**BIKE RENTAL PREDICTION**

Submitted by: N. Laxmi

Date: 28.11.2019

**CONTENTS**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **1** | **Introduction** | | | **3** |
|  | 1.1 | Problem Statement | | 3 |
|  | 1.2 | Data |  | 3 |
| **2** | **Methodology** | | | **5** |
|  | 2.1 | Data Preprocessing | | 5 |
|  |  | 2.1.1 | Missing Value analysis | 5 |
|  |  | 2.1.2 | Univariate Analysis | 5 |
|  |  | 2.1.3 | Bivariate Analysis | 8 |
|  |  |  |  |  |
|  | 2.2 | Outlier Analysis | | 10 |
|  | 2.3 | Feature Selection | | 14 |
|  | 2.4 | Dimensionality reduction | | 15 |
|  | 2.5 | Feature scaling | | 16 |
| **3** | **Modeling** | |  | **17** |
|  | 3.1 | Model building | | 14 |
|  | 3.2 | Model evaluation | | 15 |
|  | 3.3 | Model selection | | 17 |
|  | 3.4 | Conclusion | | 19 |
|  |  |  | |  |

**1. INTRODUCTION**

1. **Problem Statement**

The Bike Rental Data contains the daily count of rental bikes between the year 2011 and 2012 with corresponding weather and seasonal information. We would like to predict the daily count of bike usage in order to automate the system.

1. **Data**

Our task is to build a Regression model which will give the daily count of bikes rented based on weather and environmental conditions.

Given below is a sample of the data set that we are using to predict the count:

Table 1.1: Bike Rental Sample Data (Columns: 1-8)

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| instant | dteday | Season | yr | mnth | holiday | weekday |
| 1 | 1/1/2011 | 1 | 0 | 1 | 0 | 6 |
| 2 | 1/2/2011 | 1 | 0 | 1 | 0 | 0 |
| 3 | 1/3/2011 | 1 | 0 | 1 | 0 | 1 |
| 4 | 1/4/2011 | 1 | 0 | 1 | 0 | 2 |
| 5 | 1/5/2011 | 1 | 0 | 1 | 0 | 3 |

Table 1.2: Bike Rental Sample Data (Columns: 9-14)

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| weathersit | temp | atemp | Hum | windspeed | casual | registered | cnt |
| 2 | 0.344167 | 0.363625 | 0.805833 | 0.160446 | 331 | 654 | 985 |
| 2 | 0.363478 | 0.353739 | 0.696087 | 0.248539 | 131 | 670 | 801 |
| 1 | 0.196364 | 0.189405 | 0.437273 | 0.248309 | 120 | 1229 | 1349 |
| 1 | 0.2 | 0.212122 | 0.590435 | 0.160296 | 108 | 1454 | 1562 |
| 1 | 0.226957 | 0.22927 | 0.436957 | 0.1869 | 82 | 1518 | 1600 |

Below are the variables we used to predict the count of bike rentals

Table 1.3: Bike Rental Predictors

|  |  |
| --- | --- |
| s.no | Variables |
| 1 | Dteday |
| 2 | Season |
| 3 | Yr |
| 4 | Mnth |
| 5 | Holiday |
| 6 | Weekday |
| 7 | workingday |
| 8 | weathersit |
| 9 | Temp |
| 10 | Atemp |
| 11 | Hum |
| 12 | windspeed |
| 13 | Casual |
| 14 | registered |

**CHAPTER 2**

**METHODOLOGY**

1. **Data Preprocessing:**

Any predictive modeling requires that we look at the data before we start modeling. However, in data mining terms *looking at data* refers to so much more than just looking. Looking at data refers to exploring the data, cleaning the data as well as visualizing the data through graphs and plots. This is often called as Exploratory Data Analysis.

**2.1.1 Missing Value Analysis:**

Missing values in data is a common phenomenon in real world problems. Knowing how to handle missing values effectively is a required step to reduce bias and to produce powerful models.

Below table illustrate no missing value present in the data.

2.1 missing values

|  |  |  |
| --- | --- | --- |
| s.no | Variables | missing values |
| 1 | dteday | 0 |
| 2 | season | 0 |
| 3 | yr | 0 |
| 4 | mnth | 0 |
| 5 | holiday | 0 |
| 6 | weekday | 0 |
| 7 | workingday | 0 |
| 8 | weathersit | 0 |
| 9 | temp | 0 |
| 10 | atemp | 0 |
| 11 | hum | 0 |
| 12 | windspeed | 0 |
| 13 | casual | 0 |
| 14 | registered | 0 |

**2.1.2. Univariate Analysis**

Here, we plot the probability density functions numeric variables present in the data including target variable ‘cnt’..

