Bike Sharing Prediction

by Surya

The objective of this project is to build a model that predicts the hourly count of rental bikes.

Data Wrangling

We are visualizing a sample of the data and the data types of the variables provided by the bikeshare system below,

	instant	dteday	season	yr	mnth	hr	holiday	weekday	workingday	weathersit	temp	atemp	hum	windspeed	casual	registered	cnt
0	1	2011-01-01	1	0	1	0	0	6	0	1	0.24	0.2879	0.81	0.0	3	13	16
1	2	2011-01-01	1	0	1	1	0	6	0	1	0.22	0.2727	0.80	0.0	8	32	40
2	3	2011-01-01	1	0	1	2	0	6	0	1	0.22	0.2727	0.80	0.0	5	27	32
3	4	2011-01-01	1	0	1	3	0	6	0	1	0.24	0.2879	0.75	0.0	3	10	13
4	5	2011-01-01	1	0	1	4	0	6	0	1	0.24	0.2879	0.75	0.0	0	1	1

```
season          int64
yr              int64
mnth            int64
hr              int64
holiday         int64
weekday         int64
workingday      int64
weathersit      int64
temp          float64
atemp         float64
hum           float64
windspeed     float64
casual          int64
registered      int64
cnt             int64
dtype: object
```

Inferences

- Converting categorical variables (season, yr, mnth, hr, holiday, weekday, workingday, weathersit) into its appropriate data type
- Removing dteday column as it does not provide any additional information
- Removing instant column as it is an index variable

Exploratory Data Analysis

Trying to understand the overall spread of the numerical variables,

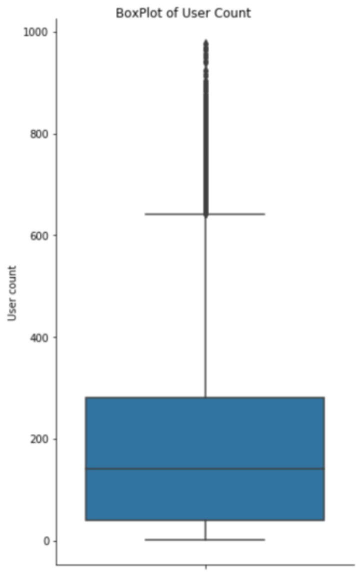
	temp	atemp	hum	windspeed	casual	registered	cnt
count	17379.000000	17379.000000	17379.000000	17379.000000	17379.000000	17379.000000	17379.000000
mean	0.496987	0.475775	0.627229	0.190098	35.676218	153.786869	189.463088
std	0.192556	0.171850	0.192930	0.122340	49.305030	151.357286	181.387599
min	0.020000	0.000000	0.000000	0.000000	0.000000	0.000000	1.000000
25%	0.340000	0.333300	0.480000	0.104500	4.000000	34.000000	40.000000
50%	0.500000	0.484800	0.630000	0.194000	17.000000	115.000000	142.000000
75%	0.660000	0.621200	0.780000	0.253700	48.000000	220.000000	281.000000
max	1.000000	1.000000	1.000000	0.850700	367.000000	886.000000	977.000000

INFERENCES

- Windspeed has a mean of 0.19 indicating an imbalance.

TARGET VARIABLE ANALYSIS

Generating a boxplot on the target variable (cnt) to understand how it is spread across,

