Understanding and Predicting User Engagement Metrics of Facebook News Posts Using Machine Learning.

Springboard Capstone Project 2

By: Harish Prabhala

Mentor: Ryan Rosario

Introduction

Facebook is not just a social networking company anymore; it is a media behemoth where hundreds of millions of users come to consume news everyday. In this context, it is pertinent to analyze and predict how the users are reacting to the news content so as to improve user engagement, which will ultimately help drive ad revenue. Our project sets out to explore the Facebook news posts of several different news organizations and analyze the user engagement metrics such as "likes" and "shares". The number of *likes* and *shares* for each of the news post is directly proportional to the engagement behavior of the users. The higher the number of *likes* and *shares,* the more is the user engagement for that particular post. Given these engagement metrics, a machine learning model can be trained to identify and predict the consumption pattern for the news posts. Such a model can be used by Facebook or any other news agency to learn about the engaging topics, words, etc. and predict the engagement patterns of their news posts which can help drive their user consumption and improve their ad revenue.

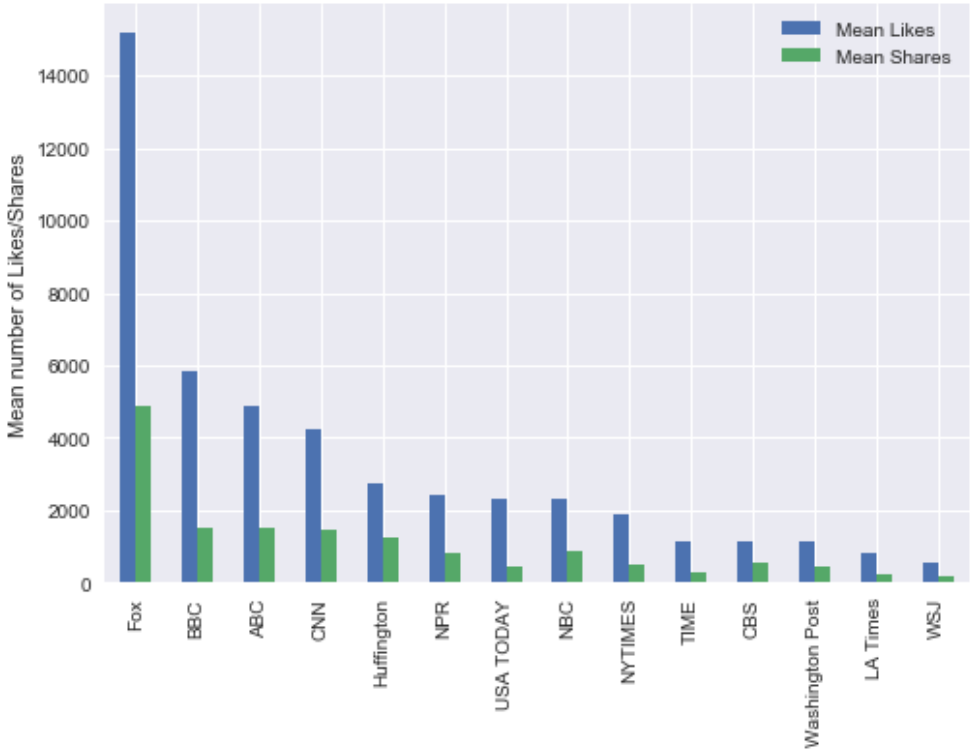
Exploratory Data Analysis:

The dataset used for this project is a collection of Facebook posts from 15 of the top mainstream media sources from 2012-2016. This dataset consists of around 500k Facebook news posts. Each new post has the following features:

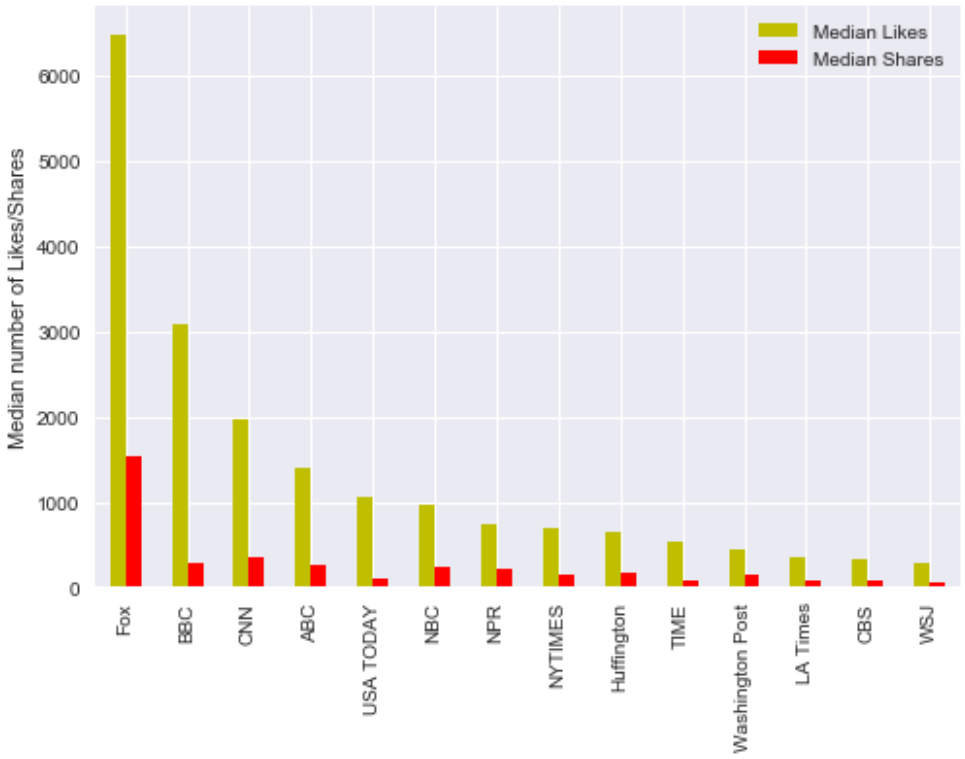
- ID
- Page ID
- Name – The title of the post
- Text
- Description
- Caption
- Post Type (type of the post, ex: photo, video, status update)
- Status type
- Like count
- Comment count
- Share count
- Link – This is the URL to the article
- Post time (GMT)

The name, text and description columns consist of all the text related to the post. Of the engagement metrics columns, for the purpose of this project we will only look at the likes count, shares count and post time columns.

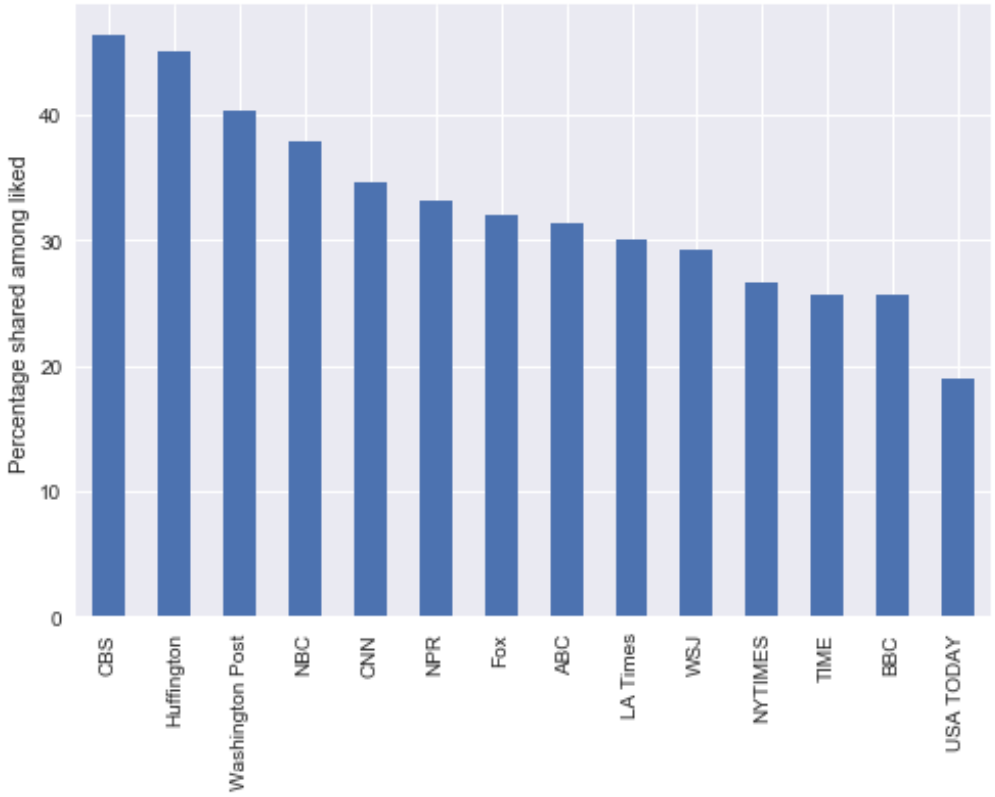
We conducted some exploratory data analysis to gain an insight into how the users responded to the news posts from different news agencies. First, we looked at the mean likes and shares for each news agency. Fox News had the highest mean number of likes and shares and Wall Street Journal had the lowest.



