How can we predict which Posts go viral on Facebook?

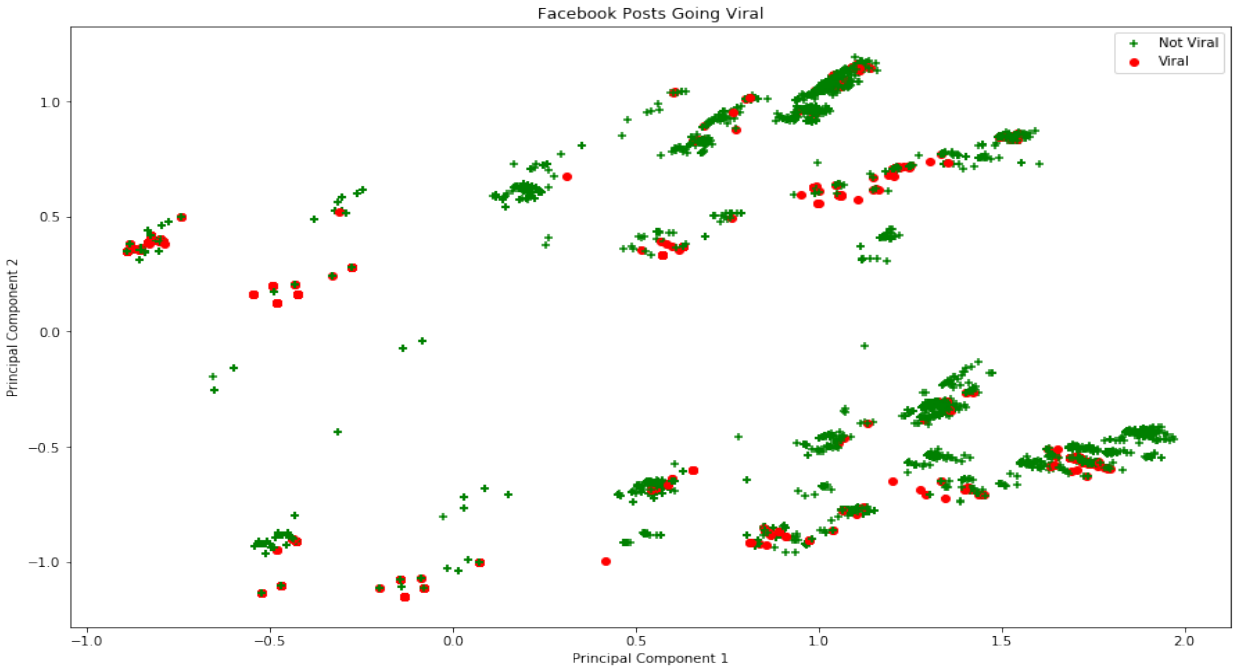
- Maragatham K N

Problem Statement

Companies post about their products on their Facebook pages. Facebook has a large audience, so predicting which posts would go viral can save them a lot of money which they spend on advertising their products. This would give them a chance to give more importance to the advertising/publicity of products which are not that popular with the audience.

Objective

To separate the viral and non-viral posts from the company pages on Facebook. Below is the PCA plot of the posts.