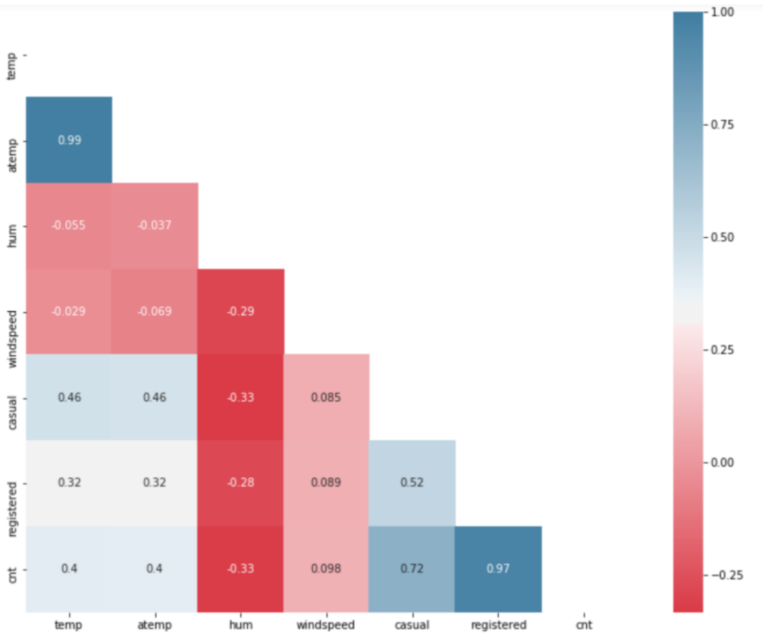


INFERENCES

- We can see from the visual the presence of outliers.
- Removing outliers from the target variable that are beyond 2.5 standard deviations.

CORRELATION MATRIX

Constructing a correlation matrix to understand the correlation between the variables,

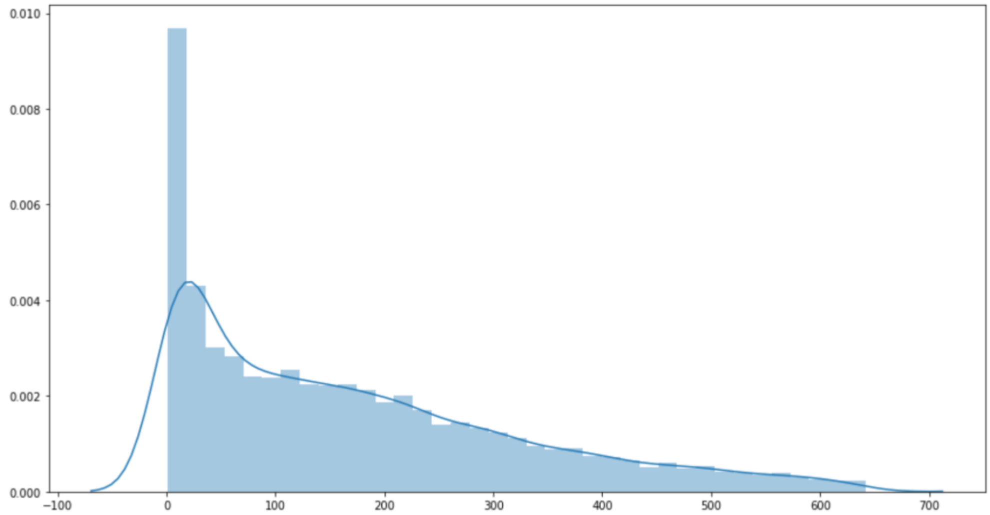


INFERENCES

- This visualization helps us understand the existence of multicollinearity between variables 'temp' and 'atemp'. Therefore, one among the two variables is removed.
- We are also able to infer that the variables 'casual' and 'registered' are highly correlated with the target variable (user count). This is because, the sum of 'casual' and 'registered' is the target variable. Therefore, one among the two variables is removed in order to prevent data leakage during model building.
- The feature windspeed would not be very useful to the target variable due to its weak correlation.