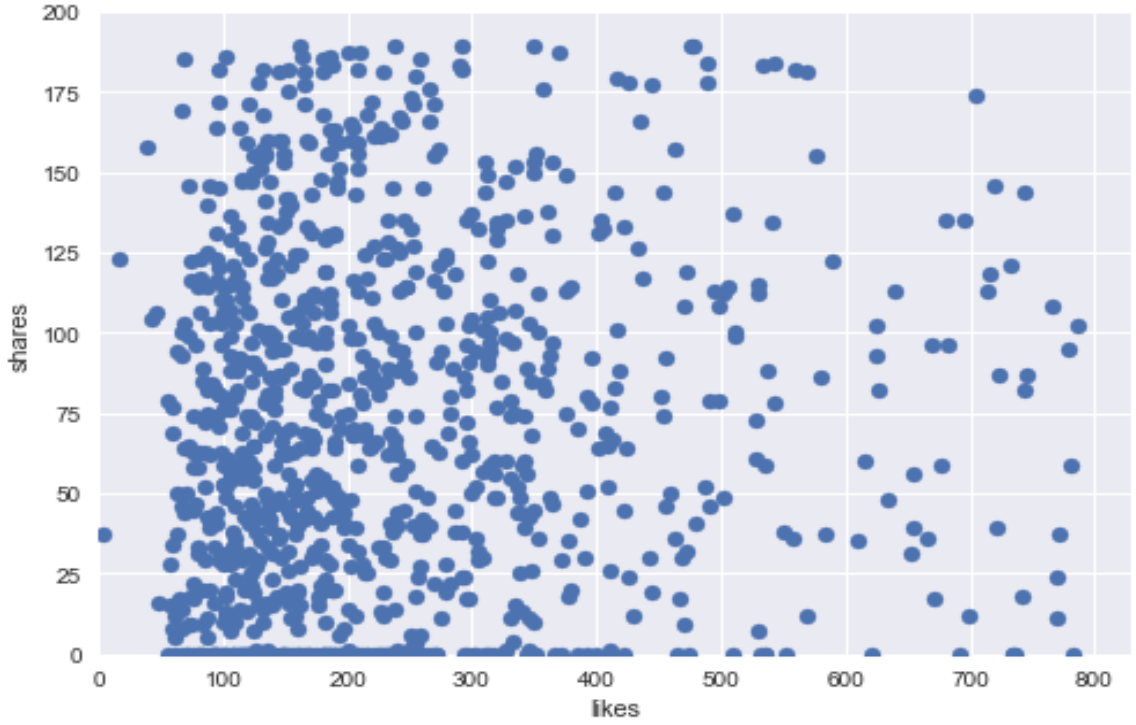
Next, we looked at the median likes and shares for each news agency and the saw the same pattern, as the mean likes and shares.

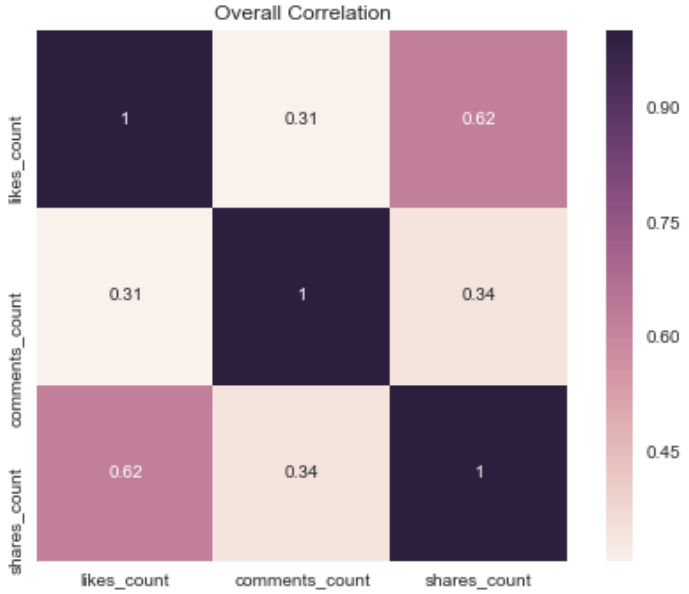


If we look just at the mean likes and shares, we want to look at what percent of the liked posts get shared. Below we can see that CBS and Huffington Post had the highest shares among the liked posts.



