Content

1. Problem statement
2. EDA/Feature analysis
3. Data Preprocessing
4. Machine Learning Models
5. Challenges
6. Conclusion

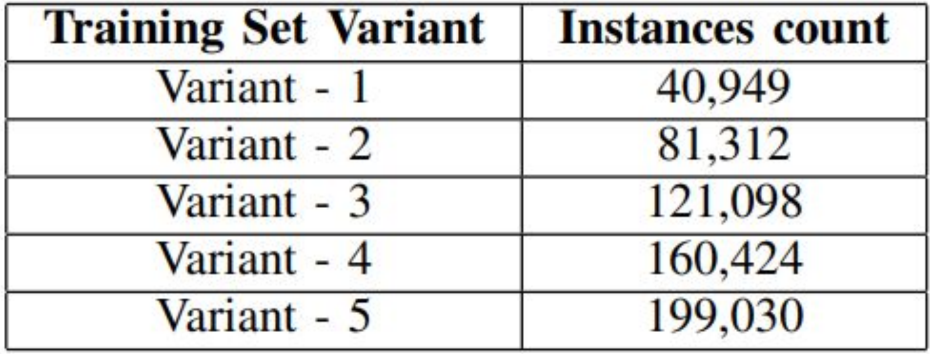
Problem Statement

- Prediction of comment volume traffic or simply to predict the number of comments a Facebook post would get within a certain number of hours after posting.

- Target variable was of continuous nature so it was a regression problem.

- We implemented linear and non-linear models such as Multiple Linear Regression, Regularized Regression, PCA, Random Forest and XGBoost to solve the problem.

Data Summary



Training Set Variant	Instances count
Variant - 1	40,949
Variant - 2	81,312
Variant - 3	121,098
Variant - 4	160,424
Variant - 5	199,030

Train dataset

- We had 5 variants of train dataset with different number of observations.

Test dataset

- We had 10 different test dataset with 100 observations each

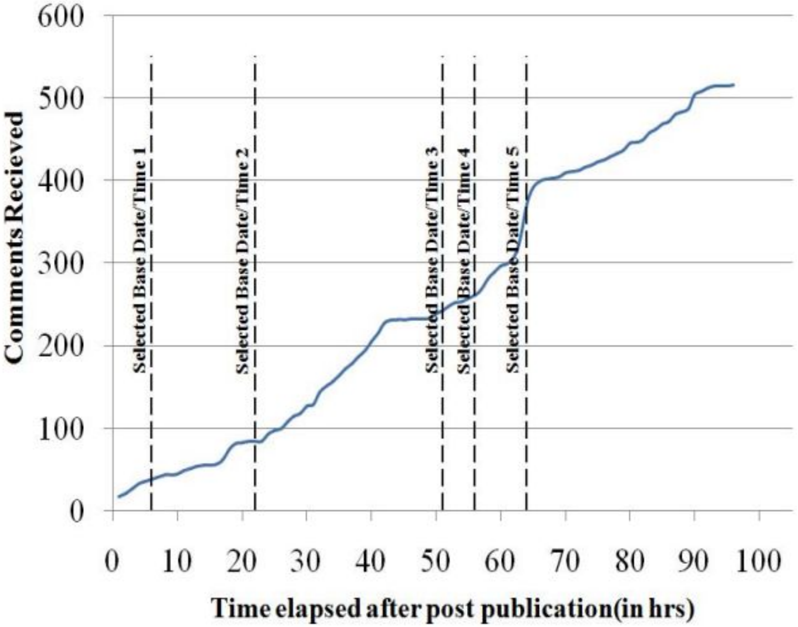
Features

- We had 53 predictor variables and 1 target variable (continuous)

Train Dataset Analysis

- Total 5 variants. Variant is defined as, how many instances of final training set is derived from single post of training set. This is done by selecting different base date/time for same post at random, process them individually.

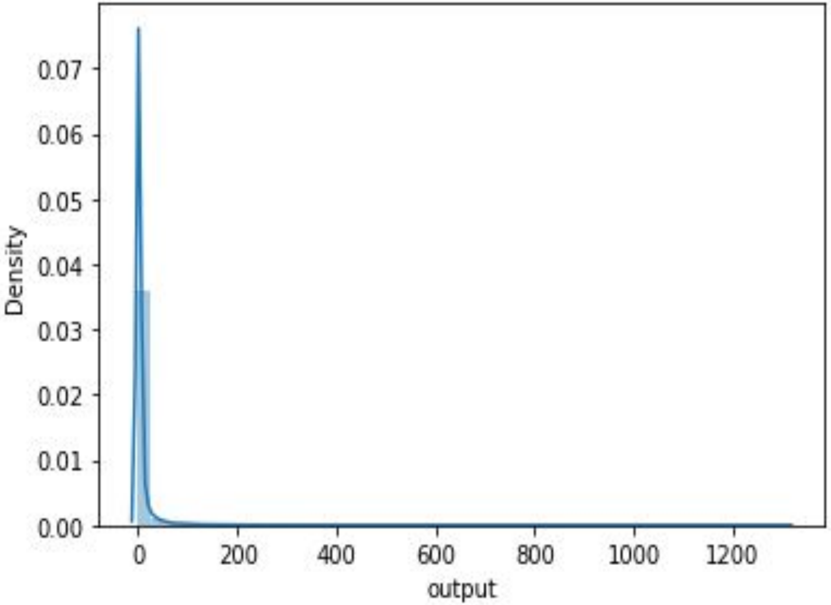
- No missing value was present

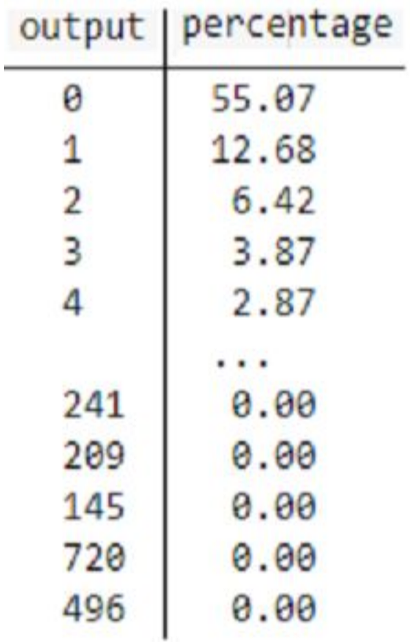


Feature Analysis (Target)

