Bike Rental Count Prediction

Vikash Kumar

9th July 2019

Contents

1. **Introduction**
   1.1 Problem Statement
   1.2 Data

2. **Methodology**
   2.1 Pre-Processing
      2.1.1 Outlier Analysis
      2.1.2 Feature Selection

   2.2 Visualization

   2.3 Modeling
      2.2.1 Model Selection
      2.2.2 Linear Regression Model
      2.2.3 Decision Tree
      2.2.4 Support Vector Regression

3. **Conclusion**

   3.1 Model Evaluation

   3.2 Improvement

References

Introduction

Problem Statement

We have got a problem to count the number of bikes on daily basis given some weather and seasonal condition. Aim of this project is to find the number of bikes on daily basis given some conditions. If bike rental company wants to add some more bike to its center it needs the insights of the data the when the bikes are in high demand and which factor is affecting the number of bikes to be rented. If the bike rental company wants to invest more on specific season it should have knowledge of requirements of bikes by the user.

Data

The data we have got is well normalized and standardized so we do not have to perform normalization on the data to make it use it in our model. On the data received we try to build regression models and check which model is predicting well and which model is to be selected.

instant	dteday	season	yr	mnth	holiday	weekday	workingday	weathersit
1	2011-01-01	1	0	1	0	6	0	2
2	2011-01-01	1	0	1	0	0	0	2
3	2011-01-03	1	0	1	0	1	1	1
4	2011-01-04	1	0	1	0	2	1	1
5	2011-01-05	1	0	1	0	3	1	1

instant	weathersit	temp	atemp	hum	windspeed	casual	registered	cnt
1	2	0.344	0.363	0.805	0.160	331	654	985
2	2	0.363	0.353	0.696	0.248	131	670	801
3	1	0.196	0.189	0.437	0.248	120	1229	1349
4	1	0.200	0.212	0.590	0.160	108	1454	1562
5	1	0.226	0.229	0.436	0.186	82	1518	1600

Data Stats:

	instant	season	yr	mnth	holiday	weekday	workingday	weathersit
count	731	731	731	731	731	731	731	731
mean	366	2.49	0.50	6.51	0.02	2.99	0.68	1.39
std	211	1.11	0.50	3.45	0.16	2.00	0.46	0.54
min	1.00	1.00	0.00	1.00	0.00	0.00	0.00	1.00
25%	183.5	2.00	0.00	4.00	0.00	1.00	0.00	1.00
50%	366	3.00	1.00	7.00	0.00	3.00	1.00	1.00
75%	548	3.00	1.00	10.0	0.00	5.00	1.00	2.00
max	731	4.00	1.00	12.0	1.00	6.00	1.00	3.00

	temp	atemp	hum	windspeed	casual	registered	cnt
count	731	731	731	731	731	731	731
mean	0.49	0.47	0.62	0.19	848	3656	4504
std	0.18	0.16	0.14	0.07	686	1560	1937
min	0.05	0.07	0.00	0.02	2.00	20.0	22.0
25%	0.33	0.33	0.52	0.13	315	2497	3152
50%	0.49	0.48	0.62	0.18	713	3662	4548
75%	0.65	0.60	0.73	0.23	1096	4776	5956
max	0.86	0.84	0.97	0.50	3410	6946	8714

The characteristics of the dataset are very favorable because it was already processed. It is very concise, and missing values are not a problem. Also, most of the data is already normalized or binary. Other categorical data like 'weekday' or 'working day'/'holiday' were processed and transformed into dummy variables.