1. Target variable ‘cnt’ is normally distributed
2. Independent variables like‘temp’,’atemp’, and ‘registered’ data is distributed normally.
3. Independent variable ‘windspeed’ is skewed to the left so; there are chances of getting outliers.

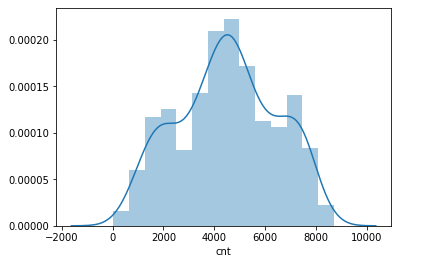


Figure 2.1 Bar graph showing distribution of target variable ‘cnt’

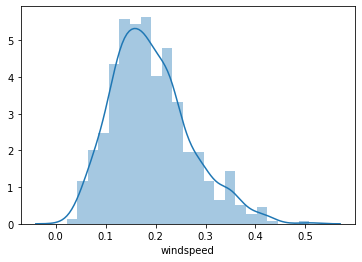


Figure 2.2: Distribution of ‘windspeed’ variable.

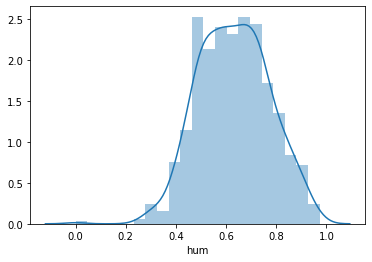


Figure 2.3: Distribution of ‘hum’ variable.

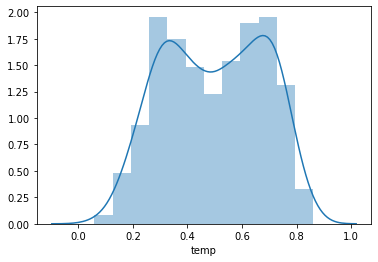


Figure 2.3: Distribution of ‘temp’ variable.

**2.1.3. Bivariate Analysis**

Bivariate descriptive displays or plots are designed to reveal the relationship between two variables. As was the case when examining single variables, there are several basic characteristics of the relationship between two variables that are of interest. These include:

* the form of the relationship
* the strength of the relationship, and
* the dependence of the relationship on external (to the two variables being examined) circumstances.

Bivariate plots provide the means for characterizing pair-wise relationships between variables.

A scatter plot displays the values of two variables at a time using symbols, where the value of one variable determines the relative position of the symbol along the X-axis and the value of a second variable determines the relative position of the symbol along the Y-axis.

Below figures shows relationship between independent variables and also with numeric target variable using ggpair

1. Below scatter graph is showing clearly that relationship between various independent variables and target variable.
2. The relationship between ‘hum’ , ‘windspeed’ with target variable ‘cnt’ is less.

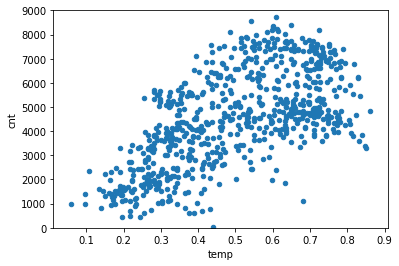


Figure: Relation between Numerical Variable 'temp' and target variable 'cnt'

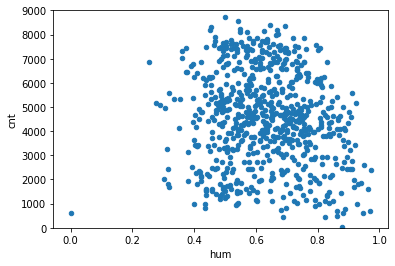


Figure: Relation between Numerical Variable hum and target variable 'cnt'

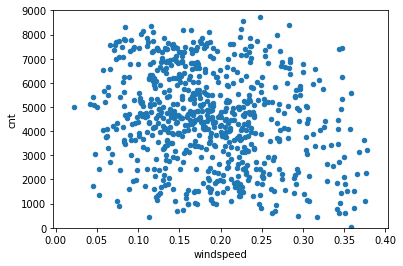


Figure: Relation between Numerical Variable ‘windspeed’ and target variable 'cnt'

**2.2. Outlier Analysis**

The other steps of preprocessing technique is Outliers analysis , an outlier is an observation point that is distant from other observations. Outliers in data can distort predictions and affect the accuracy, if you don’t detect and handle them appropriately especially in regression models.