- 55% posts with nil comments

- High number of posts with very few comments





output	percentage
0	55.07
1	12.68
2	6.42
3	3.87
4	2.87
	...
241	0.00
209	0.00
145	0.00
720	0.00
496	0.00

Feature Analysis (Predictors)

Page Features - Page likes, Page type, Check-in Places, Page Returns

Comments Features w.r.t Time Intervals - CC1, CC2, CC3, CC4, CC5

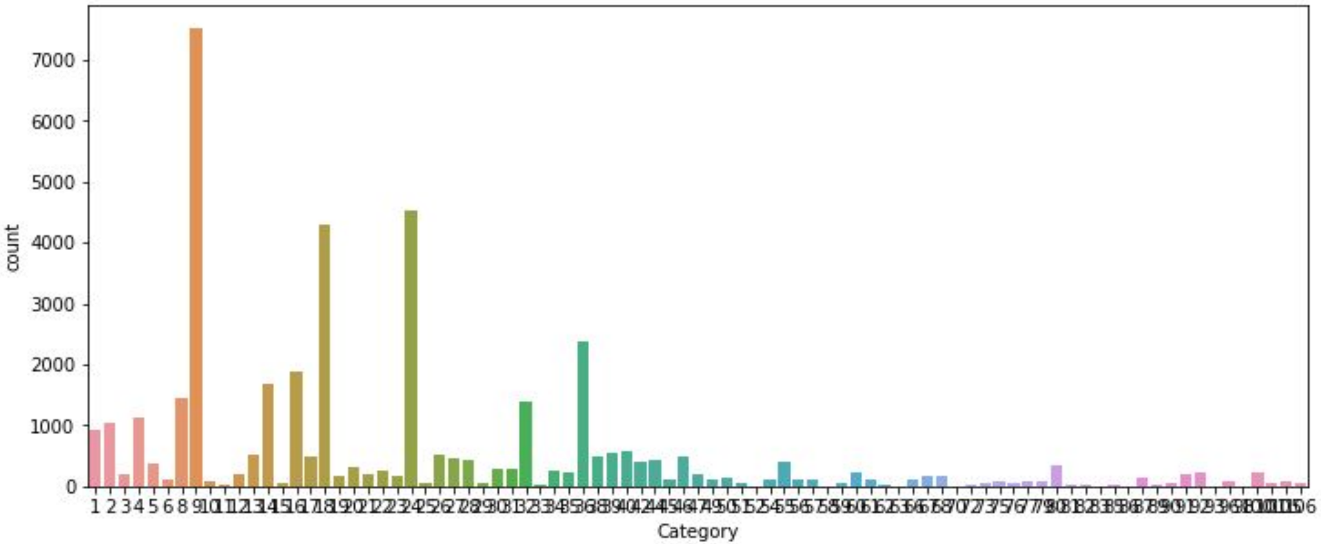
Derived Features - Min, Max, Avg, Med and Sd of CC features.

Date/Time Features - 7 post published day and 7 base date/time day

Other basic Features - Len of Post, Base Time, Total Hours (for which comments received), Post Share Count.

Feature Analysis (Predictors - Categorical)

- Page Type
- Too many labels with unknown mapping



- Unable to aggregate labels of same category like business, entertainment, political

- Removed the variable

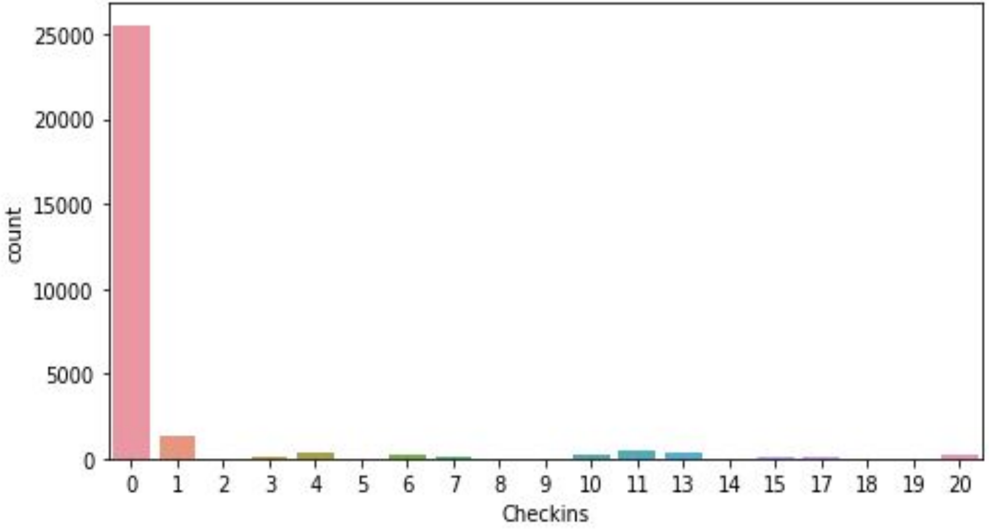
Feature Analysis (Predictors - Categorical)

