SEIS -763 MACHINE LEARNING PROJECT REPORT

Facebook Comment Volume Dataset

Supriya Nithin
SEIS -763 Machine Learning
University of St.Thomas
MN, USA
nith9189@stthomas.edu

Mohammed Hesham Ahmed
SEIS -763 Machine Learning
University of St.Thomas
MN, USA
mhahmed@stthomas.edu

Bhavya Veluvolu
SEIS -763 Machine Learning
University of St.Thomas
MN, USA
bveluvolu@stthomas.edu

Sahithi Sai Katuri
SEIS -763 Machine Learning
University of St.Thomas
MN, USA
sskaturi@stthomas.edu

Abstract:

It’s evident from the past decade that the advent of Social Networking services has brought in a paradigm shift on society and change in the state of data related services. This gives an opportunity to conduct study on this phenomenon that transformed user interaction towards data services.

The first step is to understand user activity pattern and to examine the productiveness of machine learning modeling perspective on leading social networking service called “Facebook”. Based on user comment patterns, we are classifying if a post received a comment or not.

The Author of this dataset is Kamaljot Singh and This dataset is collected by web scraping the Facebook website.

The analysis is done by modeling the comment patterns using variety of classification modeling techniques,

specifically, SVM, Naïve Bayes, Decision Trees, Logistic Regression. In addition to these classification models, we have also made use of boosting technique in ensemble model called Adaboost.

We also learned and implemented XG boosting technique to scale up and push the limits of computing power for boosting algorithms as it was built and developed for the sole purpose of model performance and computational speed

Additionally, we had created a pipeline to transform the data depiction through series of steps which helped to automate the workflow and clean visualization.

The reason we chose this dataset:
Since this dataset has 54 features and 120k+ observations, it looked challenging enough for us to perform the algorithms that we learned in class. It has a lot of scope for classification as well as prediction, for example if a post receives a comment within H hours.
From marketing standpoint, most business sales practices are through creating social networking pages for their products, and our analysis can help understand the interest of public from their responses, when a post receives a comment or not

Feature Information:

1. Page popularity/Likes(Decimal Encoding – Page Feature):
Defines the popularity or support for the source of the document

2. Page CheckIn (Decimal Encoding – Page Feature):
Describes how many individuals so far visited this place. This feature is only associated with the places eg: some institution, place, theater etc.

3. Page talking about (Decimal Encoding – Page Feature):
Defines the daily interest of individuals towards source of the document/ Post. The people who actually come back to the page, after liking the page. This include activities such as comments, likes to a post, shares, etc by visitors to the page.

4. Page Category (Value Encoding – Page Feature):
Defines the category of the source of the document eg: place, institution, brand etc.

5 - 29. Derived (Decimal Encoding – Derived Feature)
These features are aggregated by page, by calculating min, max, average, median and standard deviation of essential features.

30. CC1 (Decimal Encoding – Essential Feature)
The total number of comments before selected base date/time

31. CC2 (Decimal Encoding – Essential Feature)
The number of comments in last 24 hours, relative to base date/time.

32. CC3 (Decimal Encoding – Essential Feature)
The number of comments in last 48 to last 24 hours relative to base date/time
33. CC4 (Decimal Encoding – Essential Feature)

The number of comments in the first 24 hours after the publication of post but before base date/time.

34.CC5 (Decimal Encoding – Essential Feature)
The difference between CC2 and CC3.

35. Base Time (Decimal 0-71 Encoding – Other Feature)
Selected time in order to simulate the scenario.

36. Post Length (Decimal Encoding – Other Feature)
Character count in the post.

37. Post Share Count (Decimal Encoding – Other Feature)
This feature counts the no of shares of the post, that how many peoples had shared this post on to their timeline.

38. Post Promotion Status (Binary Encoding – Other Feature)
To reach more people with posts in News Feed, individual promote their post and this features tells that whether the post is promoted(1) or not(0).