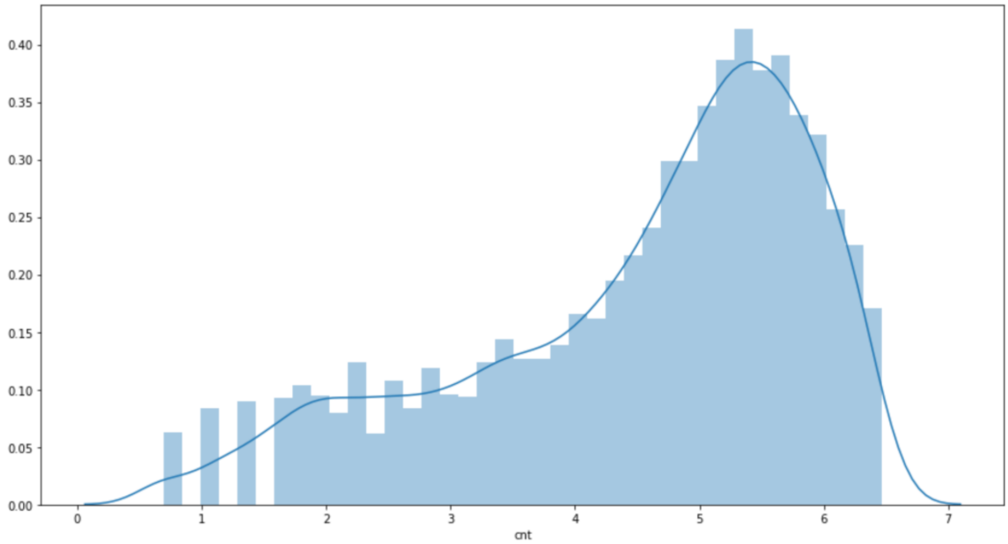
ANALYSIS OF DISTRIBUTION ON THE TARGET VARIABLE

Generating a plot to understand the distribution of the target variable,



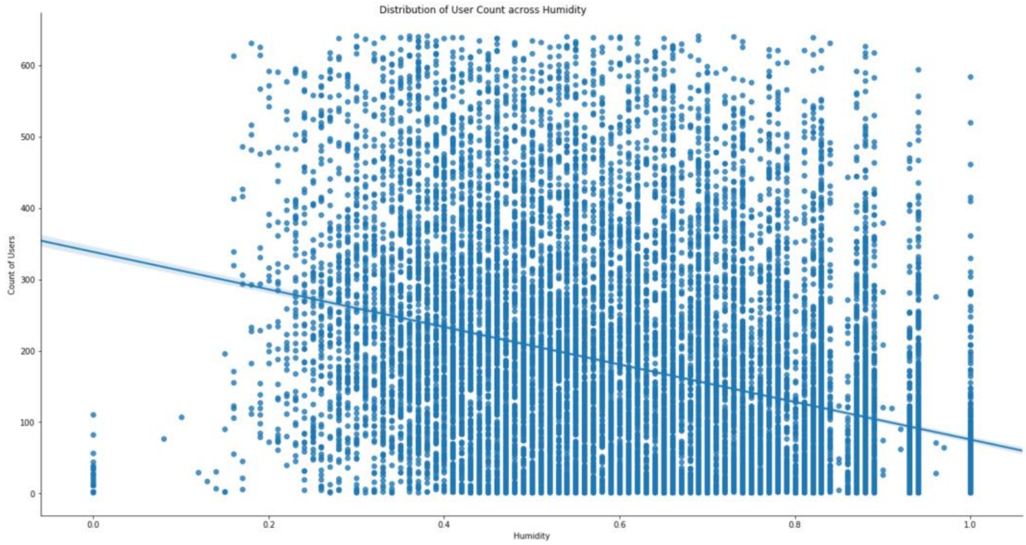
INFERENCES

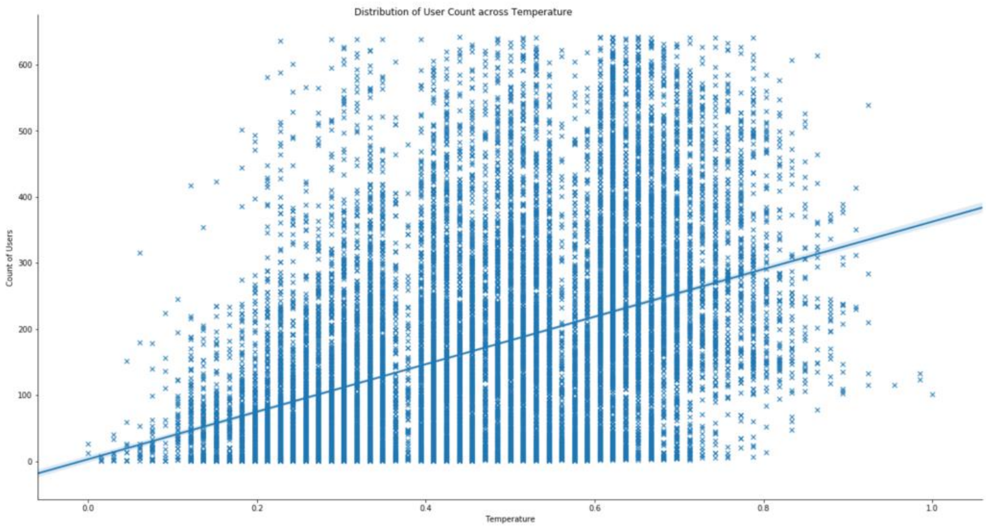
- We can view that the distribution is skewed to the right.
- Applying a log transformation to fix the skewness.