Check-ins

1. Too many labels with unknown label encoding

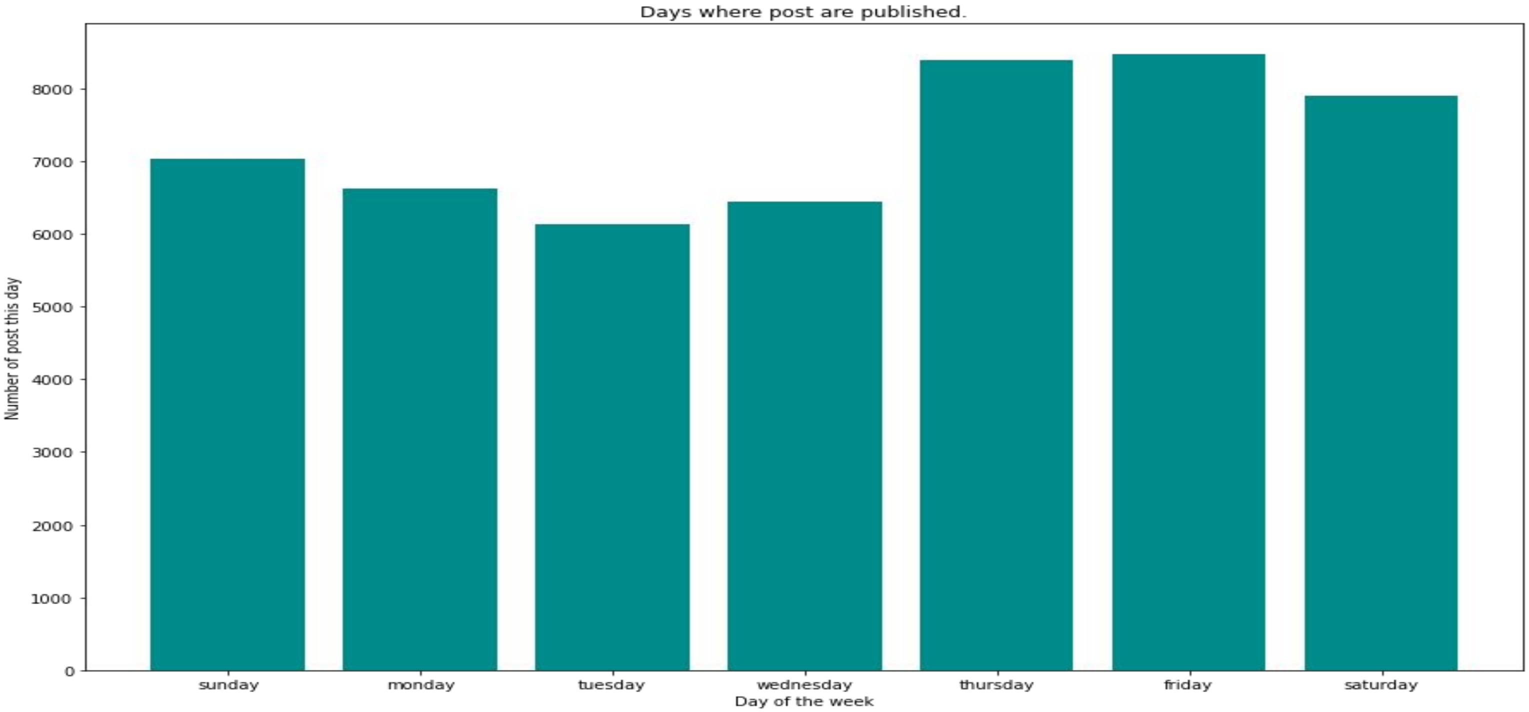
2. 62% with 0 label

3. Removed the variable



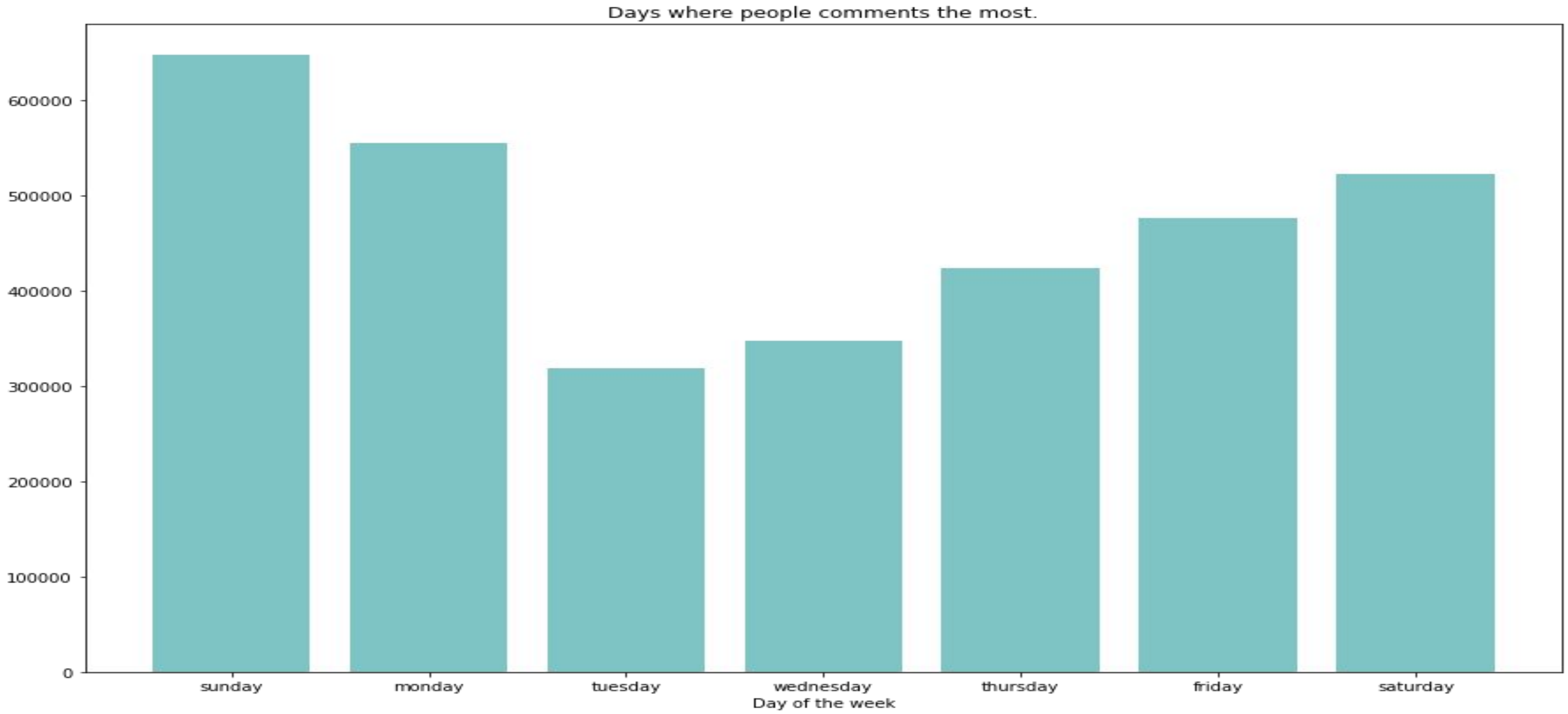
Feature Analysis (Predictors - Categorical)

