

Machine Learning Rent IT!

1.1 PROBLEM STATEMENT

The project is about a bike rental company who has its historical data, and now our objective of this Project is to predict the bike rental count on a daily basis, considering the environmental and seasonal settings. These predicted values will help the business to meet the demand on those particular days by maintaining the amount of supply.

1.2 DATA

The given dataset contains 14 variables and 6000 observations. The “Rented Bike Count” is the target variable and remaining all other variables are the independent variables. Our objective is to develop a model that can determine the count for future test cases. And this model can be developed by the help of given data. A snapshot of the data is mentioned following.

	Date	Rented Bike Count	Hour	Temperature(°C)	Humidity(%)	Wind speed (m/s)	Visibility (10m)	Dew point temperature(°C)	Solar Radiation (MJ/m2)	Rainfall(mm)	Snowfall (cm)	Seasons	Holiday	Functioning Day
6943	16/09/2018	508.411177	7	21.9	95	1.3	424	21.0	0.02	0.0	0.0	Autumn	No Holiday	Yes
6218	17/08/2018	629.481163	2	22.8	50	1.4	2000	11.8	0.00	0.0	0.0	Summer	No Holiday	Yes
3037	6/4/2018	683.000000	13	7.9	58	4.3	488	0.1	1.88	0.0	0.0	Spring	No Holiday	Yes
2092	26/02/2018	46.000000	4	-2.3	69	1.1	1411	-7.2	0.00	0.0	0.0	Winter	No Holiday	Yes
1261	22/01/2018	281.000000	13	3.9	33	0.6	1249	-10.9	0.19	0.0	0.0	Winter	No Holiday	Yes

METHODOLOGY:

In Methodology following processes are followed:

• Pre-processing:

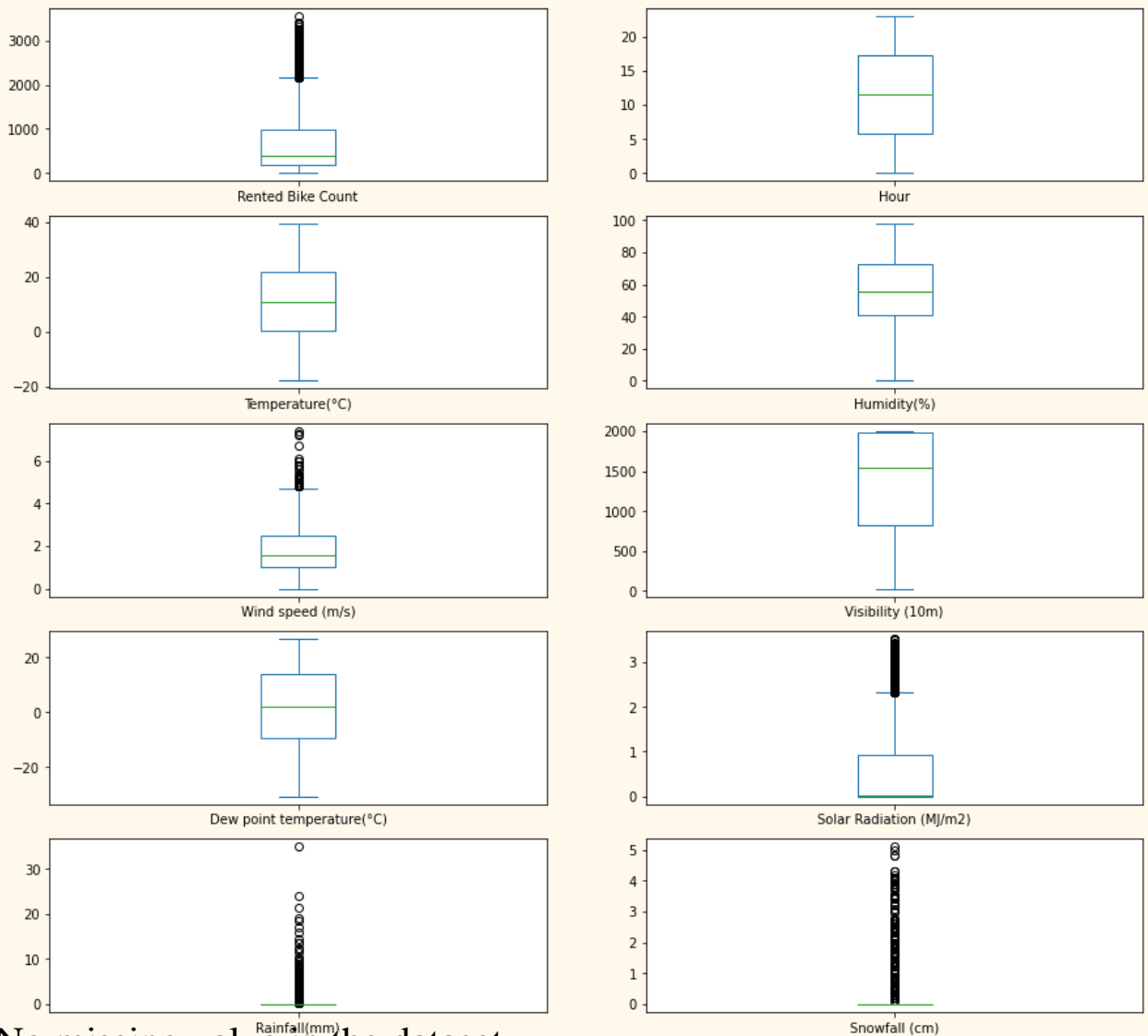
It includes missing value analysis, outlier analysis, feature selection and feature scaling.