ANALYSIS ON NUMERICAL VARIABLES

We can see the existence of relationships between variables using the regression plots below,



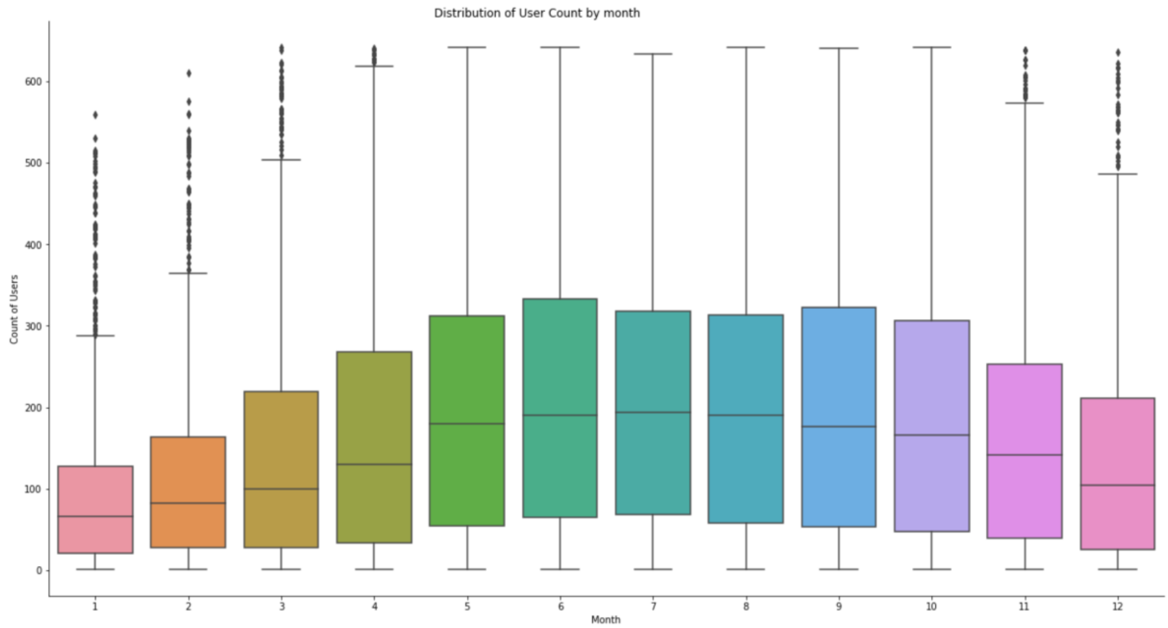


INFERENCES

- We can fully see the existence of the correlation relationship between the target variable and the variables containing the temperature and humidity. It would help the model to an extent.