39. H Local (Decimal Encoding – Other Feature)

This describes the H hrs, for which we have the target variable/ comments receive**d.**

40 – 46 Post published weekday (Binary Encoding – Weekdays Feature)

This represents the day(Sunday...Saturday) on which the post was published.

47 – 53 Base DateTime (Binary Encoding – Weekdays Feature)

This represents the day(Sunday...Saturday) on selected base Date/Time.

54. Target Variable (Decimal – Target)

The no of comments in next H hrs(H is given in Feature no 39).

Algorithms Used:

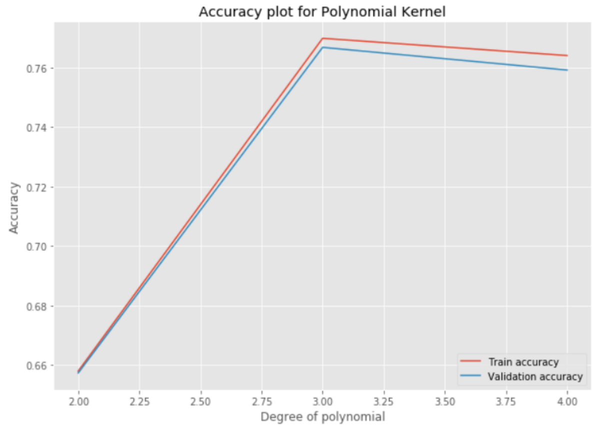
Several classification models have been used, to classify our target for this dataset. They are implemented using pipelines to streamline the process, as follows:

1. Linear SVM:

   Due to the large volume of training dataset, cross validation was not performed for Linear SVM. The model was trained for C = 1 (C being the regularization parameter). The accuracy obtained is 80.20% which is only slightly higher than the result for logistic regression. The confusion matrix is [[18833 1254] [5937 10306]].

2. Polynomial SVM:

   This model has been implemented over different degrees of polynomials. This was done in order to check for the degree which yields the best accuracy. 3-Fold Cross validation was also performed for polynomial degrees of [1,2,3]. As seen from the graph, model with degree 3 gives the best accuracy and generalizes it well. The accuracy obtained is 77.43% with a confusion matrix of [[19184 903] [7294 8949]].



3. Sigmoid SVM:

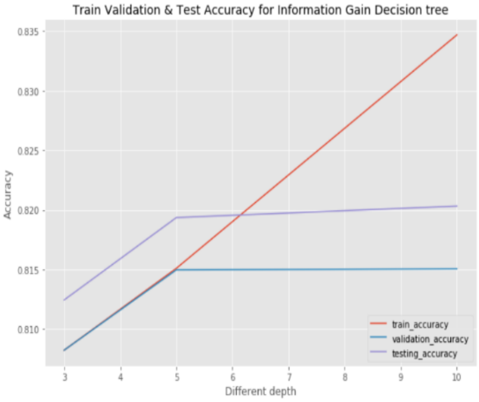
   Sigmoid kernel behaves like a logistic model and is of good help in classifying the output labels. This is again implemented for the values of C = [0.0001, 0.1, 1] and 3-fold cross validation is performed for this. It is observed that there is not much of difference in the accuracy of the train and validation set. This model performs low on the data set with accuracy of 71.59% and confusion matrix [[14901 5186] [5135 11108]].

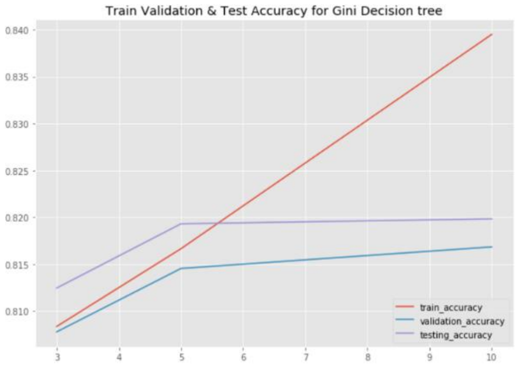
4. Naïve Bayes:

   Naïve Bayes works surprisingly because classification doesn’t require accurate probability estimates as long as maximum probability is assigned to correct class. Here again 3-Fold Cross Validation is performed, and we end up getting an accuracy of 73.17% with a confusion matrix of [[19245 842] [8902 7341]].