Methodology

Pre-Processing

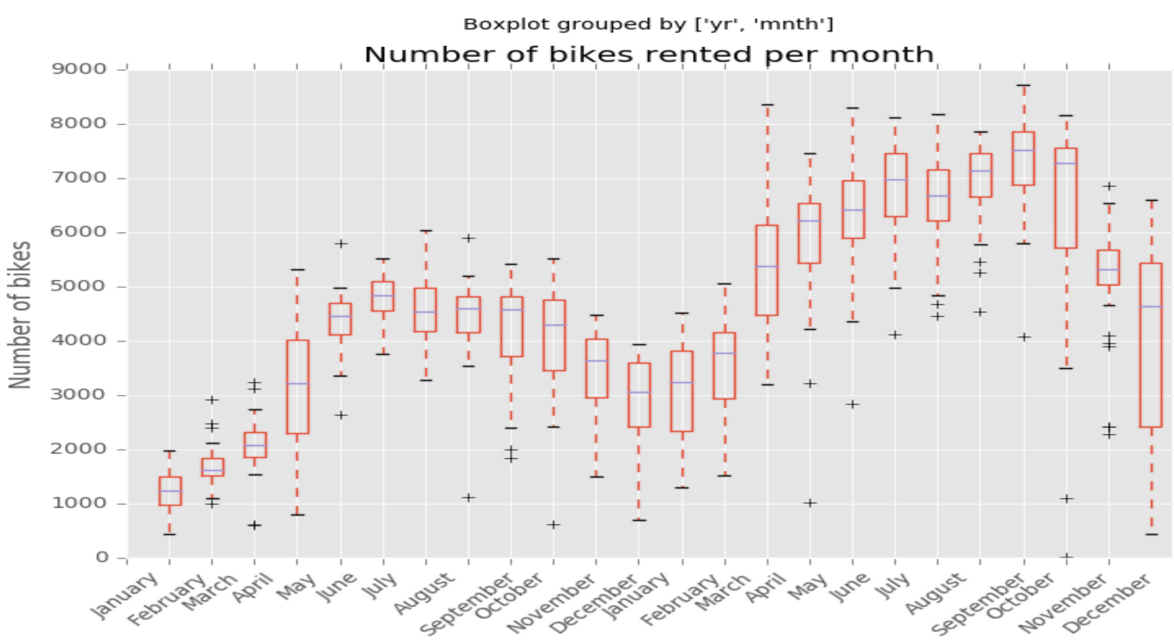
As described in ‘Characteristics’ most of the preprocessing is provided with the data set. Dates get dropped because the regressor can not read this datatype and the order information is already stored in the index. The instant variable replicates this information also. These features are dropped because the order should not differentiate the data points. The January 1st of 2011 is not better or worse than January 1st 2012 by the order of the data set. It should differentiate on the years feature, but that information is stored in the ‘yr’ feature already. Keeping date (and instance) in would overemphasize these features.

Visualization

The visualization shows a classic seasonal pattern with an up-trend year over year.

Unsurprisingly bike renting is much more popular in the summer month. Spring and autumn months show higher volatility than the rest of the year, which is likely due to changing weather conditions.

There are some outliers throughout the dataset, mostly on the lower end. These are left in the dataset because they are not due to measurement errors, but to extreme weather conditions. Because extreme weather conditions are part of the problem the data is not excluded.



Modelling

I used different regression model to check, which model performs well and then selecting the model on the basis of their score and evaluation metrics. I have used two evaluation metrics to determine the accuracy of model i.e. R^2 square and RMSE value. To measure the performance of the regressions three standard regression metrics are used: Root Mean Squared Error (RMSE) and the coefficient of determination (R^2). Both metrics are calculated for both regressor types. For comparison RMSE is used and R^2 for parameter tuning. “The RMSE is directly interpretable in terms of measurement units, and so is a better measure of goodness of fit than a correlation coefficient.”

We uses different modelling techniques to check the accuracy of the model on the given sample data. On the basis of evaluation metrics we check the model predictions and its accuracy level.

Model evaluation

We evaluate model on the basis of R2 score and RMSE value and then select the model as required. We got different evaluation score for each evaluating model and then we select the model which is having high accuracy and less errors in prediction.