ANALYSIS ON CATEGORICAL VARIABLES

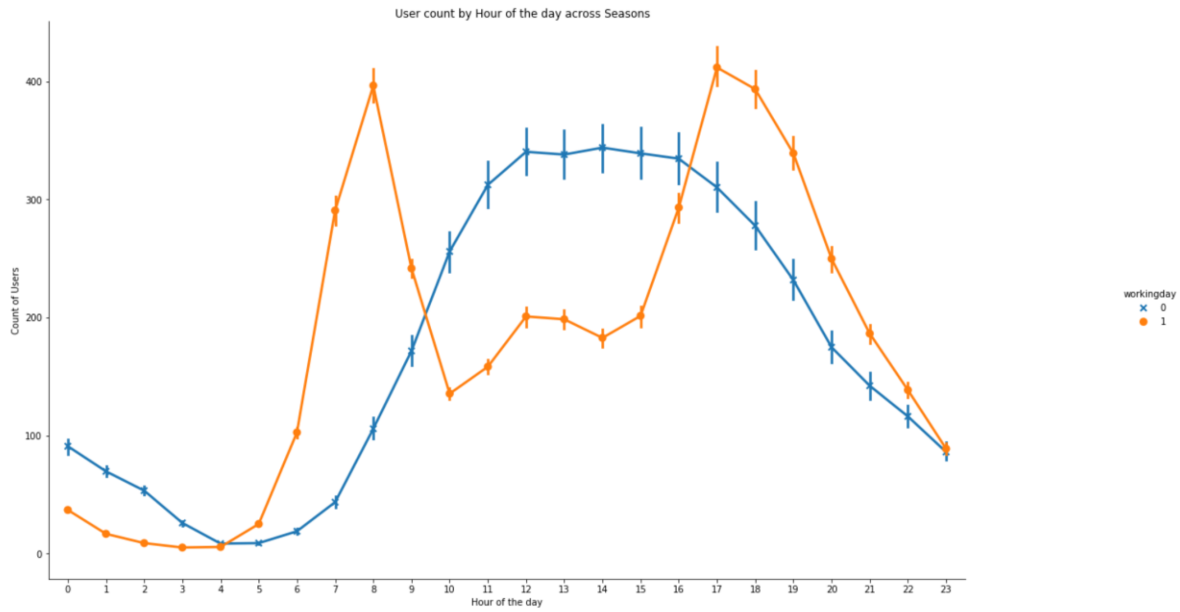
The box plot below is generated to understand how the count is spread across different months,



Inferences

- We can infer that the average user count is comparatively high in Summer (May, June and July).

The following plot helps us understand more about the spread of the user count across various hours of the day,



Inferences

- We can see from the above plot that the user count is generally higher around 7 to 8 A.M and around 5 - 6 P.M when it is a working day, we can attribute this count to the office and school going user base.
- The trend for a non-working day is slightly different and peaks out between 12 to 2 P.M.

Model Selection

It is better to go with Non-Parametric models rather than parametric models because it does not make any strong assumptions. Non-Parametric models are more robust to outliers, nonlinear relationships, and does not depend on many population distribution or assumptions and therefore, it is more suitable for the current dataset and the problem at hand.

- The algorithms being considered are Random Forest, Gradient Boosting and XG Boosting.
- Hyper Parameters are being tuned by using grid search cross validation method.
    - Grid search helps us identify the best parameters for the model to use,

```
The best parameters for Random Forest are :
{'max_features': 'auto', 'n_estimators': 750, 'max_depth': 25}

 The best parameters for Gradient Boosting are :
{'subsample': 0.8, 'learning_rate': 0.1, 'min_samples_leaf': 50, 'n_estimators': 250, 'min_samples_split': 300, 'max_feature
s': 'auto', 'max_depth': 20}

The best parameters for XGBoosting are :
{'subsample': 0.4, 'learning_rate': 0.1, 'min_samples_leaf': 50, 'n_estimators': 150, 'min_samples_split': 150, 'max_feature
s': 'auto', 'max_depth': 7}
```

