VIT®

Vellore Institute of Technology

(Deemed to be University under section 3 of UGC Act, 1956)

CHENNAI

TITLE OF THE PROJECT:

Facebook Data Analysis

School of Computer Science and EngineeringVIT Chennai

Winter 2021-2022

Course: Big Data Framework

Course Faculty: Dr. Suganeshwari

Submitted By

Srikar Alladi-19MIA1103

ACKNOWLEDGEMENT:

Primarily, we would like to thank God for giving us the resources and strength for being able to complete this project with success.

Further, we would like to express my special thanks and gratitude to our Big Data Frameworks Faculty Dr. Amrit Pal, whose valuable guidance has helped us patch this project and make it full proof success. His suggestions and instructions have served as the major contributor towards the completion of this project. Throughout the course, he has always entertained and shared his knowledge on this topic with great enthusiasm. We would like to thank him for his constant support and encouragement towards making this project.

We would also take this opportunity to thank Dr. Ganesan R., Dean School of Computer Science Engineering (SCOPE), for extending the facilities of the school towards our project and for his unstinting support.

Finally, we would like to thank our parents and family who have encouraged us throughout this project to come up with new ideas and helped us get through problems with valuable solutions.

We have learnt a lot through this project and feel ourselves prepared to give solutions for upcoming challenges towards Facebook Data Analysis.

Table Of Contents:

FACEBOOK DATA ANALYSIS:

In this Project we are going to analyze the Facebook data using Spark for the purpose of better decision making for the business.

Abstract:

The concept of big data has been around for years; most organizations now understand that if they capture all the data that streams into their businesses, they can apply analytics and get significant value from it. But even in the 1950s, decades before anyone uttered the term “big data,” businesses were using basic analytics (essentially numbers in a spreadsheet that were manually examined) to uncover insights and trends.

The new benefits that big data analytics brings to the table, however, are speed and efficiency. Whereas a few years ago a business would have gathered information, run analytics and unearthed information that could be used for future decisions, today that business can identify insights for immediate decisions.

The ability to work faster – and stay agile – gives organizations a competitive edge they didn’t have before. Through our project we intend to carry out analysis on a preferably large dataset. So we have chosen the dataset obtained from several Facebook users. By carrying out certain operations, we intend to harness their data and use it to identify new opportunities.

Through our project we intend to carry out analysis on a preferably large dataset. So we have chosen the dataset obtained from several Facebook users. By carrying out certain operations, we intend to harness their data and use it to identify new opportunities.

The main Objective of the project is to analyze the Facebook data using pyspark for the purpose of better decision making for the business. Business owners utilize the data to understand customer needs and their behavior to make profit in their business. Facebook data analysis is the process of collecting, analyzing Facebook data and visualizing extracted results to the end users.

1. Introduction:

Smartphones without social media usage in the daily lifestyle of people is unthinkable. As per 2017 statistics, nearly 1.37 billion daily active users for Facebook. Every user contributes some type of data in structured or semi-structured or unstructured data format.

Business owners utilize this data to understand customer needs and their behavior to make profit in their business. Facebook data analysis is the process of collecting, analyzing Facebook data and visualizing extracted results to the end user.

It’s important to be able to analyze customers and their behavior on a micro level due to Facebook’s ever-changing algorithm, and the implications for our content and business. Facebook is said to have more than 500 million users in

2010. The field of social networks and their analysis has evolved from graph theory, statistics and sociology and it is used in several other fields like information science, business application, communication, economy etc.

Analyzing a social network is similar to the analysis of a graph because social networks form the topology of a graph. Graph analysis tools have been there for decades. But they are not designed for analyzing a social network graph which has complex properties. An online social network graph may be very large. It may contain millions of nodes and edges. Social networks are dynamic i.e. there is continuous evolution and expansion. A node in a social network usually has several attributes. There are small and large communities within the social graph. Old graph analysis tools are not designed to manage such large and complex social network graphs.

Facebook is a preferred social network by marketers, not only because of the sheer number of users represented but also because of its incredibly insightful analytics suite. It’s important to be able to analyze customers and their behavior on a micro level due to Facebook’s ever-changing algorithm, and the implications for our content and business. If we refuse to adapt our approach based on these insights, we’re doomed to obscurity on the news feed.