5. Decision Trees:

Decision tree is another algorithm which helps to overcome the amount of time taken by SVM for computing the values. Training data till the tree grows to its extreme may result in overfitting the dataset. In order to avoid this, pruning is performed on the data whilst fitting the model. Because of pruning, the model has less variance added and it can perform better when subjected to new data values.
The model is implemented taking both the entropy (information gain) approach as well as the GINI criterion for training. The tree depth values given are 3, 5 and 10. Figure below shows linear increase in the training accuracy as depth increases while test and validation accuracy increase minimally after tree depth 5. One can say the accuracy almost plateaus after that. But overall the best accuracy is obtained for the tree with max depth = 10 showing overall test accuracy of 82.03% and its confusion matrix is [[17353 2734] [3794 12449]]. The GINI approach also gives comparable results with the highest accuracy being obtained for tree depth = 10. Comparing both the approaches, which yield very similar results, it was observed that the computational time for the entropy approach was lesser than GINI which made it preferable to use. But that may change depending on the data.



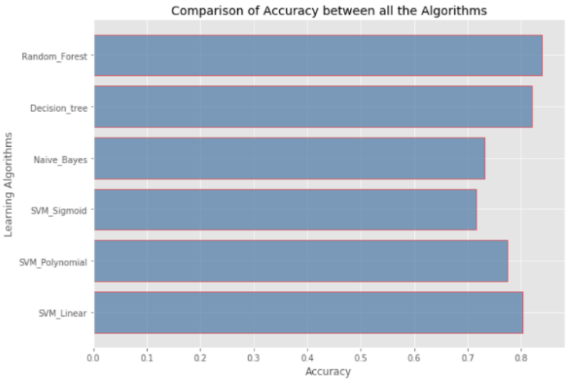


6. Random Forests:

The trees of the forest and more importantly their predictions here are uncorrelated (or at least have low correlations with each other). While the algorithm itself via feature randomness tries to engineer these low correlations, the features selected, and the hyper-parameters chosen serve this concept well. Based on the idea of ensemble learning, here a random forest classifier has been implemented using the default number of trees as 10 and the criterion as Entropy. This has resulted in an accuracy of 83.88% with a confusion matrix of [[17983 2104] [3750 12493]].

Overall Comparison:

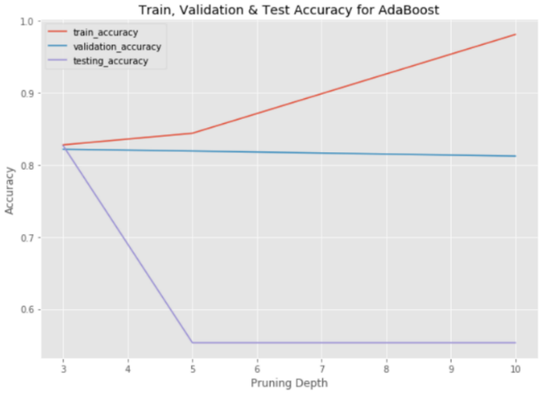
The figure below shows the accuracy of the different models that gave best results on the test data. Random Forests gives the best accuracy followed closely by Decision tree and then followed by Linear SVM.



Boosting Techniques:

1. Adaboost:

This algorithm uses base learner as Decision Tree classifier and if pruning isn't performed on the dataset, the model will overfit the data and may result in 100% accuracy. In order to avoid this and make the model more practical which can handle unknown values properly, pruning is performed. Pruning is done with three values 3, 5 and 10. The best results obtained are for the model with pruned value as 5. The below figure shows that the as the depth increases the testing accuracy degrades but the training accuracy increases. This is a case of overfitting the model.