Results obtained from Decision tree:-

```
Dicision tree results:

R2_score: 0.794462
Rmse: 858.613871

MAPE results:

24.17879818607351
```

Results obtained from Linear Regression:

```
Statsmodel OLS :

R2_score:0.727670
rmse:989.283451

MAPE error:

137.65045656074582
```

Results obtained from Linear Regression 2:

```
R2_score:0.817429
RMSE: 840.551999

MAPE Results

125.03926305770298
```

Results obtained from Support Vector Regression:

```
R^2_Score SVR: 0.018109
RMSE SVR: 1957.802353

 MAPE Results

62.117049686534344
```

Results obtained from Support Vector Regression after tuning:

```
Support Vector Regression Results

Score SVR tuned GS: 0.840838

RMSE SVR tuned GS: 778.672119

MAPE results:

16.70721320805383
```

Conclusion

As expected the tuning of the parameters of the regressors improves the performance. Parameter tuning with grid search can improve the performance even further after we apply different regression model on it.

The tuned SVR beats all the model by far and gives 80% coefficient.

Splitting the dataset and predicting casual and registered customers separately may increase the R^2 score also slightly, which is not done on this project.

A coefficient of determination of more than 80% is a decent result for the SVR regressor. All the other models performance is not satisfactory and cannot be selected in the predictions of rental bike count. We have to stick to the tuned SVR model for that as if now.

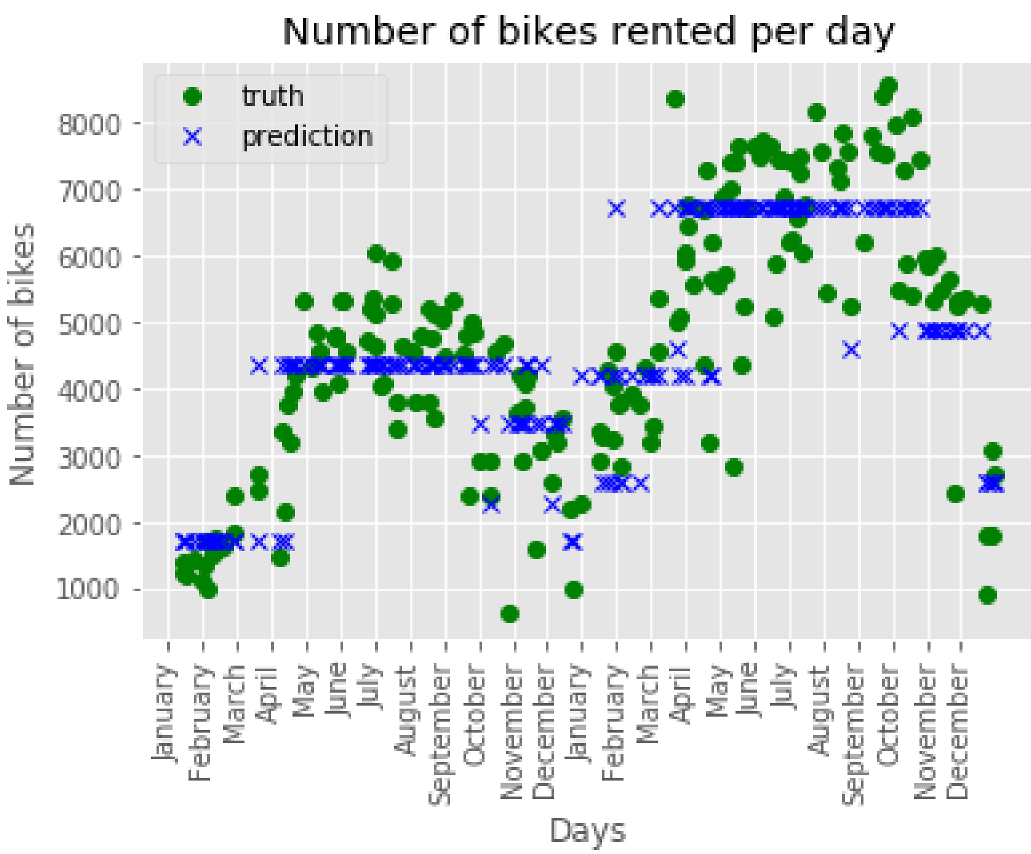
In coming future we can apply some more regression algorithm and try to calculate the accuracy and its coefficient.