Day when post published



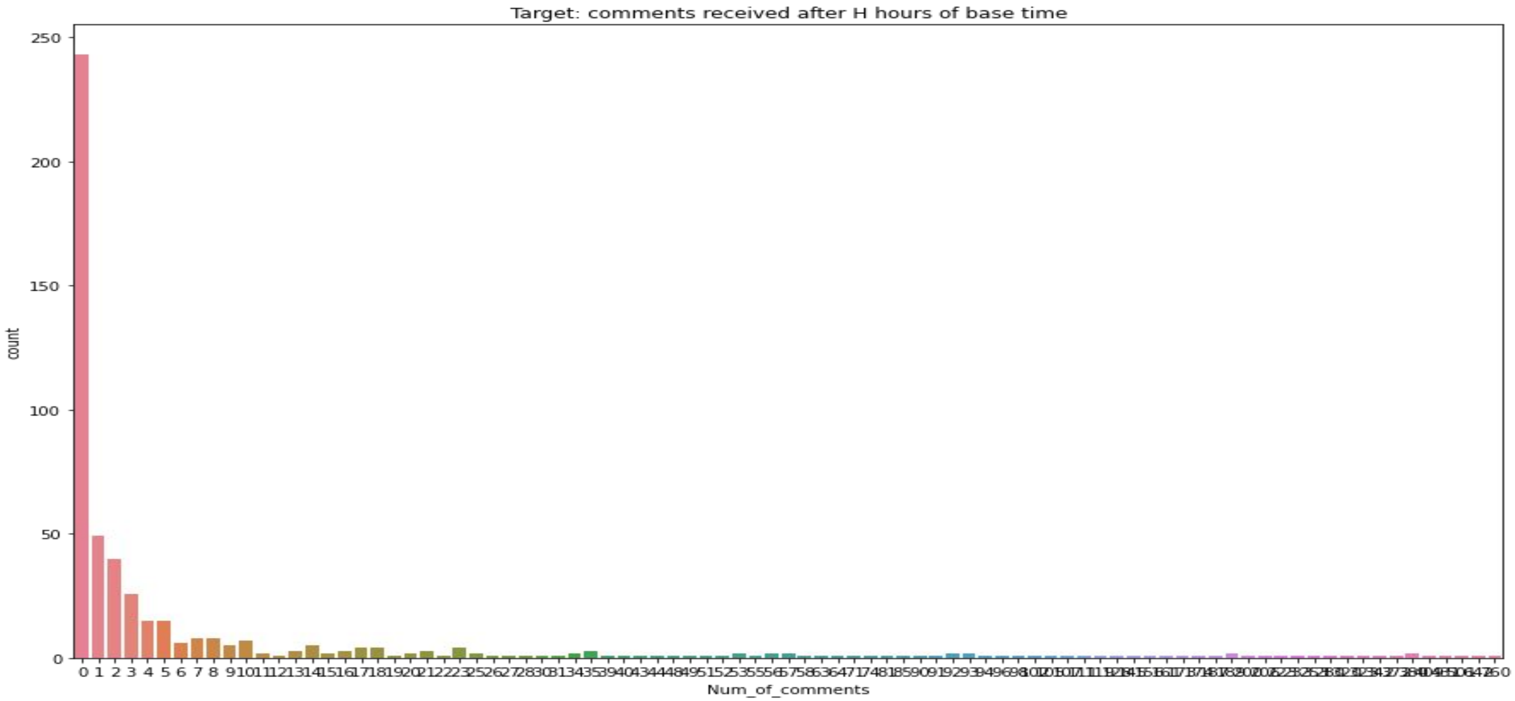
Feature Analysis (Predictors - Categorical)

Day when people actively commenting



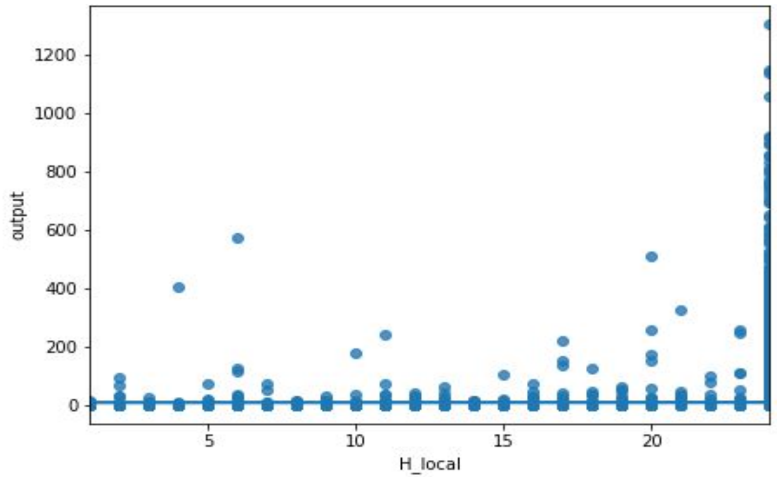
Feature Analysis (Predictors - Continuous)