• Model development:

It includes identifying suitable Machine learning Algorithms and applying those algorithms in our given dataset.

2.1 Pre-processing:

- Shuffle train
- We used the date to get the weekday and we extracted from the hours day and night
- We one hot encoded the categorical data because categorical data were not ordinal
- Handling outliers was not useful because it yielded to slightly weaker accuracy.



☐ No missing value in the dataset.

We did not use normalization because Random forest, catboost and lightgbm are not sensitive to the magnitude of variables.

☐ Correlation and Feature Selection:

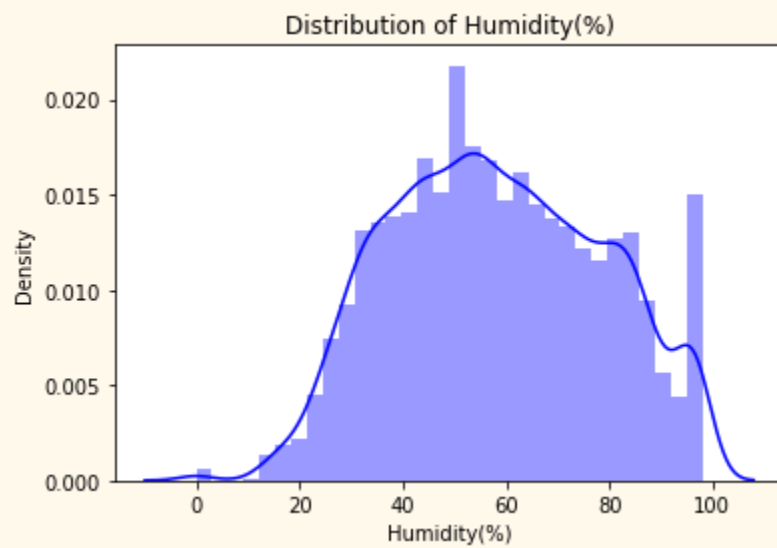
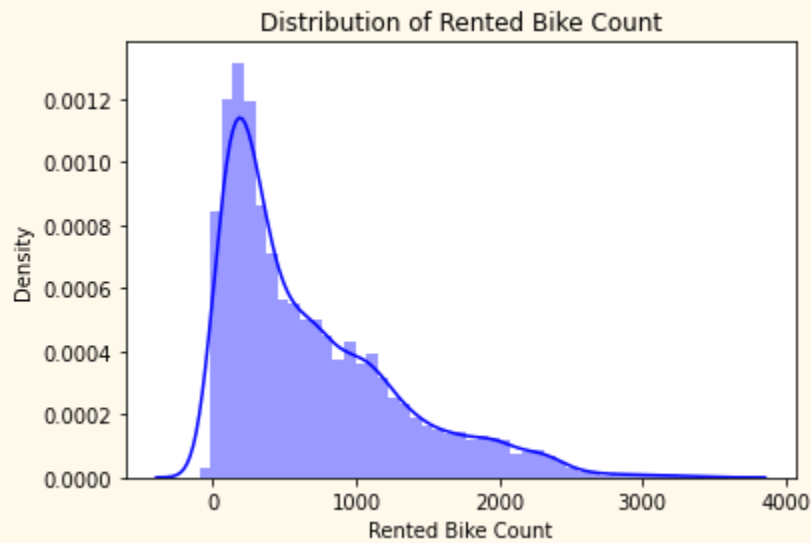
Rented Bike Count	1	0.6	-0.2	0.15	0.21	0.43	0.29	-0.15	0.031	-0.028	-0.19
Temperature(°C)	0.6	1	0.16	-0.036	0.035	0.91	0.35	-0.22	0.05	-0.013	-0.098
Humidity(%)	-0.2	0.16	1	-0.34	-0.54	0.54	-0.46	0.11	0.048	-0.037	0.24
Wind speed (m/s)	0.15	-0.036	-0.34	1	0.17	-0.18	0.33	-0.0036	-0.082	-0.022	-0.22
Visibility (10m)	0.21	0.035	-0.54	0.17	1	-0.18	0.15	-0.12	0.078	0.031	-0.024
Dew point temperature(°C)	0.43	0.91	0.54	-0.18	-0.18	1	0.094	-0.15	0.065	-0.029	0.025
Solar Radiation (MJ/m2)	0.29	0.35	-0.46	0.33	0.15	0.094	1	-0.072	-0.03	0.0083	-0.46
Snowfall (cm)	-0.15	-0.22	0.11	-0.0036	-0.12	-0.15	-0.072	1	0.055	-0.023	-0.011
Month	0.031	0.05	0.048	-0.082	0.078	0.065	-0.03	0.055	1	0.0092	2.6e-17
weekdays_weekend	-0.028	-0.013	-0.037	-0.022	0.031	-0.029	0.0083	-0.023	0.0092	1	2.9e-18
label_day_night	-0.19	-0.098	0.24	-0.22	-0.024	0.025	-0.46	-0.011	2.6e-17	2.9e-18	1
	Rented Bike Count	Temperature(°C)	Humidity(%)	Wind speed (m/s)	Visibility (10m)	Dew point temperature(°C)	Solar Radiation (MJ/m2)	Snowfall (cm)	Month	weekdays_weekend	label_day_night