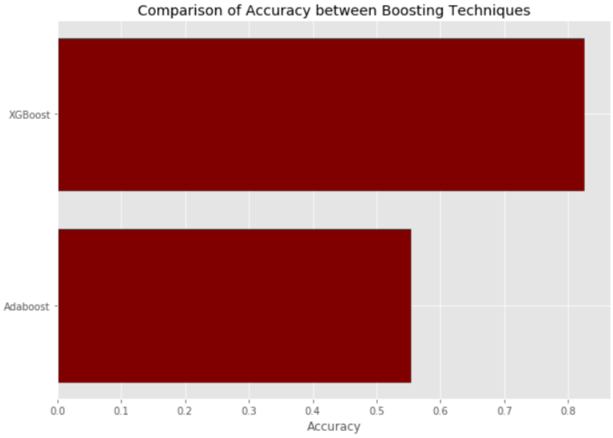
In addition to this, Adaboost was also used with SVM -sigmoid kernel since it yielded the weakest results amongst all of its other kernels. It was observed that using it was computationally very expensive as the time taken for it to run successfully was more than an hour. Apart from this, the accuracy obtained was 55% which again, isn't impressive.

2. XGBoost:

Whilst researching about the various boosting techniques, the highly recommended one was XGBoost. Extreme gradient boosting or XGBoost is has one of the fastest execution times of all the boosting techniques as well as other algorithms. It being extremely computationally efficient, uses/implements gradient boosting decision tree algorithm. This has resulted in the highest accuracy obtained at 82.50%, scraping past decision tree's accuracy of 82.03%, with a confusion matrix of [[17743 2344] [4013 12230]].

Overall Comparison between boosting techniques:

Clearly, XGBoost comes out on top here, both in terms of computational efficiency as well as the accuracy of 82% when compared with Adaboost's ~50% accuracy, making XGBoost the more preferred choice.



Drawbacks/Challenges faced:

The dataset contains 121097 instances and 54 rows.

- Being a vast dataset, it is very time consuming and will consume most of the CPU usage, which may lead to system hanging sometimes.
- Since the Data is created for addressing the prediction algorithms there are a lot of extra steps were needed to change the dataset suitable for the classification algorithm.
- We are not performing feature selection on the dataset as we have all the essential variables left after data cleaning.
- Since our project is based on the classification algorithms, we have PCA and Kernel PCA options for implementing dimensionality reduction.
- PCA: Though we could successfully implement PCA on the dataset, implementing the models on the data obtained after the dimensionality reduction is resulting in lower accuracy when compared to the implementation of model on the data obtained after cleaning.
- Kernel PCA: Kernel PCA would have been a better solution for the dataset than PCA, but due to vast data in the dataset we are encountering with the memory issue when trying to implement Kernel PCA on the data.
- Due to the presence of the different variables in the dataset, we even found it a difficult process on visualizing the data.
- We tried including all the algorithms supporting the classification model in the project, in order to understand which model is best for this scenario. We have implemented Logistic Regression, SVM with linear, RBF, sigmoid and polynomial kernels, Decision Trees with Entropy and Gini Classifier, Naïve Bayes. When we tried using K Nearest Neighbor algorithm, we were facing the memory issue because of which could not produce desired results.
- Implementing of ensemble methods like Ada Boost which would improve the accuracy of the model is very time taking and took a lot of effort understanding and implementing the algorithm.

Conclusion:

We can conclude that for Dataset the best accuracy is obtained by Decision tree. This project has made us think on data from a broader aspect regarding data handling, feature selection and thinking from different point of views. The dataset was very diverse in terms of the information provided and it was really challenging to work on both datasets. Additional things that might have resulted in better accuracy could be; performing Kernel Principal Component analysis on the data and reducing the dimensions. Picking only those components which covers most of the variance in the dataset.

This notebook has also been hosted on Google Cloud Platform under its version of Jupyter called Google Cloud Datalab.

References:

1. UCI ML Repo: https://archive.ics.uci.edu/ml/datasets/Facebook+Comment+Volume+Dataset#
2. Kaggle: https://www.kaggle.com/kiranraje/prediction-facebook-comment
3. Class lectures and Jupyter notebooks.
4. Google/YouTube for multiple concepts