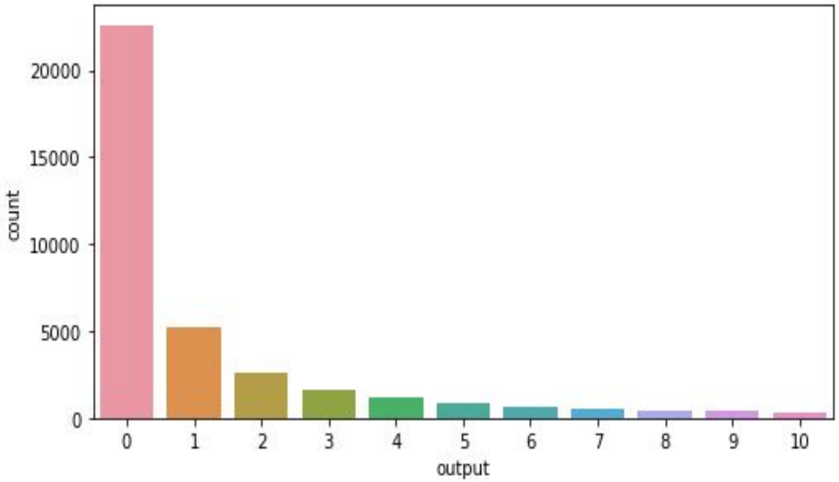
Comments Received after H hours



Feature Analysis (Predictors - Continuous)

H_local - Hours, for which we have the comments (target) received





less than 10 comments distribution for < 24hr

For 98% of post the time was taken for 24 hrs

1. For hours less than 24, most of the posts got less than 10 comments

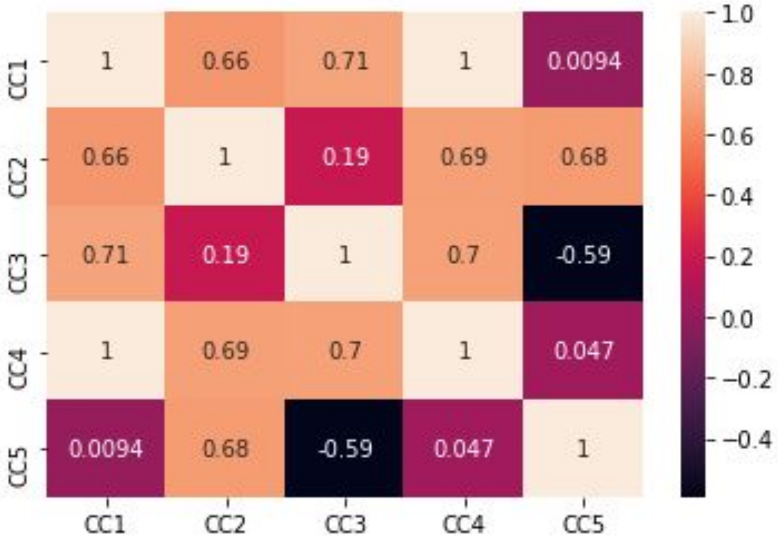
Feature Analysis (Predictors - Continuous)

CC1, CC2, CC3, CC4, CC5 and Derived Features

1. **CC5 was defined as CC2-CC3. So, among CC2, CC3 and CC5 we can remove one column to reduce the redundancy**

2. **CC1 & CC4 has strong correlation (1)**

3. **Derived features also have high correlation**



	CC1	CC2	CC3	CC4	CC5
CC1	1	0.66	0.71	1	0.0094
CC2	0.66	1	0.19	0.69	0.68
CC3	0.71	0.19	1	0.7	-0.59
CC4	1	0.69	0.7	1	0.047
CC5	0.0094	0.68	-0.59	0.047	1

4. **We have applied multi-collinearity removal methods to solve the issue**

Pre-processing (Final Steps)

- Removed cat variables Page type, Check-ins, Post promotion status
- Changed date/time features to categorical and created two columns as weekdays and weekends.
- Removed redundant column CC3 and its corresponding derived features after checking with Random Forest Feature Importance
- Scaled the data using standardscaler method
- Applied regularization techniques to tackle multicollinearity