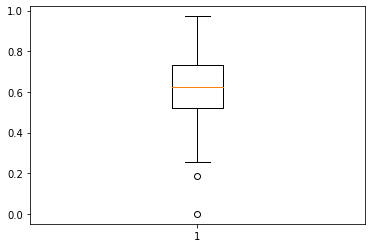


Figure: Boxplot of variable ‘hum’ shows no outliers

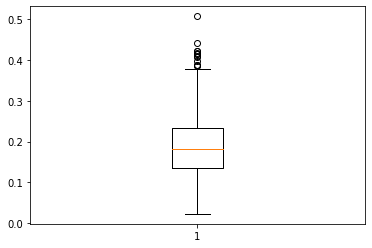


Figure: Boxplot of variable ‘windspeed’ shows the presence of outliers

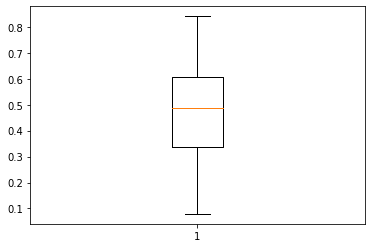


Figure: Boxplot of variable ‘atemp’ shows no outliers

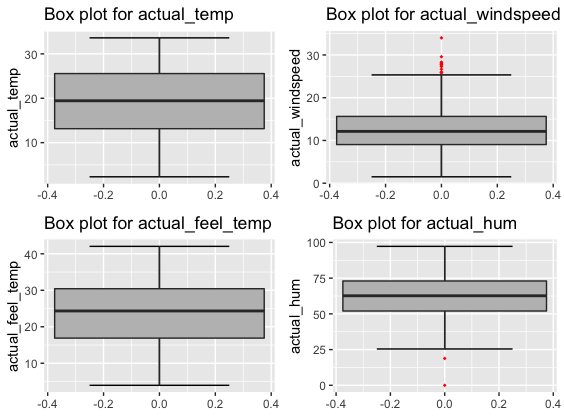
****

Figure: Boxplot of variables namely: temp, atemp, windspeed and hum.

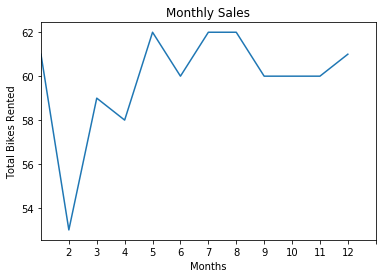


Figure: Graph showing the bikes rented each ‘month’

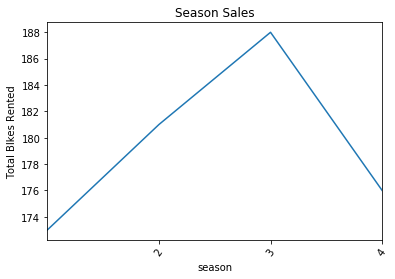


Figure: Graph showing the bikes rented each ‘season’

**2.3. Feature Selections**

Feature Selection is one of the core concepts in machine learning which hugely impacts the performance of your model. The data features that you use to train your machine learning models have a huge influence on the performance you can achieve. Irrelevant or partially relevant features can negatively impact model performance. Feature selection and Data cleaning should be the first and most important step of your model designing.

We should consider the selection of feature for model based on below criteria

1. The relationship between two independent variable should be less and
2. The relationship between Independent and Target variables should be high.

Below figure illustrates that relationship between all numeric variables using Corrgram plot .

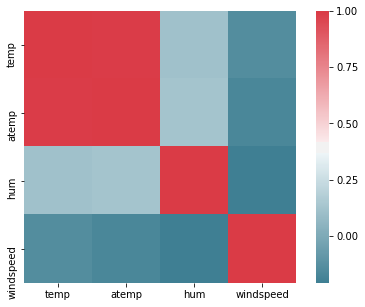
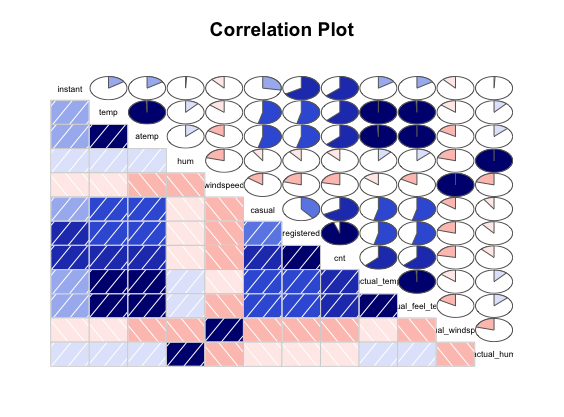