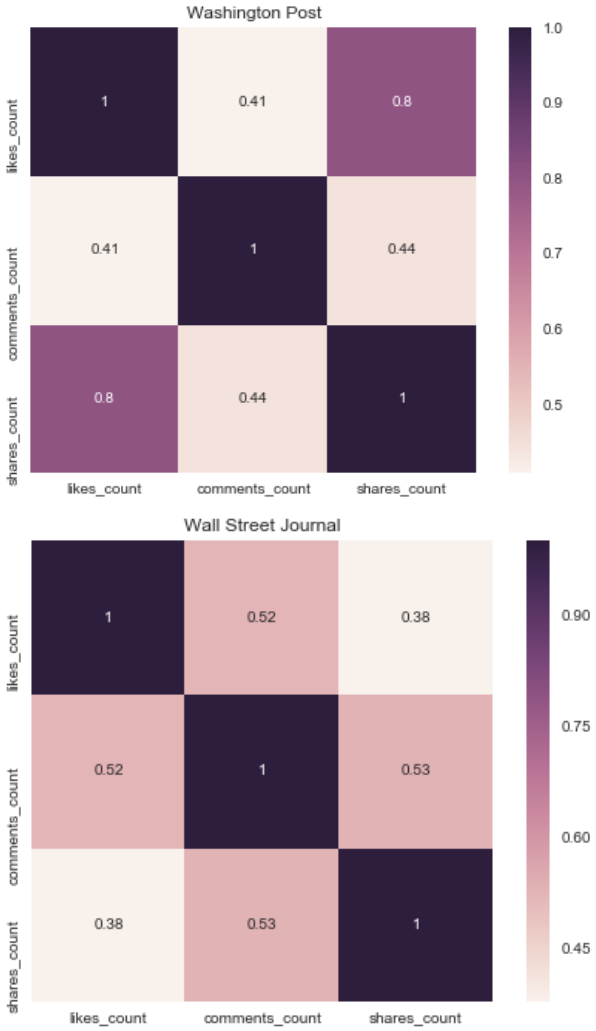
We also wanted to see if there is any correlation among the number of likes, shares and comments. We first looked at the correlation of all the posts and then looked at the correlation for each individual news agency. Below are the scatter plots and heat map for the correlation.