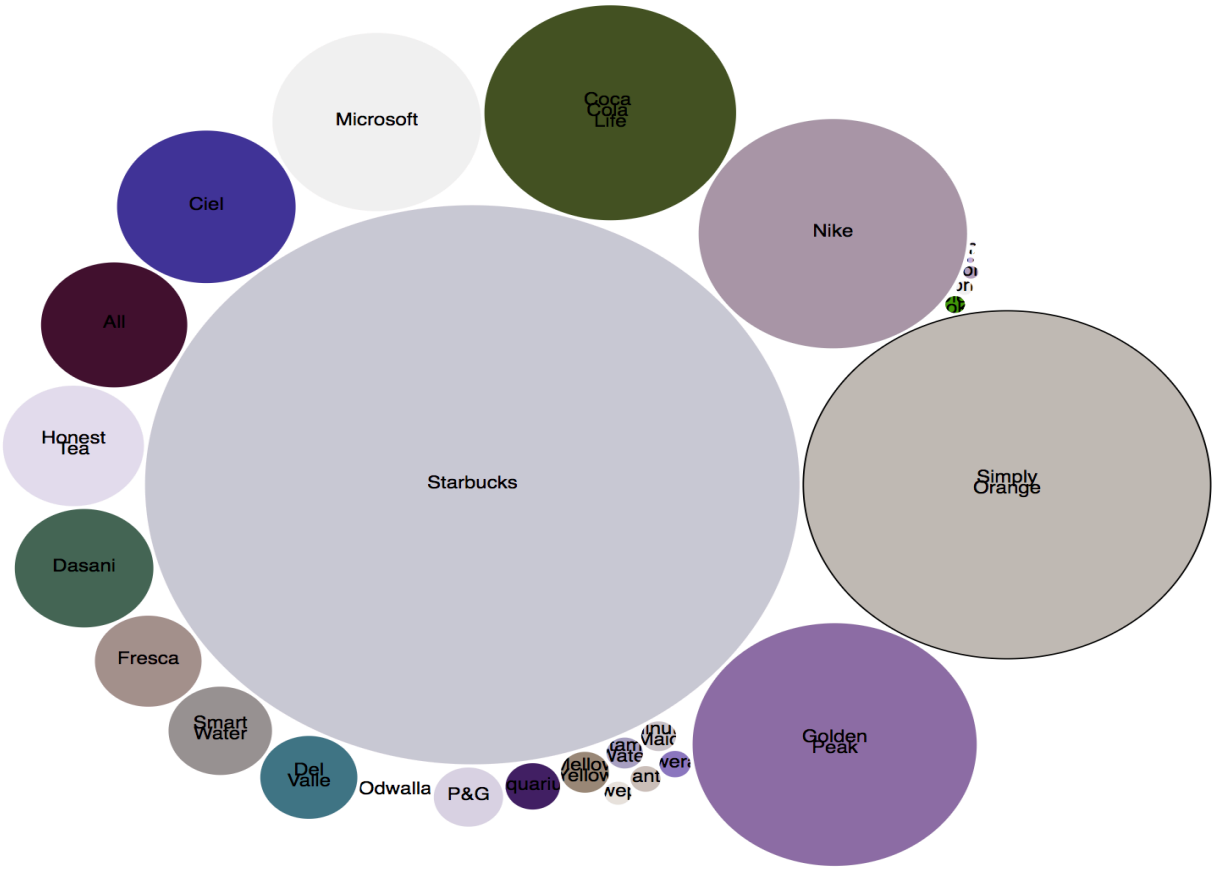
Data Retrieval

I collected data from the Facebook database using the Graph API and stored it on a Postgre SQL server in AWS instance and integrated with python using sqlalchemy to carry out my analysis.

For my analysis I selected 5 companies and their brands.

- Starbucks
- Coca Cola
- Microsoft
- Nike
- P&G

Below is the figure of the brands sized as per the no of posts in D3 I am considering for my analysis.