Reflection

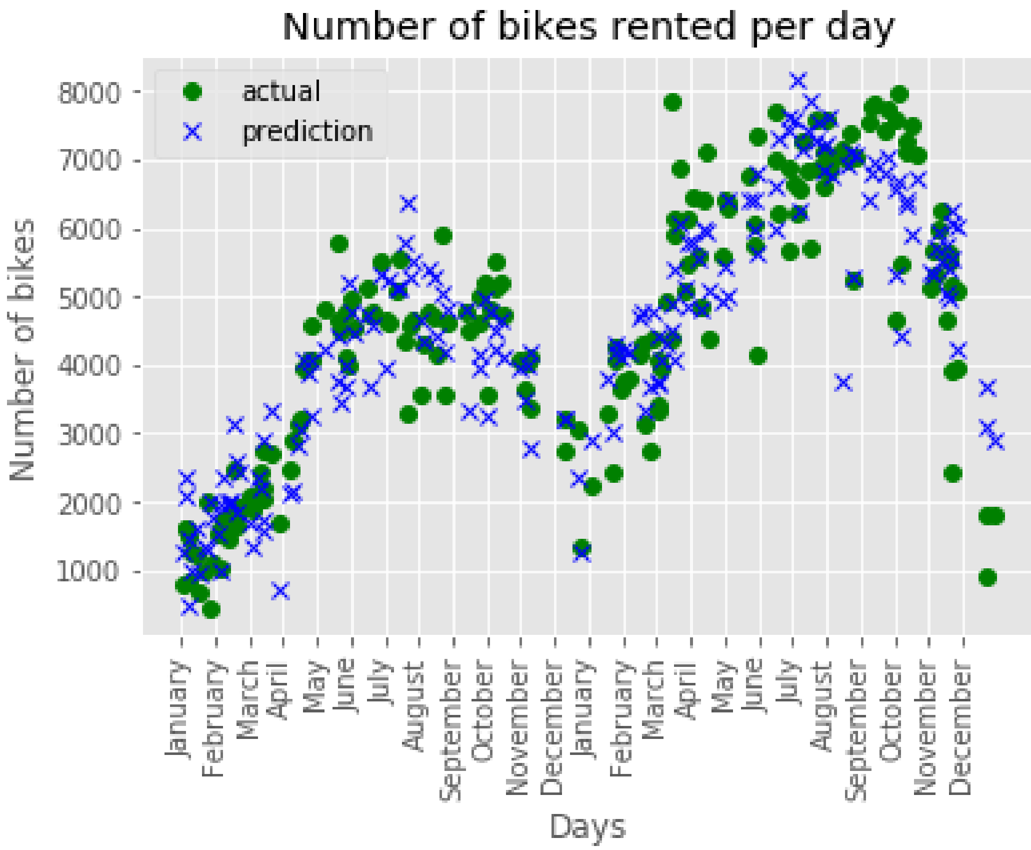
We have used different regressor model to check which model is performing well and which is having coefficient value is higher. Higher the coefficient value better the model. We have created decision tree model, linear regression model of two types and a support vector regression. We can also implement other models such as DNN regressor which is also a good model for high amount of data, which can also be used here to check if it is working fine when tuned properly.