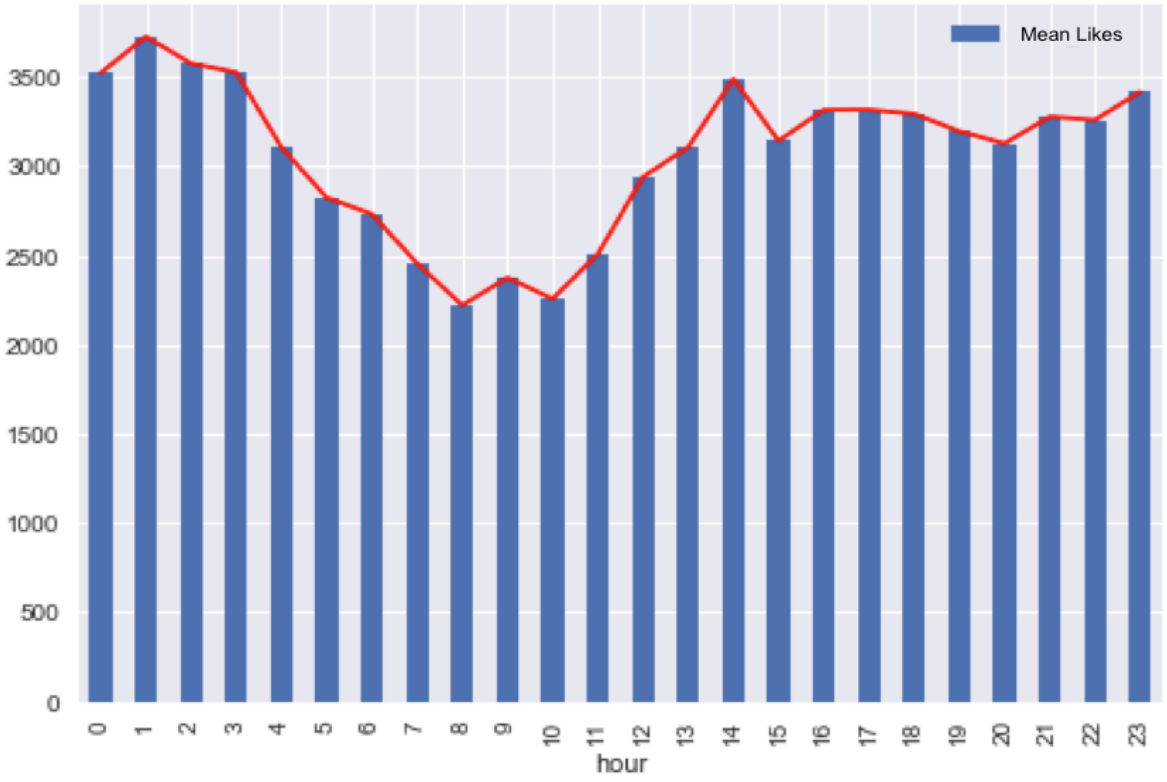


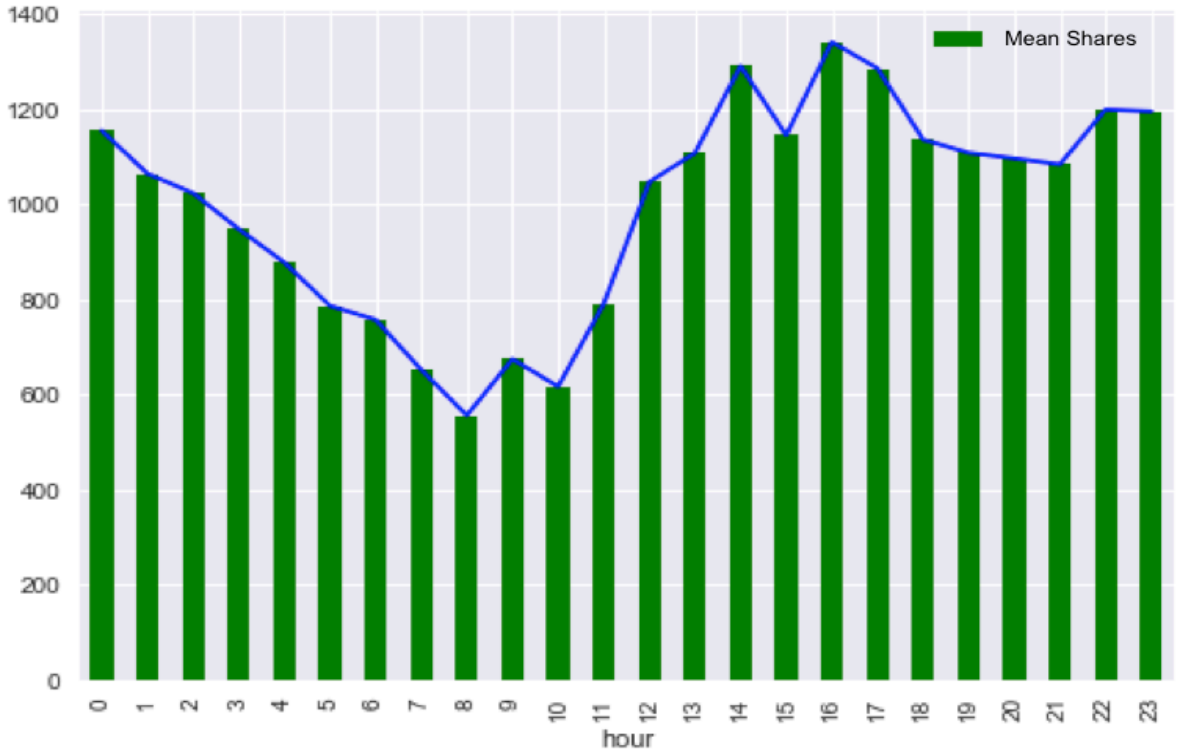
Overall there appears to be a significant correlation between the likes and shares and moderate correlation between likes and comments.