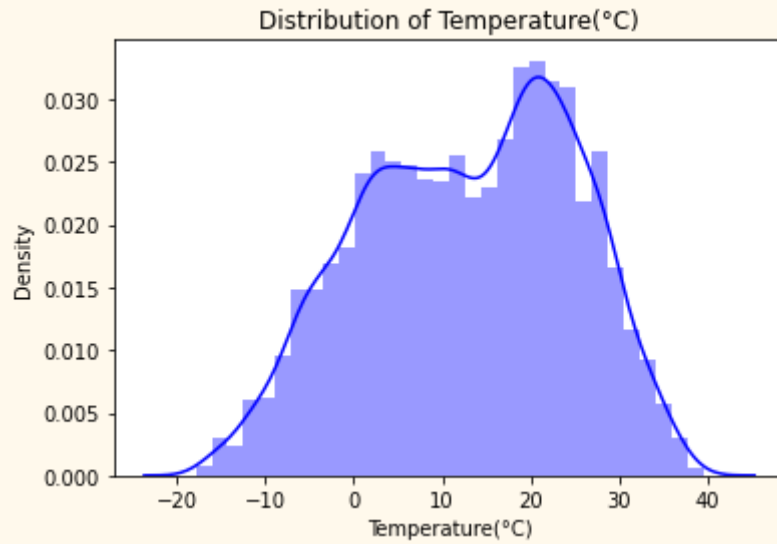
Observation:

1. Dew Temperature and temperature are highly correlated so we dropped Dew Temperature.
2. Snowfall and wind speed have very low correlation with the output

rented bike count so we dropped them as well.

Distribution of Data:





For Numerical Variables Range check:

	Rented Bike Count	Hour	Temperature(°C)	Humidity(%)	Wind speed (m/s)	Visibility (10m)	Dew point temperature(°C)	Solar Radiation (MJ/m2)	Rainfall(mm)	Snowfall (cm)
count	8760.000000	8760.000000	8760.000000	8760.000000	8760.000000	8760.000000	8760.000000	8760.000000	8760.000000	8760.000000
mean	686.918721	11.500000	12.882922	58.226256	1.724909	1436.825799	4.073813	0.569111	0.148687	0.075068
std	617.394430	6.922582	11.944825	20.362413	1.036300	608.298712	13.060369	0.868746	1.128193	0.436746
min	-94.000000	0.000000	-17.800000	0.000000	0.000000	27.000000	-30.600000	0.000000	0.000000	0.000000
25%	206.000000	5.750000	3.500000	42.000000	0.900000	940.000000	-4.700000	0.000000	0.000000	0.000000
50%	486.000000	11.500000	13.700000	57.000000	1.500000	1698.000000	5.100000	0.010000	0.000000	0.000000
75%	1022.250000	17.250000	22.500000	74.000000	2.300000	2000.000000	14.800000	0.930000	0.000000	0.000000
max	3556.000000	23.000000	39.400000	98.000000	7.400000	2000.000000	27.200000	3.520000	35.000000	8.800000