Visualization

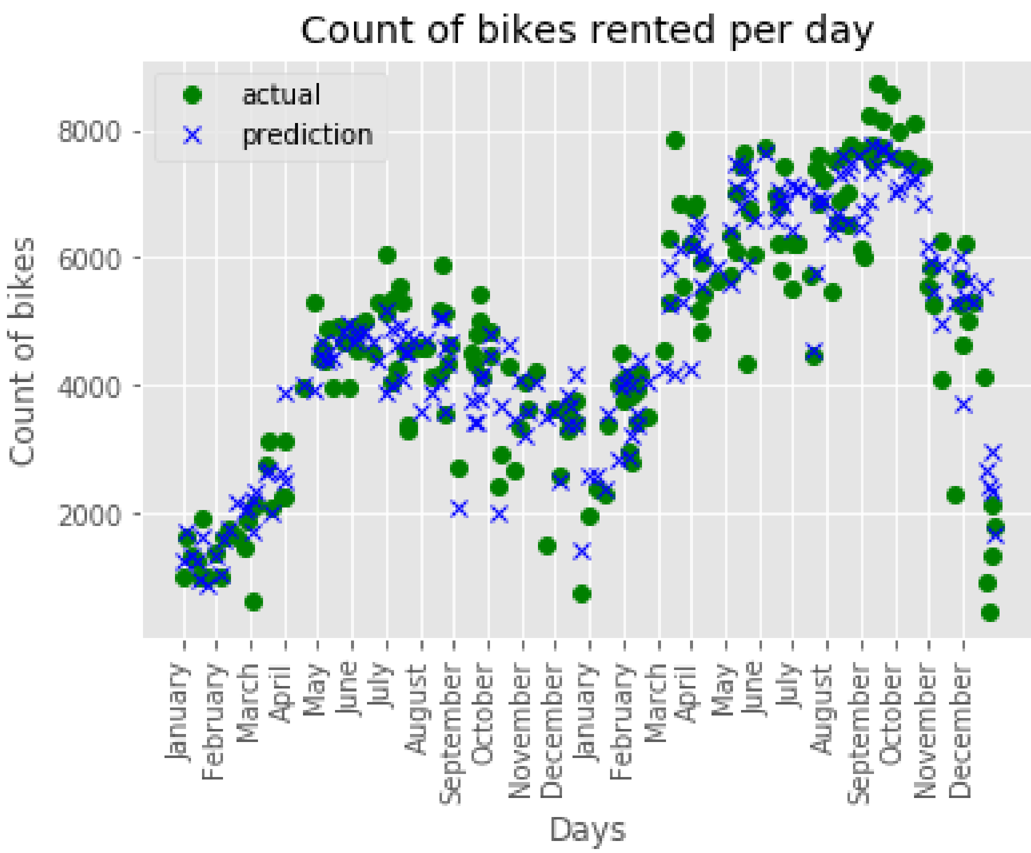
Output of decision tree



Output of linear regression



Output of Support Vector Regression



Improvement

The coefficient of determination of the regressors could be increased by additional iterations in training and the number of folds in the cross validation, at the expense of computing time. Of course, there are also other regressors available that might perform better on this particular dataset. For example, a wide and deep learning algorithm might be a better-performing alternative. We can also different hyperparameter tuning techniques such as randomized search and many other techniques to tune a model.

References

> http://www.capitalbikeshare.com

> http://archive.ics.uci.edu/ml/datasets/Bike+Sharing+Dataset

> http://freemeteo.de/wetter/

> http://dchr.dc.gov/page/holiday-schedules

> http://scikit-learn.org/stable/

> https://www.tensorflow.org

> https://www.tensorflow.org/versions/r0.9/tutorials/linear/overview.html#large-scalelinearmodels-with-tensorflow

>https://www.tensorflow.org/versions/r0.9/api_docs/python/contrib.learn.html#DNNRegressor

> http://scikit-learn.org/stable/tutorial/machine_learning_map/index.html

> http://scikit-learn.org/stable/modules/generated/sklearn.svm.SVR.html#sklearn.svm.SVR

> http://scikit-learn.org/stable/modules/sgd.html#regression

> http://scikit-learn.org/stable/modules/svm.html

> http://scikit-learn.org/stable/auto_examples/svm/plot_rbf_parameters.html

> http://scikit-learn.org/stable/modules/sgd.html#regression

> http://www.vernier.com/til/1014/

> https://github.com/tensorflow/skflow/blob/master/g3doc/api_docs/python/

estimators.md

> http://scikit-learn.org/stable/modules/generated/

sklearn.grid_search.GridSearchCV.html

>http://scikitlearn.org/stable/modules/generated/sklearn.grid_search.RandomizedSearchCV.html

>http://scikitlearn.org/stable/modules/generatedsklearn.metrics.mean_squared_error.html#sklearn.metrics.mean_squared_error

> http://scikit-learn.org/stable/modules

Complete R Code:

#Bike Renting problem solved and the number of bikes to be predicted given several condition using several regressor model.