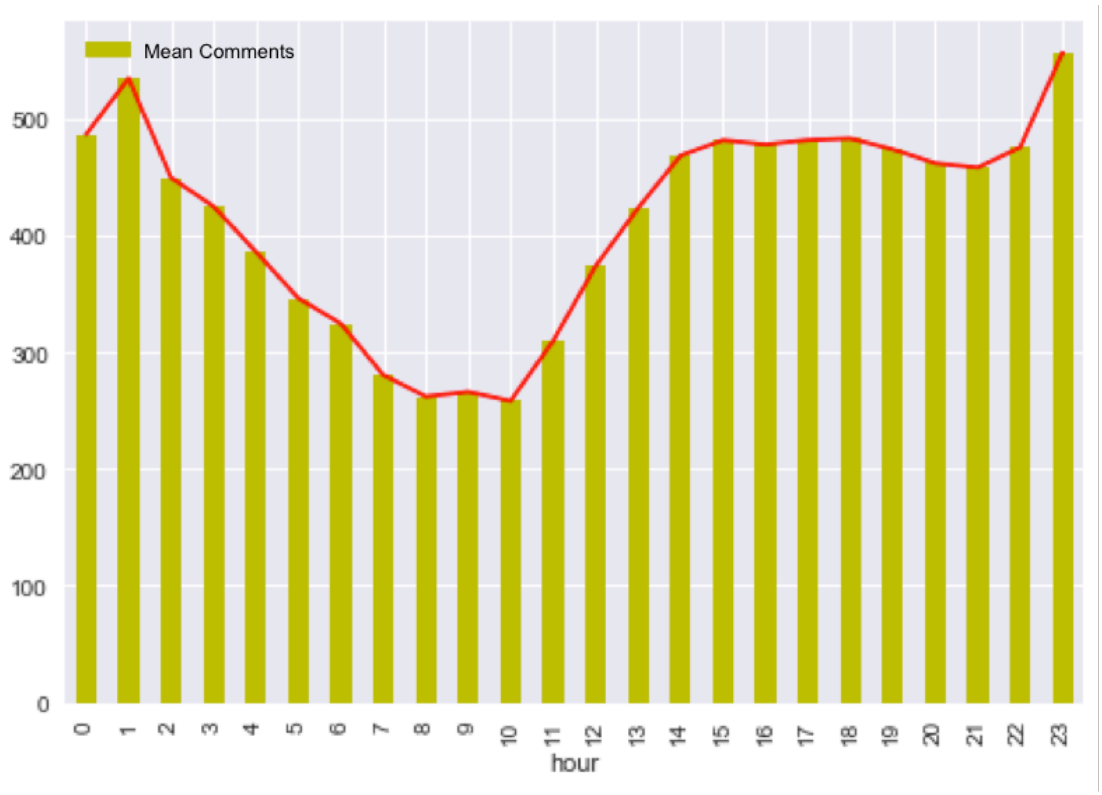
When we explored the correlation patterns for each news agency individually, we observed that The Washington Post had the highest likes and shares correlation while the Wall Street Journal had the least.



Our next step was to analyze the user engagement behavior at different time points. To achieve this, we broke down the timestamp (UTC time zone) of the post into hours, days and months and looked at the engagement patterns pertaining to each time stamp.

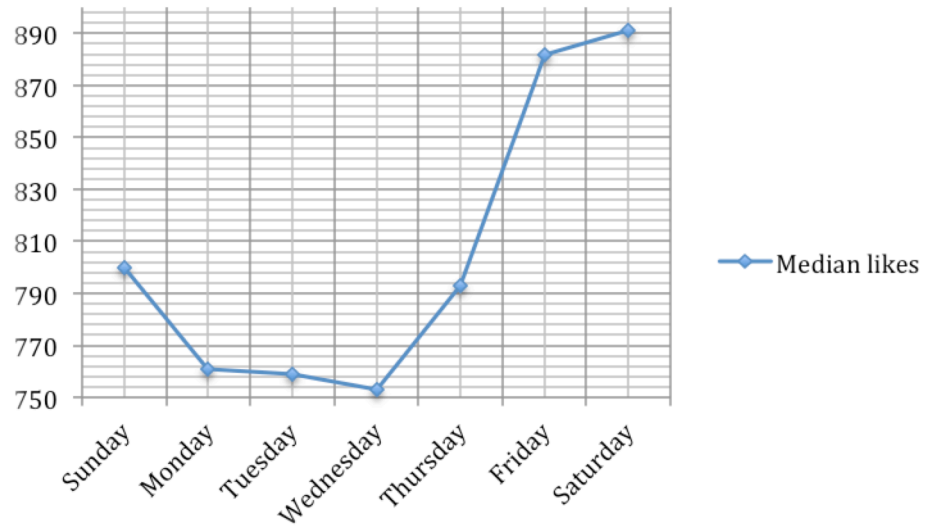

Mean Likes
3500
3000
2500
2000
1500
1000
500
0
0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23
hour