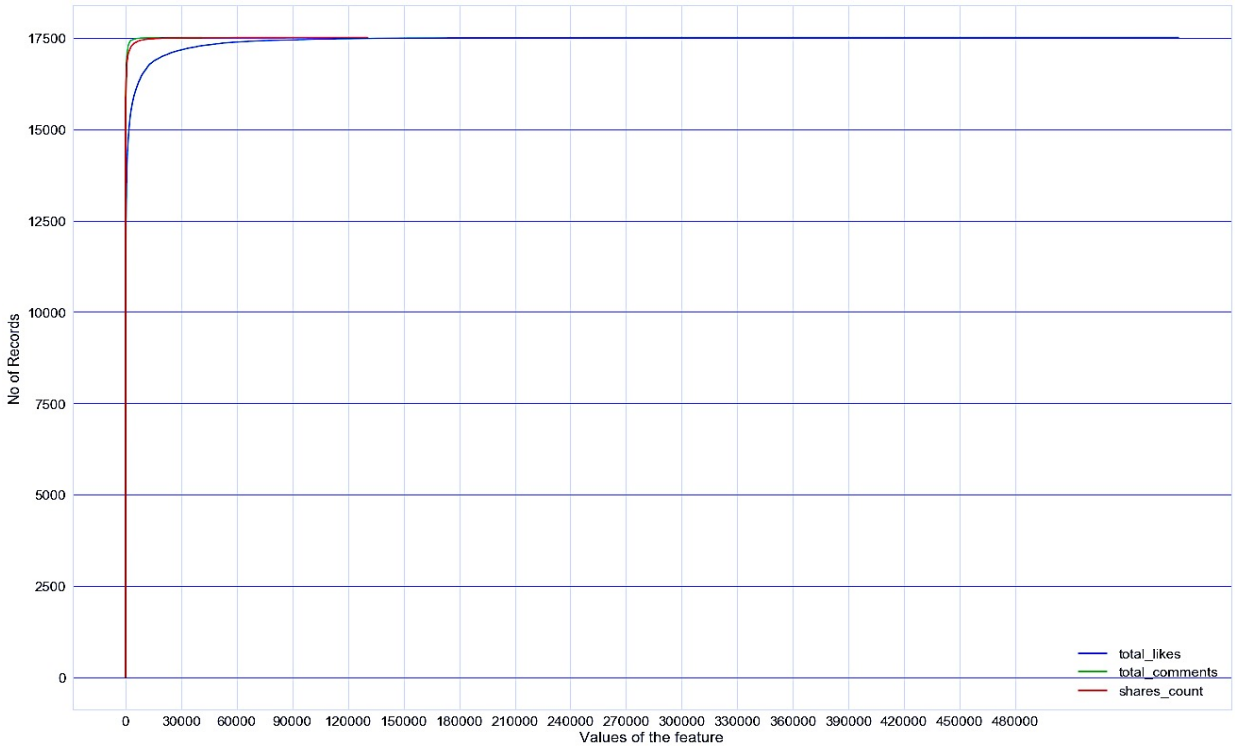


Setting a threshold to decide Viral/Non-viral posts.

To decide if a post is Viral or not I selected 4 parameters:

- Total Comments
- Total Likes
- Total Reactions
- Total Shares

Generally, the posts have less of comments, likes etc. But there is a steep increase in those parameters for the Viral posts. So, I plotted those parameters and selected the point where this steep increase is noticed.



I set my threshold to be 40,000 likes, 1000 comments, 1000 shares.