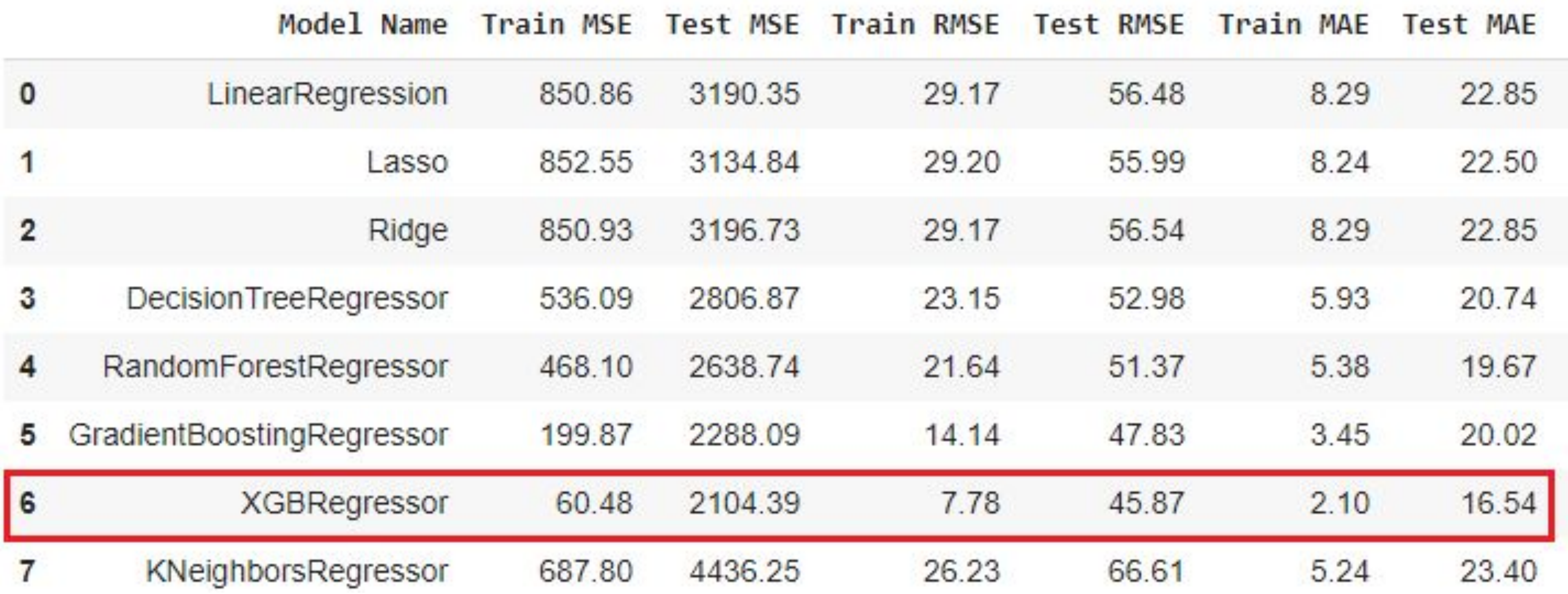
Machine Learning Models

We have Implemented Below Models

1. Multiple Linear Regression
2. Lasso Regression
3. Ridge Regression
4. Decision Tree Regression
5. Random Forest Regression
6. XGBoost Regression
7. Gradient Boosting Regression
8. KNN Regression

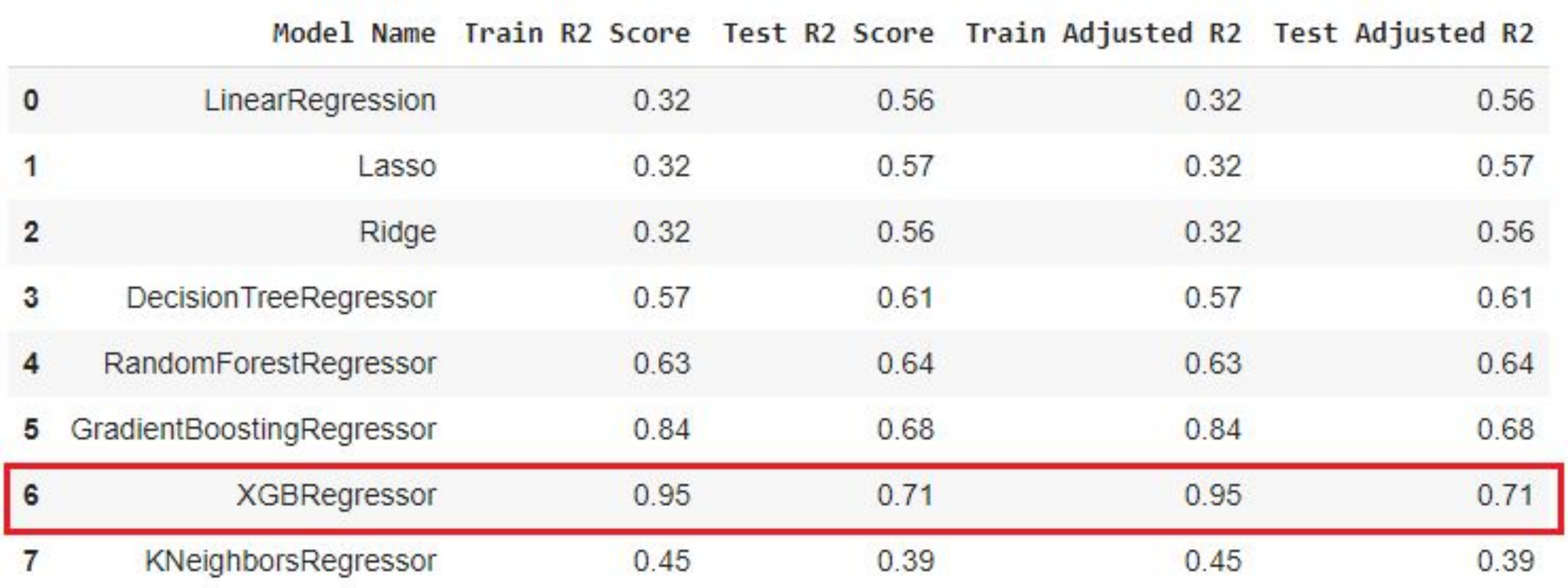
Compared Train and Test accuracy for all models

Model Evaluation (MSE, RMSE, MAE)