2.2 Model Development:

2.2.1 Model Selection:

Baseline models with its accuracy:

	Model	RMSLE
0	Linear Regression	402.701246
1	Ridge Regression	402.712448
2	Random Forest Regressor	155.370968
3	AdaBoost Regressor	366.772096
4	Bagging Regressor	167.727607
5	SVR	648.251821
6	KNeighbors Regressor	368.711271
7	XGBoost Regressor	190.044408
8	Gradient Boosting Regressor	189.260428

1. Optimized Models:

- **RandomForest:** We used five fold cross validation and grid search for best parameters.

Score on kaggle

Submission_R_o.csv	471.11679	473.22504	<input type="checkbox"/>
a month ago by NouranHanyMoh			
add submission details			

- **LightGBMRegressor:** We used ten fold cross validation and grid search for best parameters.

Score on kaggle


Submission_lgbm_with_preprocessing_full	384.69600	378.76989	<input type="checkbox"/>
a month ago by Sara Ahmed Abd-El fatah			
lgb with new two columns , full data,hot encoding func,holy			

- **GradientBoostingRegressor:** We used 5 folds cross validation with gridsearch


Submission_br	471.10354	476.26174	<input type="checkbox"/>
a month ago by NouranHanyMoh			
add submission details			

- **BaggingRegressor:** We used 5 folds cross validation



```
# Fit cv to the training set:
cv_br_model.fit(X_train, y_train)
y_pred_baggingreg = cv_br_model.predict(X_test)
y_pred_baggingreg_real = cv_br_model.predict(Xx_test)
# y_pred_log = np.expm1(y_pred)
# y_test_log = np.expm1(y_test)
score = np.sqrt(mean_squared_error(y_test,y_pred_baggingreg))
print("squared error: {}".format(score))
```

 Fitting 5 folds for each of 1 candidates, totalling 5 fits
squared error: 160.20562898316726

- **XGBRegressor:** Yielded to one the best submission optimized with grid search and five fold cross validation

add submission details			
SubmissionX	442.88887	448.24640	<input type="checkbox"/>
a month ago by NouranHanyMoh			
add submission details 			

- **CatBoostRegressor:** Yielded to one the best submission optimized with grid search and five fold cross validation

add submission details 			
Submission_cat	453.31136	456.49145	<input type="checkbox"/>
a month ago by NouranHanyMoh			
ensembling cat light xg			

- **Voting Regressor:** voting with lightgbm and catboost



Difference between models:

<u>Models</u>	<u>XGBoost</u>	<u>LightGBM</u>	<u>Catboost</u>
Handling Categorical Variables	Unlike CatBoost or LGBM, XGBoost cannot handle categorical features by itself, it only accepts numerical values similar to Random Forest. Therefore one has to perform various encodings like label encoding, mean encoding or one-hot encoding before supplying categorical data to XGBoost.	Similar to CatBoost, LightGBM can also handle categorical features by taking the input of feature names. It does not convert to one-hot coding, and is much faster than one-hot coding.	CatBoost has the flexibility of giving indices of categorical columns so that it can be encoded as one-hot encoding
Summary	Stands for eXtreme Gradient Boosting	is a gradient boosting framework based on decision trees	

		to increases the efficiency of the model and reduces memory usage.	
--	--	--	--

Further Steps to Improve the Above Models:

1. Using weighted average for models with best accuracies, with trial and error we got the best results with this method.

add submission details			
Submission_average_140	349.72604	350.11851	<input type="checkbox"/>
a month ago by NouranHanyMoh			
add submission details			
Submission_average_	348.50456	347.95855	<input type="checkbox"/>
a month ago by NouranHanyMoh			
add submission details			