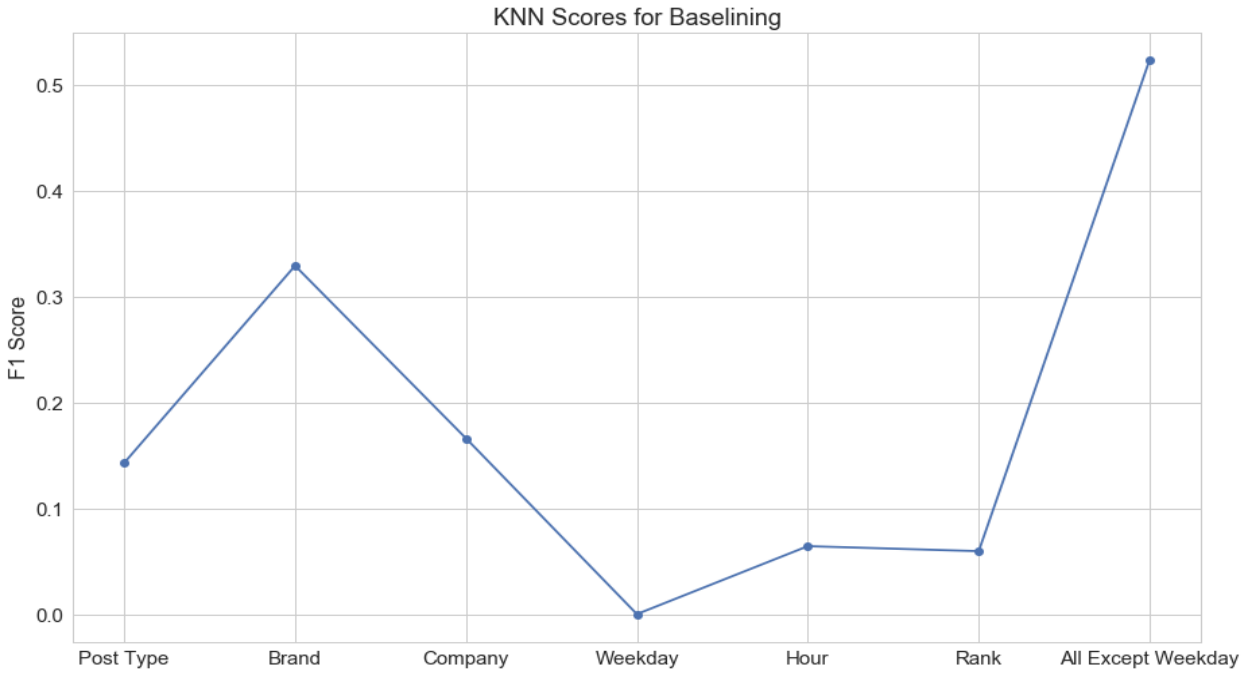
Feature Engineering

Introduced features like hour of the day, day of the week and words from the post messages. Translated the words from other languages to English using API, googletrans.

Approach

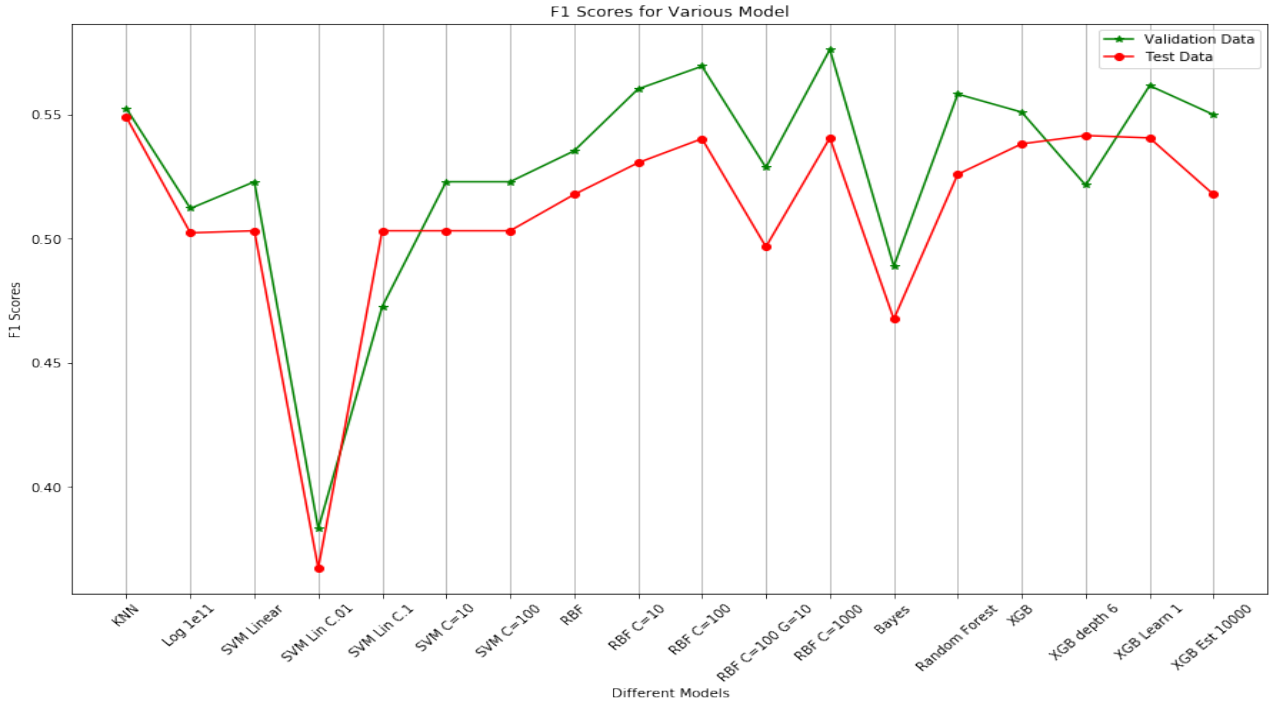
Baselining

Started baselining with KNN for my dummy features, brand, company, Weekday, Hour of the day, Rank etc.

