# *BIKE SHARING DEMAND ANALYSIS AND PREDICTION*

Presented by : Sushil Deshmukh

SUID: 578130808

# *TABLE OF CONTENTS*

# *About the data*

This bike sharing dataset can be found on Kaggle.

Bike sharing systems are a means of renting bicycles where the process of obtaining membership, rental, and bike return is automated via a network of kiosk locations throughout a city.

The train file and the test file are loaded into respective pandas dataframe for exploration. Here are some basic information about the data files.

1. There are total of 17377 rows in combined two files.

2. There are total of 12 columns or attributes that are variables in this dataset.

The URL for the dataset is as follows:

https://www.kaggle.com/c/bike-sharing-demand

# *About the data*

The data generated by these systems makes them attractive for researchers because the duration of travel, departure location, arrival location, and time elapsed is explicitly recorded. Bike sharing systems therefore function as a sensor network, which can be used for studying mobility in a city. In this competition, participants are asked to combine historical usage patterns with weather data in order to forecast bike rental demand in the Capital Bikeshare program in Washington, D.C.

# About the data

Different data fields:

datetime - hourly date + timestamp

season - 1 = spring, 2 = summer, 3 = fall, 4 = winter

holiday - whether the day is considered a holiday

workingday - whether the day is neither a weekend nor holiday

weather - 1: Clear, Few clouds, partly cloudy, Partly cloudy

2: Mist + Cloudy, Mist + Broken clouds, Mist + Few clouds, Mist
3: Light Snow, Light Rain + Thunderstorm + Scattered clouds, Light Rain + Scattered clouds
4: Heavy Rain + Ice Pallets + Thunderstorm + Mist, Snow + Fog

temp - temperature in Celsius

atemp - "feels like" temperature in Celsius

humidity - relative humidity

windspeed - wind speed

casual - number of non-registered user rentals initiated

registered - number of registered user rentals initiated

count - number of total rentals

# *NA values and Outliers*

For EDA and visualizations, identifying and removing the NA and outliers are a Must.

For NA: We can see there are no NA values in the Dataset.

checking for NA values in the Data

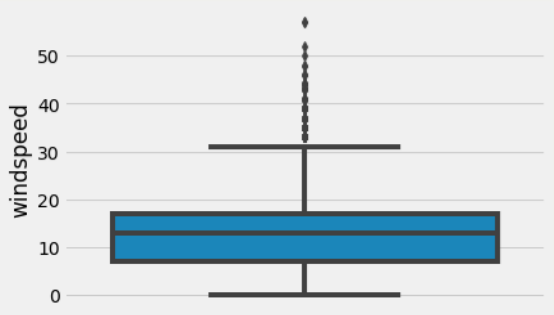
```
In [123…  df.isnull().sum()

Out[123…  datetime      0
          season        0
          holiday       0
          workingday    0
          weather       0
          temp          0
          atemp         0
          humidity      0
          windspeed     0
          casual        0
          registered    0
          count         0
          dtype: int64
```

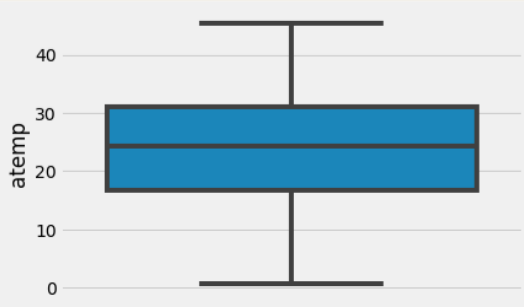
# *Na values and Outliers*

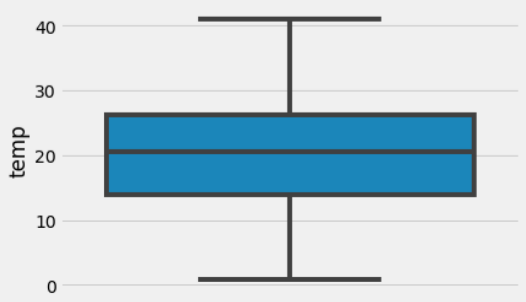
Outliers are those values of the data fields which are far away from a normal value range present in the dataset. This can hamper the model predictions. Hence, they need to be identified and removed. In this case we will be identifying the outliers by plotting box plots of the numerical attributes in the dataset.