Here is snapshot from the code:

```

141.766345630378

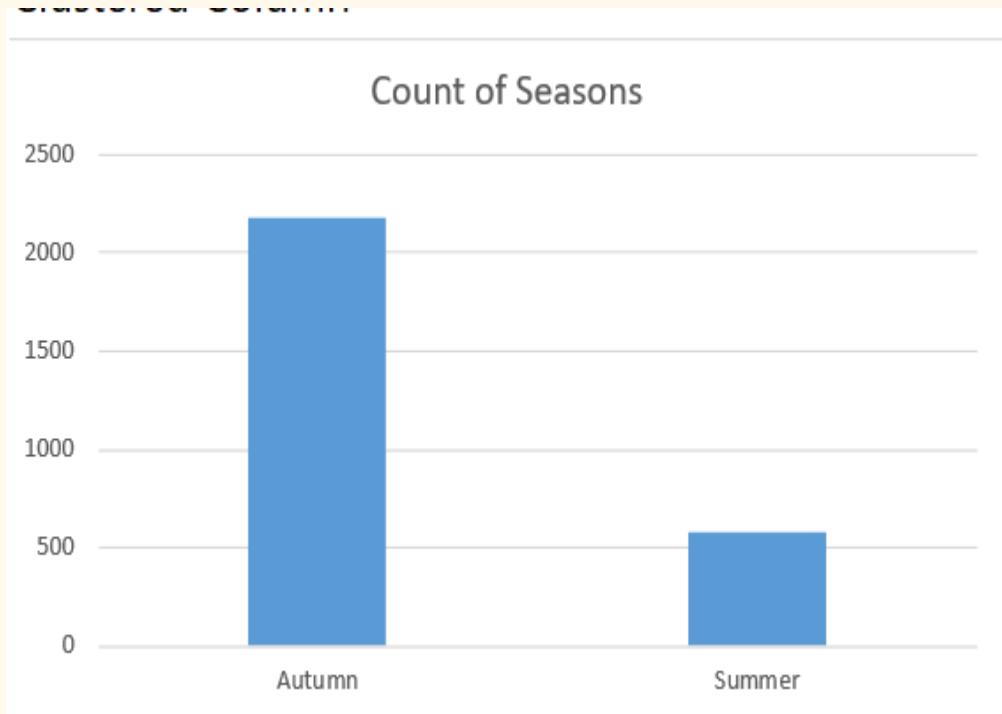
[ ] # y_avg =( y_pred_xgb*4+ y_pred_grad*5 + predicted_x_lgbm*12 + preds*9 + y_pred_randf*8) /(9+5+4+12+8)
y_avg =( y_pred_xgb*7+ predicted_x_lgbm*9 + preds*10 ) /(7+10+9)
score = np.sqrt(mean_squared_error(y_test, y_avg))
print(score)

137.96650031679187

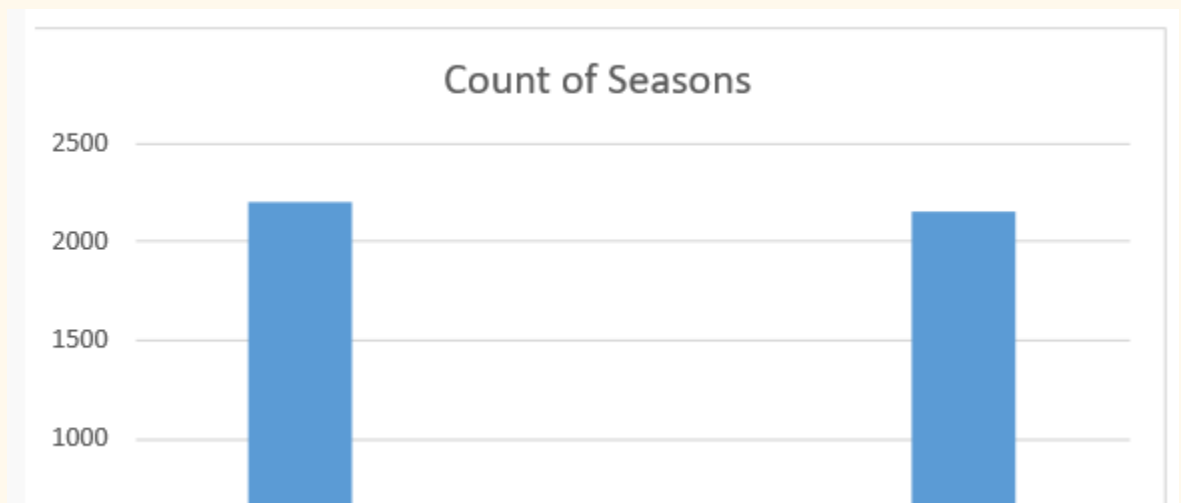
```

2. We merged the train data and test data due to data imbalance in season column i.e: train

Test season column:



Train season column



Result from merging:

It improved the generalization capability of the models and submission so we got a better score in the private leaderboard.

Conclusion:

- ☐ The Temperature, Hour & Humidity are the most important features that positively drive the total rented bikes count.
- ☐ This dataset is imbalanced so we had to merge the train and test.
- ☐ Catboost ,Xgboost and lightgbm performed the best out of all models so we used average weighting on them.
- ☐ There were features extracted from the date column which improved the accuracy.