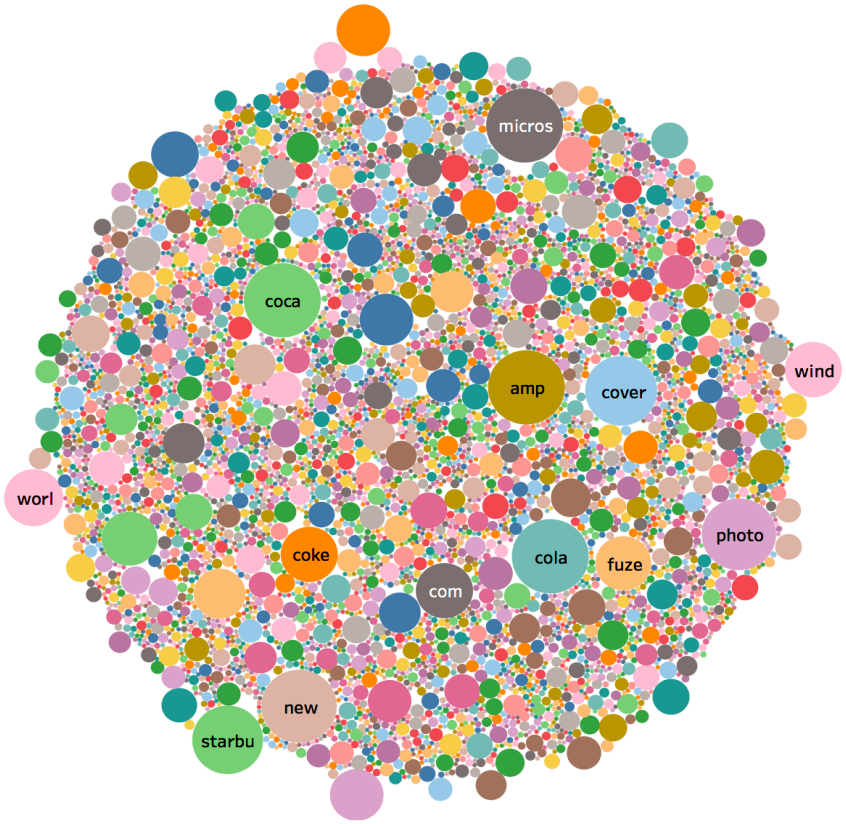


Selected my features to be brand, company, Hour, rank etc. and carried on with modelling.