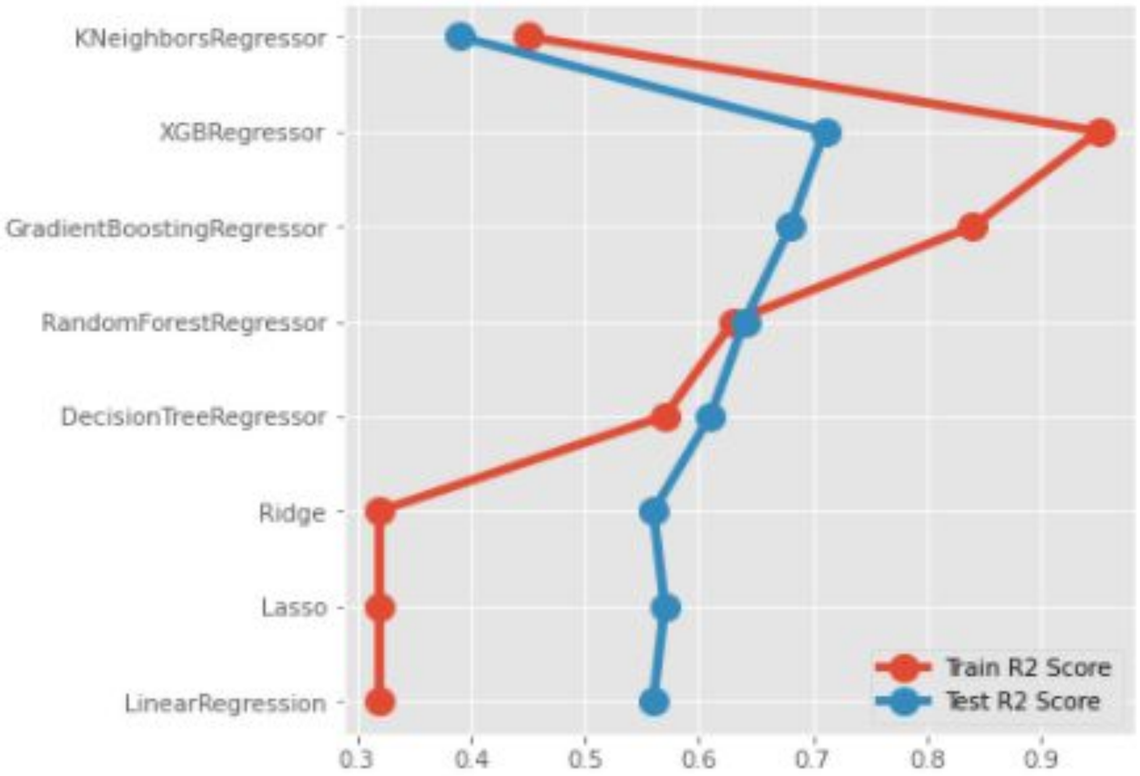
	Model Name	Train MSE	Test MSE	Train RMSE	Test RMSE	Train MAE	Test MAE
0	LinearRegression	850.86	3190.35	29.17	56.48	8.29	22.85
1	Lasso	852.55	3134.84	29.20	55.99	8.24	22.50
2	Ridge	850.93	3196.73	29.17	56.54	8.29	22.85
3	DecisionTreeRegressor	536.09	2806.87	23.15	52.98	5.93	20.74
4	RandomForestRegressor	468.10	2638.74	21.64	51.37	5.38	19.67
5	GradientBoostingRegressor	199.87	2288.09	14.14	47.83	3.45	20.02
6	XGBRegressor	60.48	2104.39	7.78	45.87	2.10	16.54
7	KNeighborsRegressor	687.80	4436.25	26.23	66.61	5.24	23.40

Model Evaluation (R2 and Adjusted R2)



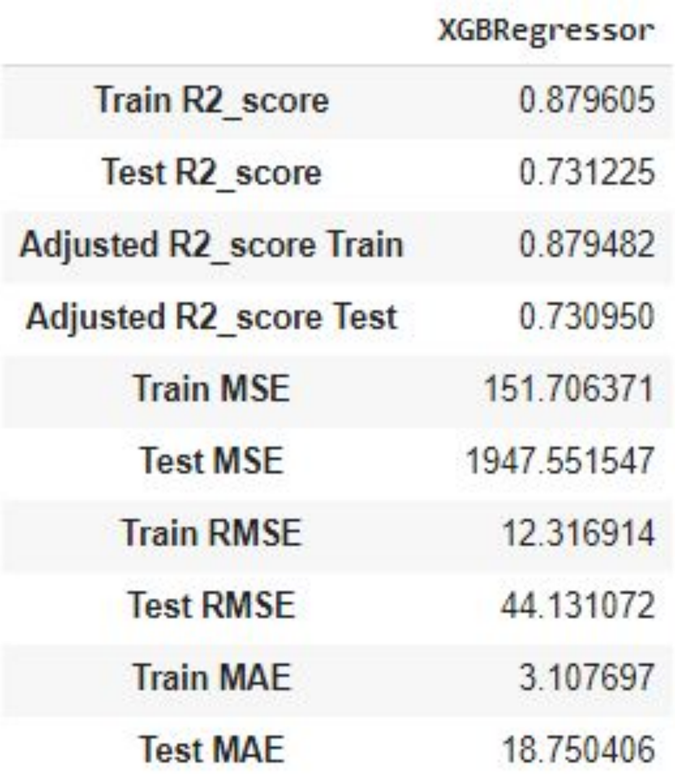
	Model Name	Train R2 Score	Test R2 Score	Train Adjusted R2	Test Adjusted R2
0	LinearRegression	0.32	0.56	0.32	0.56
1	Lasso	0.32	0.57	0.32	0.57
2	Ridge	0.32	0.56	0.32	0.56
3	DecisionTreeRegressor	0.57	0.61	0.57	0.61
4	RandomForestRegressor	0.63	0.64	0.63	0.64
5	GradientBoostingRegressor	0.84	0.68	0.84	0.68
6	XGBRegressor	0.95	0.71	0.95	0.71
7	KNeighborsRegressor	0.45	0.39	0.45	0.39

Model Evaluation (Comparison of R2 Score)