#load the data file.

bike_data=read.csv("C:/Users/Vikash Singh/Desktop/r and python/data/day.csv")

summary(bike_data)

str(bike_data)

#dteday and instant are not requried in the modelling of the data and some values of dteday are missing also so we try not to include that data

bike_data <- bike_data[-c(0:2)]

str(bike_data)

#as all the variables are well normaliszed so we try to develop a model on the data

#split data into train and test

set.seed(123)

ind <-sample(2,nrow(bike_data),replace=TRUE,prob=c(0.7,0.3))

train <- bike_data[ind==1,]

validate<-bike_data[ind==2,]

#develop a regression model on the training data

library(rpart)

library(MASS)

train<-train[-c(12:13)]

validate<-validate[-c(12:13)]

#using dicision tree

dt=rpart(cnt ~ .,data=train,method = "anova")

dt

prediction_dt=predict(dt,validate[,-14])

#defining a function to detect the error in model

```
mape=function(x,xt)
{
  mean(abs((x-xt)/x))*100
}
mape(validate[,12],prediction_dt)
library(MLmetrics)#R2_score calculation
library(DMwR)
regr.eval(validate[,12],prediction_dt,stats = c("mae","rmse","mape"))
#       mae       rmse       mape
#723.5856681 949.0313924   0.2546648
R2_Score(prediction_dt,validate[,12])
#R2_score=0.7888387



#using linear regression
#load the data
bike_data=read.csv("C:/Users/Vikash Singh/Desktop/r and python/data/day.csv")
summary(bike_data)
str(bike_data)
#dteday and instant are not requried in the modelling of the data and some values of dteday are missing also so we try not to include that data
bike_data <- bike_data[-c(0:2)]
str(bike_data)
#as all the variables are well normaliszed so we try to develop a model on the data
#split data into train and test
set.seed(123)
```

```
ind <-sample(2,nrow(bike_data),replace=TRUE,prob=c(0.7,0.3))

train <- bike_data[ind==1,]

validate<-bike_data[ind==2,]


library(usdm)

vif(bike_data[,-14])

vifcor(bike_data[,-14],th=0.9)

train<-train[-c(12:13)]

validate<-validate[-c(12:13)]

lm_model=lm(cnt ~ .,data=train)

summary(lm_model)

predictions_lm=predict(lm_model,validate[,1:11])

predictions_lm

regr.eval(validate[,12],predictions_lm,stats = c("mae","rmse","mape"))

#     mae       rmse       mape

#727.9427779 929.1479631   0.2318177

R2_Score(predictions_lm,validate[,12])

#R2_score=0.7975942




#using linear regression and deleting the variable which is having collinearity problem

#using linear regression

#load the data

bike_data=read.csv("C:/Users/Vikash Singh/Desktop/r and python/data/day.csv")

summary(bike_data)

str(bike_data)
```

```
#dteday and instant are not requried in the modelling of the data and some values of dteday are missing also so we try not to include that data

bike_data <- bike_data[-c(0:2)]

str(data)

str(bike_data)

#as all the variables are well normaliszed so we try to develop a model on the data

#split data into train and test

set.seed(123)

ind <-sample(2,nrow(bike_data),replace=TRUE,prob=c(0.7,0.3))

train <- bike_data[ind==1,]

validate<-bike_data[ind==2,]


library(usdm)

vif(bike_data[,-14])

vifcor(bike_data[,-14],th=0.9)

train<-train[-c(9)]

validate<-validate[-c(9)]

train<-train[-c(11:12)]

validate<-validate[-c(11:12)]

lm_model2=lm(cnt ~ .,data=train)

summary(lm_model2)

predictions_lm2=predict(lm_model2,validate[,1:10])

predictions_lm2

regr.eval(validate[,11],predictions_lm2,stats = c("mae","rmse","mape"))

#       mae       rmse       mape

#733.598349 934.099655   0.232666

R2_Score(predictions_lm2,validate[,11])
```

#R2_score=0.7954311

plot(lm_model2)

#support vector regression model

#load the data

bike_data=read.csv("C:/Users/Vikash Singh/Desktop/r and python/data/day.csv")

summary(bike_data)

str(bike_data)