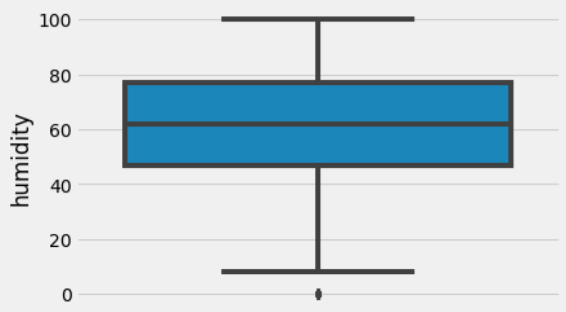
# *Na values and Outliers*

We can see that there are outliers in the windspeed and the humidity plot



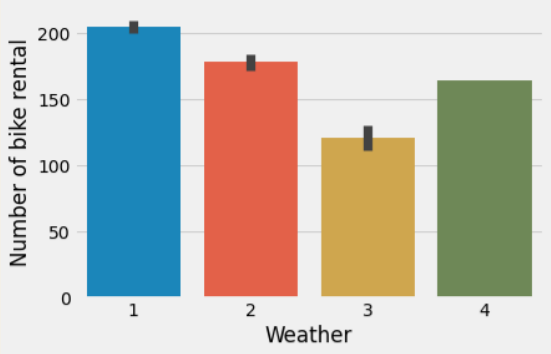



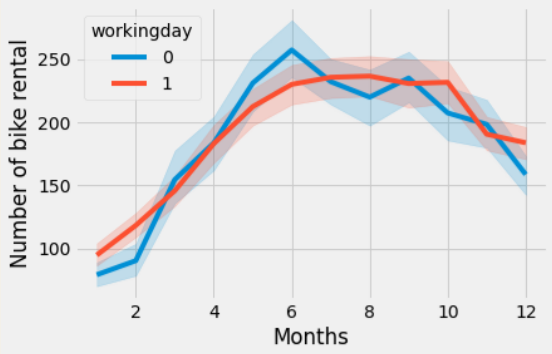



# *Visualizations*

Figure to the left shows max bikes rented in weather 1 which is clear weather

Figure to the right shows max bikes rented in periods of 6-8th month for when working day=0 and month 6$^{th}$ when working day=1.

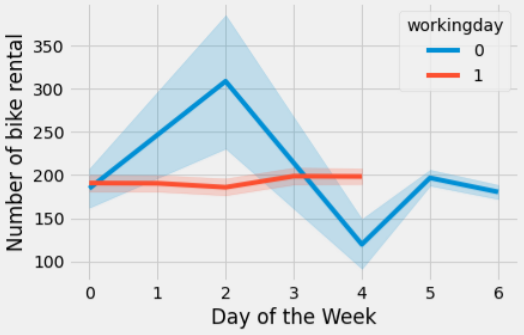


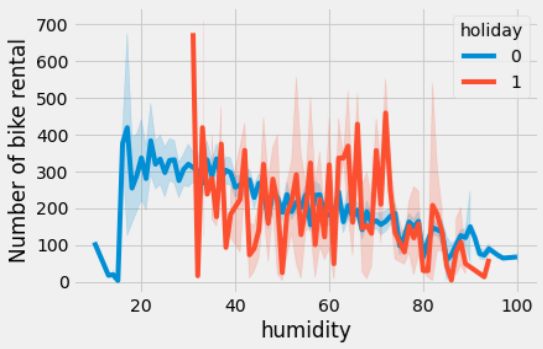



# *Visualizations*

Figure to the left shows max bikes rented in 1st -3rd day of the week.

Figure to the right shows max bikes rented decreases with increase in humidity.

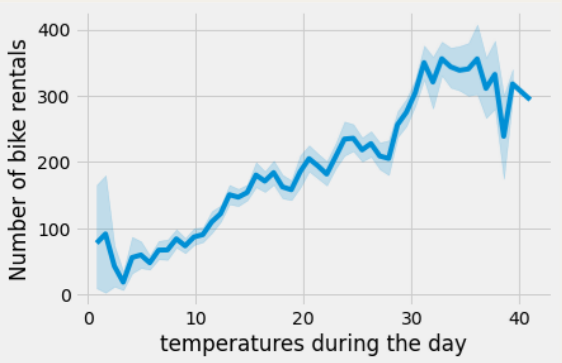


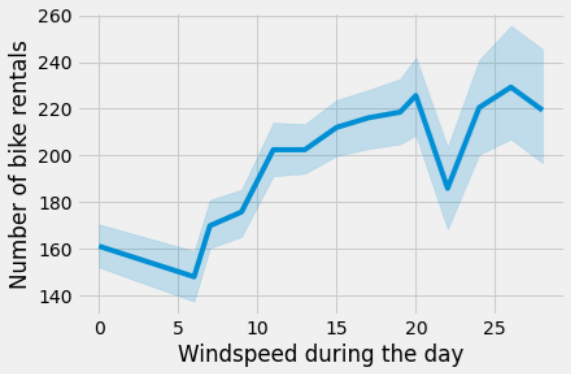



# *Visualizations*

Figure to the left shows number of bikes rented increased with temperature for a certain point

Figure to the right shows max bikes rented when windspeed between 5-20 value.