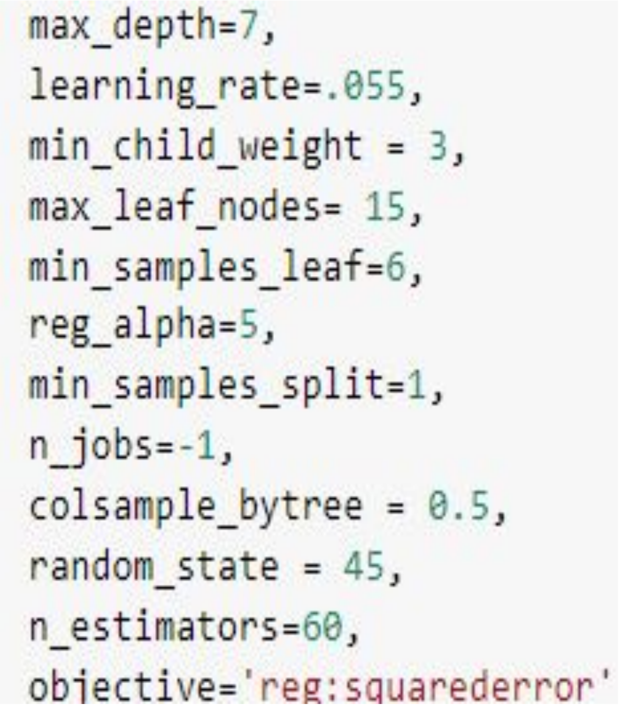
Model Evaluation (XGBoost)

Evaluation metrics



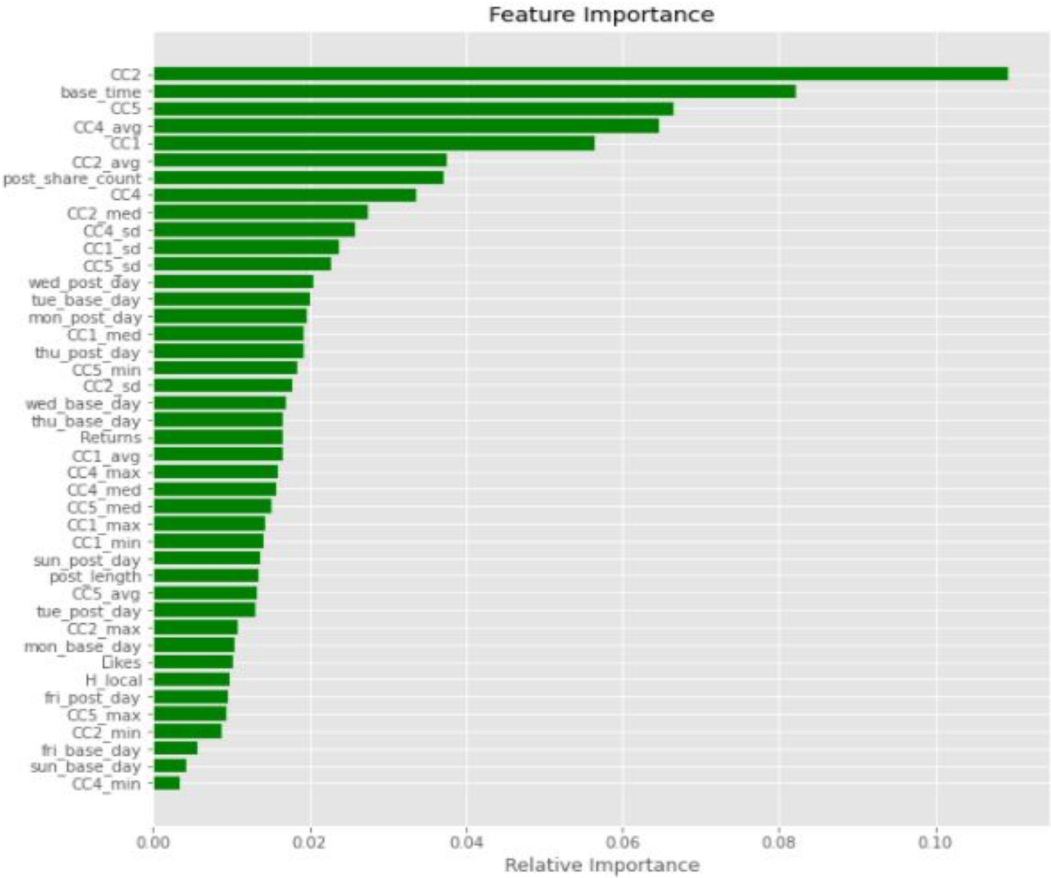
	XGBRegressor
Train R2_score	0.879605
Test R2_score	0.731225
Adjusted R2_score Train	0.879482
Adjusted R2_score Test	0.730950
Train MSE	151.706371
Test MSE	1947.551547
Train RMSE	12.316914
Test RMSE	44.131072
Train MAE	3.107697
Test MAE	18.750406

Best hyperparameter



```
max_depth=7,
learning_rate=.055,
min_child_weight = 3,
max_leaf_nodes= 15,
min_samples_leaf=6,
reg_alpha=5,
min_samples_split=1,
n_jobs=-1,
colsample_bytree = 0.5,
random_state = 45,
n_estimators=60,
objective='reg:squarederror'
```

Model Evaluation (Feature Importance)



XGBoost

Challenges

- 5 variants of datasets
- Large train dataset
- Very small test dataset
- High number of features
- Skewed target - 55% had 0 comments (unable to use log transform)
- Categorical variables with high number of labels (106 labels)
- Unknown label encoding so was not able to map labels with code
- High multicollinearity because of 25 derived variables

Conclusion

- **XGBoost performed best on our dataset with test adjusted R2 score of 0.73**
- **Gradient Boosting and Random Forest also performed well compared other models**
- **From feature importance plot of Random Forest, Gradient Boosting and XGBoosting we can conclude that most important features are**
    - **CC2 (Comment count in last 24 hrs w.r.t to selected basetime)**
    - **Base Time/Date**
- **Dataset was large, so unable to use grid-search to find the optimal hyperparameters. Model accuracy can be improved.**

Q & A