#dteday and instant are not requried in the modelling of the data and some values of dteday are missing also so we try not to include that data

bike_data <- bike_data[-c(0:2)]

str(data)

str(bike_data)

#as all the variables are well normaliszed so we try to develop a model on the data

#split data into train and test

set.seed(123)

ind <-sample(2,nrow(bike_data),replace=TRUE,prob=c(0.7,0.3))#divides the data into 70% and 30% for training and testing respectively.

train <- bike_data[ind==1,]

validate<-bike_data[ind==2,]

train<-train[-c(12:13)]

validate<-validate[-c(12:13)]

library(e1071)

```
svmt<-svm(cnt~.,data=train,kernel='linear',cost=1.0,epsilon=0.001)#can try epsilon values from 0.1 to 0.0001

svmt

predictions_svr=predict(svmt,validate[,1:11])

predictions_svr

mape(validate[,12],predictions_svr)

regr.eval(validate[,12],predictions_svr,stats = c("mae","rmse","mape"))

#        mae        rmse        mape

#703.5182510 914.3685016   0.2287892

R2_Score(predictions_svr,validate[,12])

#r2_score=0.8039821
```

#For Creator use only

#In case if we require the model for registered and casual users we can implement the model on both the variables and calcluate the required result out of it.

#we found out that support vector regression gives the best results and best value of r2 and rmse error is less.

Now we apply Support vector regression model for casual and registered users

```
bike_data=read.csv("C:/Users/Vikash Singh/Desktop/r and python/data/day.csv")

summary(bike_data)
```

```
str(bike_data)

#dteday and instant are not requried in the modelling of the data and some values of dteday are missing also so we try not to include that data

bike_data <- bike_data[-c(0:2)]

str(data)

str(bike_data)

#as all the variables are well normaliszed so we try to develop a model on the data

#split data into train and test

set.seed(123)

ind <-sample(2,nrow(bike_data),replace=TRUE,prob=c(0.7,0.3))

train <- bike_data[ind==1,]

validate<-bike_data[ind==2,]

train<-train[-c(13:14)]

validate<-validate[-c(13:14)]


svmt_c<-svm(casual~.,data=train,kernel='linear',cost=1.0,epsilon=0.001)#can try epsilon values from 0.1 to 0.0001

svmt_c

predictions_svr_c=predict(svmt,validate[,1:11])

predictions_svr_c

mape(validate[,12],predictions_svr_c)

regr.eval(validate[,12],predictions_svr_c,stats = c("mae","rmse","mape"))

#      mae       rmse       mape

#290.9666742 401.1489774   0.8366304

R2_Score(predictions_svr_c,validate[,12])

#0.6790322
```

#For Self Use

#Now we take registered users for prediction

bike_data=read.csv("C:/Users/Vikash Singh/Desktop/r and python/data/day.csv")

summary(bike_data)

str(bike_data)

#dteday and instant are not requried in the modelling of the data and some values of dteday are missing also so we try not to include that data

bike_data <- bike_data[-c(0:2)]

str(data)

str(bike_data)

#as all the variables are well normaliszed so we try to develop a model on the data

#split data into train and test

set.seed(123)

ind <-sample(2,nrow(bike_data),replace=TRUE,prob=c(0.7,0.3))

train <- bike_data[ind==1,]

validate<-bike_data[ind==2,]

train<-train[-c(12,14)]

validate<-validate[-c(12,14)]

svmt_r<-svm(registered~.,data=train,kernel='linear',cost=1.0,epsilon=0.001)#can try epsilon values from 0.1 to 0.0001 the values and model performance changes slightly.

svmt_r

predictions_svr_r=predict(svmt,validate[,1:11])

predictions_svr_r

mape(validate[,12],predictions_svr_r)

regr.eval(validate[,12],predictions_svr_r,stats = c("mae","rmse","mape"))

mae rmse mape

#2831.1847734 3213.7567470   0.7505285

R2_Score(predictions_svr_r,validate[,12])

#-2.801703

#if the r2_score is in -ve the model is performing worse, so we understand that for the registered users the model is not performing as expected.

#as data is limited the model performance can be affected.