# *Model Selection and training*

For models, after comparing different RMSE scores in supervised learning models, the best score was for random forest regression.

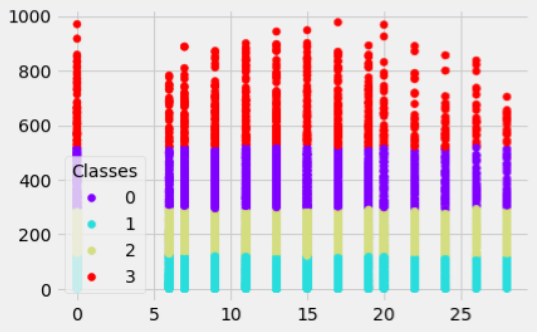
Here we have generated a model and then calculated the mean squared error and also the R2 score for the model.

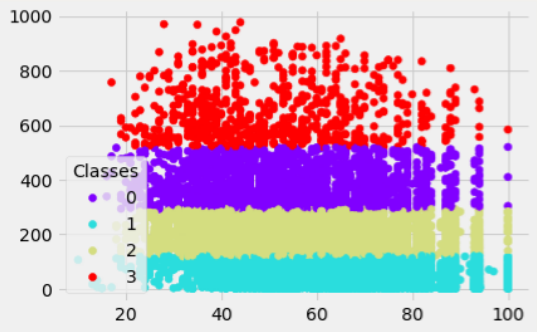
```
In [36]: x_train,x_test,y_train,y_test=sklearn.model_selection.train_test_split(df.drop('count',axis=1),df['count'],test_size=0.

In [37]: no_of_test=[500]
         params_dict={'n_estimators':no_of_test,'n_jobs':[-1],'max_features':["auto",'sqrt','log2']}
         clf_rf=GridSearchCV(estimator=sklearn.ensemble.RandomForestRegressor(),param_grid=params_dict,scoring='neg_mean_squared
         clf_rf.fit(x_train,y_train)
         pred=clf_rf.predict(x_test)

         Calculating the scores of r2 and mean squared log error for the model.

In [38]: print((np.sqrt(mean_squared_log_error(pred,y_test))))
         print(sklearn.metrics.r2_score(pred,y_test))

         0.32425930706970135
         0.9421812719150662
```

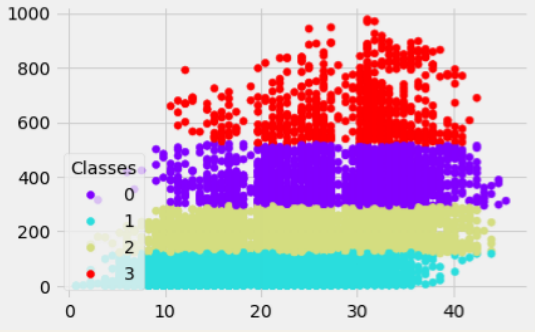
# *Unsupervised Model*

We are using a K-Means clustering models to cluster the given dataset. We first need to find the ideal value of R to be considered . For this we use the elbow method to determine the value. We can see the elbow bent at 4 , hence the value of k=4

# *Unsupervised model*



# *Unsupervised Model*

For finding the efficiency of the unsupervised model, we look at the silhouette score of the model. This score ranges from -1 to 1. this score determines how well the clusters are by looking at the density of the clusters and how well they are separated.

```
In [54]: score = silhouette_score(df, kmeans.labels_, metric='euclidean')
         print(score)

         0.5331677132846561
```

This shows a score of 0.53 which is fairly good score for a dataset not tailored for unsupervised learning.

# *Conclusions*

From the data that we have been using, we have successfully predicted the bikes rented using the Random Forest Regression technique. We have an efficiency of 94.19 % with this model which is a good score to predict the needed variable.

We have also successfully classified different clusters using an unsupervised learning algorithm which is the K-Means clustering and successfully identified clusters using the model.

# *References*


https://www.kaggle.com/c/bike-sharing-demand/data : Kaggle URL for Data information

https://dzone.com/articles/kmeans-silhouette-score-explained-with-python-exam : K-Means Clustering

https://pandas.pydata.org/pandas-docs/stable/reference/frame.html : Pandas Documentation

https://scikit-learn.org/stable/index.html : Scikit-Learn for modelling documentation for python

https://matplotlib.org/stable/index.html : Matplotlib documentation