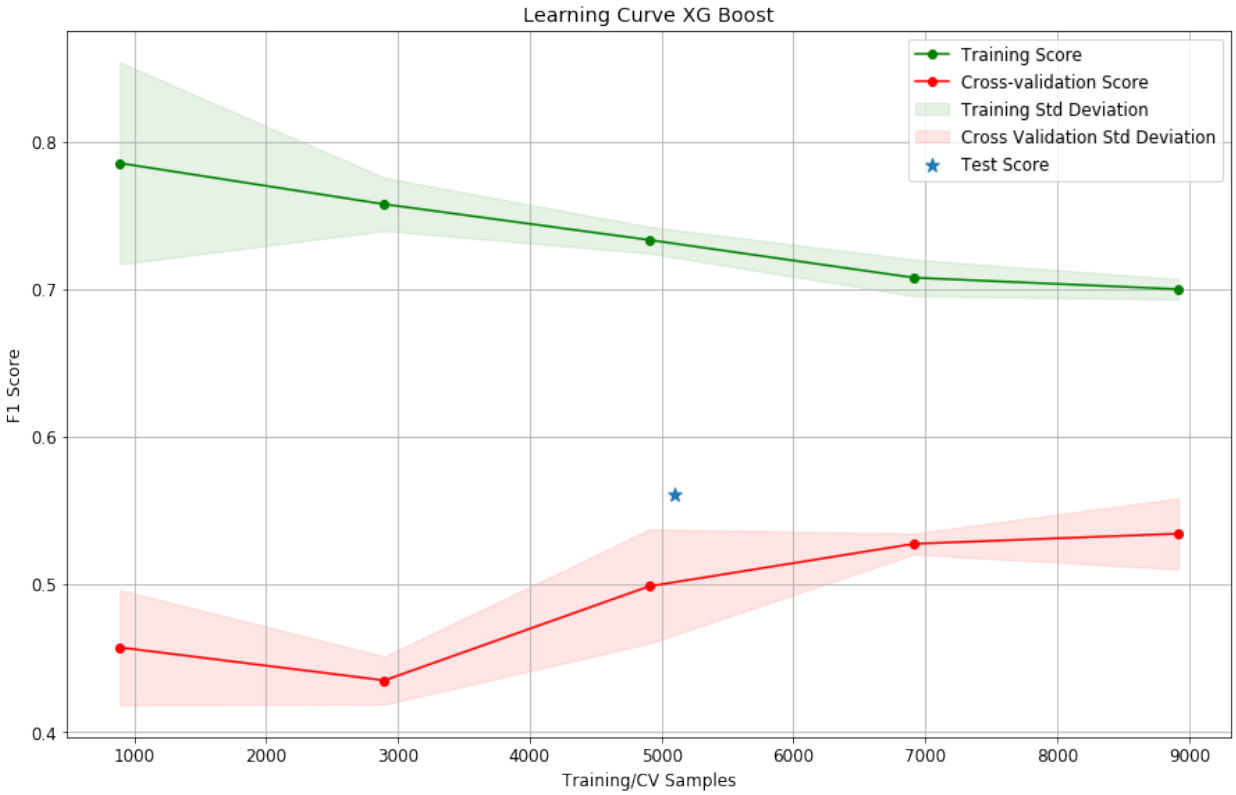
Tried all the below mentioned models with Over sampling and cross validation. Found the best one to be SVM model with RBF kernel and regularization as per the F1 score of .57



This indicated that my model needed some more features so I added the words from the post messages as my features. There were post messages in different languages so had to translate them and then take it into account.



I tried Random Forest and XGBoost model for these set of features. XGBoost worked better with a F1 score of .56.



The features did not help much in predicting the Viral posts but it did predict the non-Viral ones correctly and that is what my model needed as it had to provide the companies information about the posts on products which are not popular. So, based on this I put more weight on the precision score as I would not want to leave out many Non-Viral posts by predicting it as Viral ones. I used a F-beta score with beta being .2 for my results.

Cost Benefit Analysis

Did a cost benefit analysis based on the confusion matrix generated from my test data set.

Confusion Matrix	True Positive	False Positive
Actual Positive	4596	116
Actual Negative	192	193

If a company spends on per click basis for every post/product, they would incur a cost of 190/1000 clicks/per product i.e. $ 968,430

But if they strategize and only shows ads on per click basis for the non-viral posts/products, then they would incur a cost of $ 910,952 and saving 6% on their expenses.

Posts	Price for 1000 click $	Price per 1000 impression $	$ Spent on Ads
5097	190	0	$968,430
4788 + 309	190	3.99	$910,952
Net Gain			**6%** ($57,500)

Tools

Graph API, PostGre SQL server, AWS, Pandas, Tableau and D3.

What else I could do

Use the comments also as features
Use the content of the post, like image, video etc. as features.

Caveats

The number of features deciding the Daily Volatility could be anything ranging from the PE ratio to the company earnings, that could be incorporated.
To predict we need to take the inflation into consideration to mitigate the skew or we could take really short time periods like a month or a week and take all the ticker data(the changes in stock prices per sec) to understand the trend.
Any announcement related to the company could affect the change.

Conclusion

The features did not help much in predicting the Viral posts but it did predict the non-Viral ones correctly and that is what my model needed as it had to provide the companies information about the posts on products which are not popular.