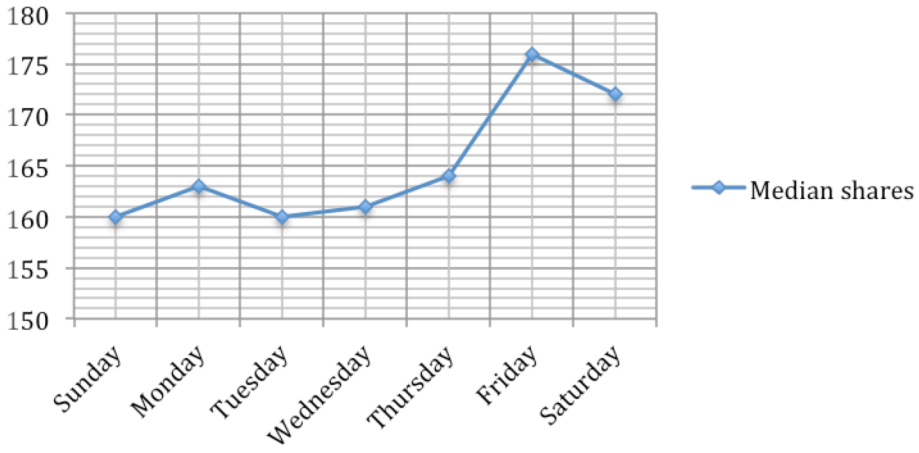

Mean Shares
1400
1200
1000
800
600
400
200
0
0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23
hour



The above graphs show the mean *likes*, *shares* and *comments* at different times of the day. The pattern shows that the users like and comment on news posts the highest between 11pm and 1am. Between 7am and 11am, the consumption is the lowest. Users share news posts the highest in the afternoon.

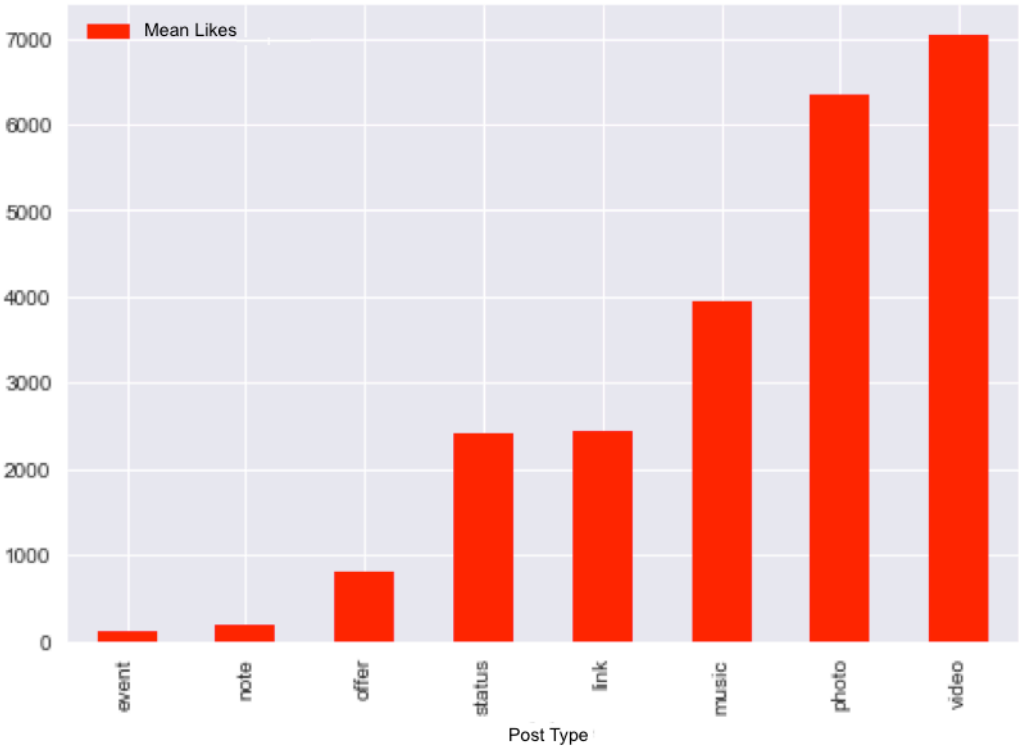
Below are the median *likes* and *shares* on a weekly basis. We found out that from Monday through Wednesday the user engagement is low and it increases mid week going into the weekend.





We then look at the user engagement metrics for the post type.

From the above graph we can learn that videos and photos get the highest number likes as compared to a status update or just a link to a post. We can infer that visual content has the highest user engagement than just text.



Data Preprocessing:

In this project, our input data for training the models will be text of the news post, which is a combination of name, message and description. Our target would be the *likes*. For the input text we will use text-preprocessing techniques used in natural language processing in order to clean our input features. We use the NLTK dependency/library in order to perform the preprocessing. The following steps are used to clean and preprocess our data;

- Upper Case to Lower Case: convert all upper case letters to lower case letters

- Stopword removal: Stopwords are the extremely common words in the English vocabulary, which do not hold significant predictive power or value of the contents or semantics of a document. Examples include *'a', 'an','the', 'all','any'* etc. Removal of stop words helps us in reducing the size of the featureset.