Figure 2.6. correlation plot of numeric variables



In above visualization we can see that only 2 variables are highly correlated with each other. Dark blue color represent highly correlated and light color represent very less correlated so we have a choice to remove either temp or atemp because these variables contains nearly equal information. So I have removed atemp variable from dataset.

Corrgram : The corrgram function produces a graphical display of a correlation matrix, called a correlogram. The cells of the matrix can be shaded or colored to show the correlation value.

**2.4. Dimensionality Reduction for numeric variables**

There is strong relationship between independent variables ‘temp’ and ‘atemp’ so considering any one feature enough to predict the better.

**Dimensional Reduction for categorical variables**

.

The above figure shows that variables ‘season’ ‘windspeed’ , ‘weekday ‘ ,‘weathersit’ and ‘holiday’ are less importance in predict the ‘cnt’ of Rental Bikes. So these variable are removed while performing model development.

**2.5. Features Scaling**

The word “normalization” is used informally in statistics, and so the term normalized data can have multiple meanings. In most cases, when you normalize data you eliminate the units of measurement for data, enabling you to more easily compare data from different places. Some of the more common ways to normalize data include:

Transforming data using a [z-score](http://www.statisticshowto.com/probability-and-statistics/z-score/) or [t-score](http://www.statisticshowto.com/probability-and-statistics/t-distribution/t-score-formula/). This is usually called standardization. In the vast majority of cases, if a statistics textbook is talking about normalizing data, then this is the definition of “normalization” they are probably using.

[Rescaling data](http://www.statisticshowto.com/what-is-rescaling-data/) to have values between 0 and 1. This is usually called feature scaling. One possible formula to achieve this is.

[http://www.statisticshowto.com/wp-content/uploads/2015/11/normalize-data.png](http://www.statisticshowto.com/wp-content/uploads/2015/11/normalize-data.png)

**CHAPTER 3**

**MODELLING**

**3.1 Model building:**

In this case we have to predict the count of bike renting according to environmental and seasonal condition. So the target variable here is a continuous variable. For continuous we can use various Regression models. Model having less error rate and more accuracy will be our final model.

Models built are

1. Decision Tree
2. Random Forest
3. Linear regression

.

**1. Decision tree:**

A decision tree is a supervised machine-learning model used to predict a target by learning decision rules from features.

For this model we have divided the dataset into train and test part using random sampling. Where train contains 80% data of data set and test contains 20% data and contains 10 variables where 10th variable is the target variable.

**In R:**

train\_index = sample(1:nrow(day), 0.8 \* nrow(day))

train = day[train\_index,]

test = day[-train\_index,]

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

dt\_predictions=predict(dt\_model,test[,-10])

df = data.frame("actual"=test[,10], "pred"=dt\_predictions)

regr.eval(trues = test[,10], preds = dt\_predictions, stats = c("mae","mse","rmse","mape"))

MAPE = function(actual, pred){

print(mean(abs((actual - pred)/actual)) \* 100)

}

MAPE(test[,10], dt\_predictions)

Output (in R) : MAPE = 17.47%, Accuracy = 82.53%

In python:

#Import Libraries for decision tree

from sklearn.tree import DecisionTreeClassifier

from sklearn.tree import DecisionTreeRegressor

model\_DT=DecisionTreeClassifier(criterion='entropy')

train,test = train\_test\_split(df, test\_size = 0.2, random\_state = 123)

dt\_model = DecisionTreeRegressor(random\_state=123).fit(train.iloc[:,0:9], train.iloc[:,9])

dt\_predictions = dt\_model.predict(test.iloc[:,0:9])

vdf\_dt = pd.DataFrame({'actual': test.iloc[:,9], 'pred': dt\_predictions})

df\_dt.head(df\_dt = pd.DataFrame({'actual': test.iloc[:,9], 'pred': dt\_predictions})

df\_dt.head()

