

- **libraries for process data**

```
import pandas as pd
import numpy as np
from numpy import math
```

- **libraries for plotting data**

```
import matplotlib.pyplot as plt
%matplotlib inline
import seaborn as sns
```

- **To prepare our train and test dataset**

```
from sklearn.model_selection import train_test_split
```

- **For Standardise our dataset (it requires for linear regression)**

```
from sklearn.preprocessing import StandardScaler
```

- **For cross validation and hyperperameter tuning**

```
from sklearn.model_selection import GridSearchCV
```

- **For Calculate VIF**

```
from statsmodels.stats.outliers_influence import variance_inflation_factor
```

- **libraries for regression**

```
from sklearn.linear_model import LinearRegression
from sklearn.linear_model import Ridge
from sklearn.linear_model import Lasso
from sklearn.tree import DecisionTreeRegressor
from sklearn.ensemble import RandomForestRegressor
from sklearn.ensemble import GradientBoostingRegressor
from sklearn.ensemble import BaggingRegressor
```

- **libraries for measuring performance metrics**

```
from sklearn.metrics import r2_score
from sklearn.metrics import mean_squared_error
```