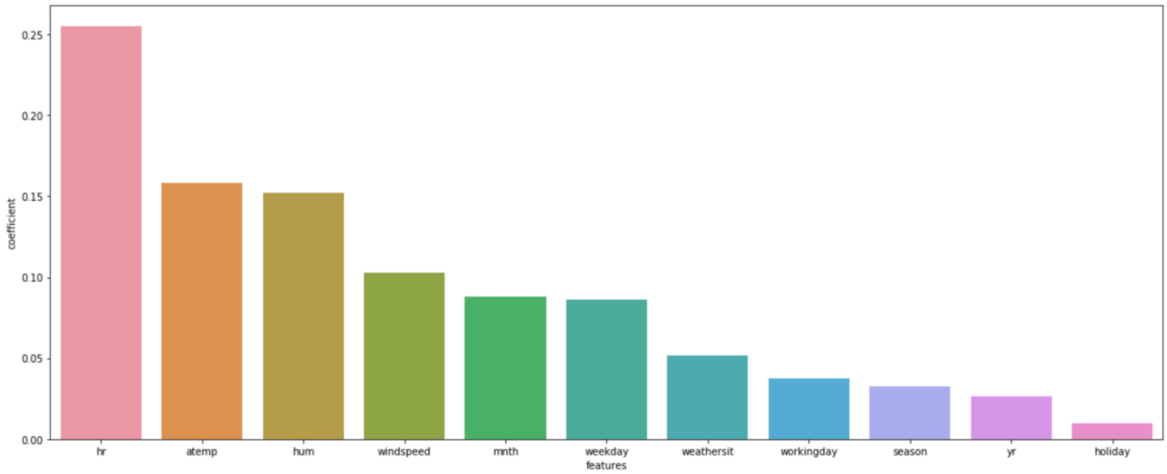
- Cross validation is applied in order to better estimate the skill of the model and in determining the best model out of the combinations being tested.

Finally, after examining the various outcomes, we can understand that the best model for this problem is the Gradient Boosting model with the following parameters,

```
 The best parameters for Gradient Boosting are :
{'subsample': 0.8, 'learning_rate': 0.1, 'min_samples_leaf': 50, 'n_estimators': 250, 'min_samples_split': 300, 'max_feature
s': 'auto', 'max_depth': 20}
```

Feature Importance

The plot displays the features and its importance to the target variable while using the gradient boosting model.

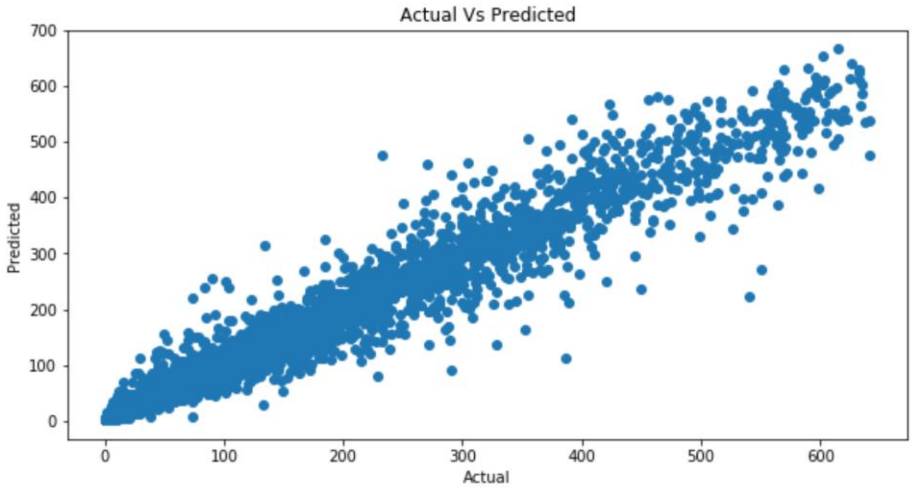


Results

The selected gradient boosting model offers the best result in terms of Mean Absolute Error.

```
   Mean Absolute Error              Model
0            24.194083      Random Forest
1            23.559362  Gradient Boosting
2            23.630971         XGBoosting
```

We can visualize the selected model's results in terms of a scatter plot of the actuals Vs predicted values,



Production Code

PICKLE

- Pickle is a technique through which we can serialize objects in python.
- Therefore, this technique is applied here to serialize the machine learning model and save this format to a file. (This part is implemented on the 'Bike Sharing Analysis' file)
- We can then load this saved file to deserialize the model and use the same to make new predictions. (This part is implemented on the 'Bike Sharing Analysis_Main' file)

There are other techniques to make codes production friendly as well,

DOCKER

- Dockers basically allow us to package and run applications on environments called containers.
- Containers are more efficient in production environment because they allow continuous improvement/continuous deployment.
- In order to achieve this,
    - we need to contain the model we have built, requirements and the training data in a docker file.
    - Create and push the docker image to our desired account.
    - Now, we can just get the image from the account and run the container.