O/p: actual pred

225 3820.0 4649.0

429 3333.0 3624.0

716 4585.0 3510.0

653 5875.0 7733.0

341 3322.0 2431.0

#Function for Mean Absolute Percentage Error

def MAPE(y\_actual,y\_pred):

mape = np.mean(np.abs((y\_actual - y\_pred)/y\_actual))\*100

return mape

MAPE(test.iloc[:,9],dt\_predictions)

# MAPE: 16.825% #Accuracy: 83.18%

**2. Random Forest:**

The same training set and test data set is used for random forest implementation.

rf\_model = randomForest(cnt~., data = train, ntree = 500)

rf\_predictions = predict(rf\_model, test[,-10])

df = cbind(df,rf\_predictions)

regr.eval(trues = test[,10], preds = rf\_predictions, stats = c("mae","mse","rmse","mape"))

MAPE(test[,10], rf\_predictions)

Output: Error rate = 11.69% and Accuracy = 88.31%

In python:

#Divide data into train and test

X = df.values[:,7:10]

Y = df.values[:,7:10]

X\_train,y\_train,X\_test,y\_test = train\_test\_split( X, Y, test\_size = 0.2)

RF\_model = RandomForestRegressor(n\_estimators = 1000, random\_state = 1337)

RF\_model.fit(X\_train, X\_test); # Train the model on training data

predictions = RF\_model.predict(y\_train) # Use the forest's predict method on the test data

errors = abs(predictions - y\_test) # Calculate the absolute errors

mape = 100 \* (errors / y\_test) # Calculate mean absolute percentage error (MAPE)

accuracy = 100 - np.mean(mape) # Calculate and display accuracy

print('Accuracy:', round(accuracy, 2), '%.')

Output: Accuracy: 84.09 %.

* 1. **Linear Regression:**

The same training set and test data set is used for linear regression implementation.

In R:

lr\_model = lm(formula = cnt~., data = train)

lr\_predictions = predict(lr\_model, test[,-10])

df = cbind(df,lr\_predictions)

head(df)

regr.eval(trues = test[,10], preds = lr\_predictions, stats = c("mae","mse","rmse","mape"))

MAPE(test[,10], lr\_predictions)

plot(test$cnt,type="l",lty=2,col="green")

lines(lr\_predictions,col="blue")

predict(lr\_model, test[2,])

Output of Linear Regression:

# Error rate of Linear Regression = 12.17%

# Accuracy of Linear Regression = 87.83%

In python:

import statsmodels.api as sm

from sklearn.metrics import mean\_squared\_error

from sklearn.linear\_model import LinearRegression

model\_lr=LinearRegression()

model\_lr.fit(X\_train,X\_test)

predictions\_lr=model\_lr.predict(y\_train)

predictions\_lr

print(model\_lr.intercept\_)

Output: [1.11022302e-16 8.07964806e-13 1.81898940e-12]

print(model\_lr.coef\_)

[[ 1.00000000e+00 -3.24779067e-17 -9.27324356e-20] [-1.04377433e-12 1.00000000e+00 -1.35152294e-17] [-1.26549751e-12 -3.79309817e-13 1.00000000e+00]]

from sklearn import metrics

metrics.r2\_score(y\_test,predictions\_lr)

Output = 1.0

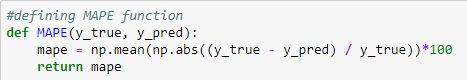
The closer the value is to 1, the better the fit, or relationship, between the two factors. The coefficient of determination is the square of the correlation coefficient, also known as "R," which allows it to display the degree of linear correlation between two variables.

#### 3.2. Model Evaluation:

Predictive performance can be measured by comparing predictions of the models with real values of the target variables, and calculating some average error measure.

* + 1. **Mean Absolute Percentage Error (MAPE)**

MAPE is one of the error measures used to calculate the predictive performance of the model. We will apply this measure to our models that we have generated in the previous sections



In above function y\_true is the actual value and y\_pred is the predicted value. It will provide the error percentage of model.

MAPE value in Python are as follows:

For Random Forest: **Accuracy: 84.09 %**.

For Decision Tree: **Accuracy = 83.18%**

MAPE(test.iloc[:,9],dt\_predictions) # MAPE: 16.825% #Accuracy: 83.18%

For Linear regression: r square = 1.0

**3.3. Model Selection**

As we predicted counts for Bike Rental using three Models Decision Tree, Random Forest and Linear Regressions, MAPE is less for random forest among the three algorithms and accuracy obtained is 88.31%

**Conclusion**: Based on the values obtained, linear regression is the best suited for analysis for this dataset.