- Lemmatization: is the process of grouping together the derived forms of a word so they can be analyzed as a single item. For example, *'running','ran','runner'* are the derived words of the root word 'run'. Lemmatizing helps us is reducing the redundancies of multiple words of the same meaning.

- Punctuation Removal: We remove the punctuation to reduce the noise.

- Numeric values removal: Numerical values in the text are not essential in classifying the sentiment. Hence we remove all the numeric values in order to reduce the size of our features.

Target variable preprocessing - Initially, our goal was to do a regression to predict the number of likes for a particular post. But, we had quickly realized that it was not a feasible option since we were looking at the data over a period of time and the user volume is a constantly changing number. So, we wanted to convert the like counts into categories and predict the category in which each news post falls. As we did not have the user following information for the Facebook pages of the news agencies, we could not take the target variable as it is. Hence, we had to normalize our data. The following steps were taken to normalize the data in order to account for the user growth.

- First, the posts were separated into their respective news agencies.
- Next, the posts were first divided into five yearly groups, from 2012 to 2016.
- For each year, the *likes* count for each post were converted into their respective percentiles for that year.

Converting the count into percentile for each respective year enables us to normalize the data across the board to take into account the user growth year over year. Once the percentile conversion is done, we binned the percentiles into a combination of buckets for classification.

Feature Selection

There are several ways to do feature selection. In this project, we used the Chi-Squared method to select the top features. Working with a combination of words, we selected the top 28k words as an optimal feature set with the highest performance.

Training the machine learning models

Now that we have our features and the target, we can now split our entire dataset into test and train samples. The next step is to train the different machine learning algorithms on our training data and to predict the outcome class.

In this project, we used five different response variables. Below are the models.

1. Outcomes classified from 1 to 10 based on their percentile values (deciles).
2. Outcomes classified into four quartiles
3. Outcomes classified into 3 classes – low, medium, high
4. Outcomes classified into 2 classes – Highest quartile vs Lowest Quartile.
5. Outcomes classified into 2 classes – Highest quartile vs the rest.

We used Logistic Regression and Multinomial Naïve Bayes as our classifiers. Although both of them had similar accuracies, we chose to stick to logistic regression as it was slightly faster and marginally higher performance. I have tuned my hyperparameters in order to select the

optimal values for the highest performance.

Results:

Below are the accuracies of the different models. The output of each of the classifier gives us test and train accuracies, precision, recall and F1 scores. The F1 score is the harmonic mean of the precision and recall. The choice of the accuracy metric is subjective and depends on the nature of the data and desired output. In this project, we mainly looked at the accuracy scores and the area under curve (AUC) as our accuracy metrics. Higher the AUC, the better performing is the model. For AUC, ideally a perfect classifier should have a value of 1.

Target variable classes	Accuracy
10 classes (deciles)	Test - 17%
	Train - 36% (Benchmark 10%)
4 classes (quartiles)	Test - 37%
	Train - 39% (Benchmark 25%)
3 classes	Test - 49%
	Train - 52% (Benchmark 33%)
2 classes (Highest Quartile vs the rest)	Test - 69%
	Train - 72% (weighted)
2 classes (Highest Quartile vs Lowest Quartile)	Test - 73%
	Train - 76%

Classifier (Logistic Regression)	Area Under Curve (AUC)
Highest Quartile vs the rest	0.64
Highest Quartile vs Lowest Quartile	0.74

As we can see in the results, when we use 10 classes corresponding to the deciles, the accuracy of the classifier is glaringly low. This is because we are asking the classifier to accurately predict between posts whose *likes* lie very close to each other. It is challenging for even humans to make that distinction when the engagement metrics are that close. So, our model to try to classify and predict 10 classes is not the best way to pose the problem. Subsequently, we ran our classifiers on models with fewer classes. We observed the highest AUC (0.74) with the model that compares the posts

from the highest quartile to the lowest. Since the engagement metrics between the posts in these two extreme quartiles would be stark, our classifier could distinguish between these features with fairly good accuracy. We also looked at the highest quartile vs the rest and our classifier gave an AUC of 0.64.

Conclusion: In conclusion, among the various outcome models that we have modeled our machine learning algorithms on, we found that a logistic regression model with the two outcome classes of highest vs lowest quartile did the best. We believe that this model is a good start and can be improved furthermore to accurately determine the *likes* of news posts. Our recommendation to Facebook is to use our logistic regression machine learning model with highest vs lowest classes to help determine whether a news post falls into the bucket of high engagement or low. Facebook can use this model as a base to learn more about the features that are highly engaging and improve the model by further feature selection and applying other machine learning models.