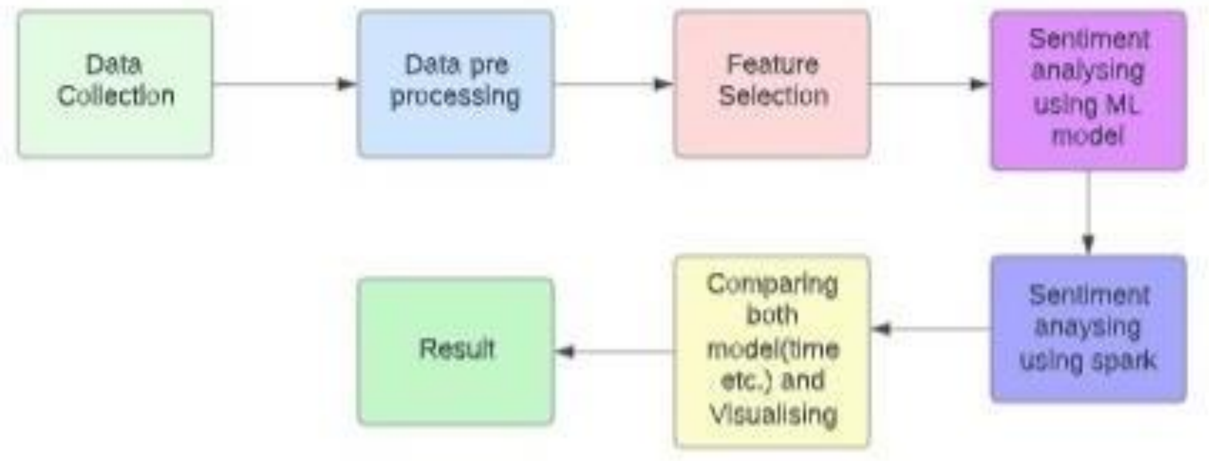
A deep Facebook data analysis shouldn’t be a one and done situation. Ideally, we’ll be auditing our efforts every few months or so at most. This will help us predict the likings, and the general summary of many users as a whole.

2. Apache Spark:

Apache Spark is a lightning-fast cluster computing system. It provides a set of high-level API namely Java, Scala, Python, and R for application development. Apache Spark is a tool for speedily executing Spark Applications. Spark utilizes Hadoop in two different ways – one is for Storage and second is for Process handling. Just because Spark has its own Cluster Management, so it utilizes Hadoop for Storage objectives.

Spark is intended to cover an extensive variety of remaining loads, for example, cluster applications, iterative calculations, intuitive questions, and streaming. Aside from supporting all these remaining tasks at hand in a particular framework, it decreases the administration weight of keeping up isolated apparatuses.

3. Proposed system:


Data Collection
Data pre processing
Feature Selection
Sentiment analysing using ML model
Sentiment anaysing using spark
Comparing both model(time etc.) and Visualising
Result

4. Data Collection

Facebook Dataset is taken from the GitHub

- GitHub gives insights from Facebook dataset which consists of identifying users that can be focused more to increase the business.
- These valuable insights should help Facebook to make intelligent decisions to identify its useful users and provide correct recommendations to them. We have taken this dataset from github.com
- Here we used a csv file. This dataset contains 9903 entries (big dataset) with 14 columns. We have columns namely age, userid, dob, etc., Column names are well defined so that everyone can interpret easily

https://github.com/jaegoan9/Facebook-Sentiment-Analysis

5. Data Pre-Processing

The collected data consists of different emotions, stop words, acronyms, etc. But during analysis this type of data needs to be converted into the proper format to extract sentiments from the user behavior.

- Tokenization
- Various Dictionaries
    1. Acronym Dictionary
    2. Stop Words Dictionary
- Emoticon

For Example:

Considering one of the Facebook posts regarding new mobile features. Users' opinion about the new phone might be positive or negative or neutral.

Example for Positive Sentiment

Looks are awesome. Battery backup is excellent. Camera is good. The display light quality is good.

Example for Neutral Sentiment

Although this is good mobile, looks good, but Problem is that it doesn’t provide separate Space for dual SIM & memory card together.

Example for Negative Sentiment

Not good one as expected. Camera quality very poor.

5.1 Tokenization

Comments extracted from Facebook are divided into tokens. This is known as tokenization process. For example, ‘Looks are awesome. Battery backup is excellent. Camera is good. The display light quality is good is divided down into ‘Looks’, ‘are’, ‘awesome’, ‘.’, ‘Battery’, ‘backup’, ‘is’, ‘excellent’, ‘.’, ‘Camera’, ‘is’, ‘good’, ‘.’, ‘The’, ‘display’, ‘light’, ‘quality’, ‘is’, ‘good’, ‘.’

5.2 Various Dictionaries:

Acronym Dictionary:

It is used to give the required acronym for the words, if needed.

Stop Words Dictionary:

It is used to remove the unrelated words in the sentiment analysis process.

Example: A, An, The, Has, Are, Is.

5.3 Emoticon:

This is used to detect the emoticons for the purpose of classifying the comment as positive or negative or neutral.

6. Deployment

6.1 Sentimental Analysis using Spark and ML

Firstly we will take the clean data and apply the Redex Tokenizer which is imported from the ml feature package then we immediately remove the stop words then after that we will apply the count Vectorizer model which helps in the conversion of the text documents to the vectors of token counts. This model produces sparse representations for the documents over the vocabulary, which can then be passed to all algorithms. After that, we are using the pipeline which allows us to maintain the data flow of all the relevant transformations that are required to reach the end result. We have taken the output of regex tokenizer, the output of stop words remover, and the output of count Vector Model, category, and label as the parameters. Then we fit and transform the pipeline. Then we perform the testing and training based on a 0.8, 0.2 ratio.

6.1.1 Naive Bayes

Naïve Bayes Classifier is one of the simplest and most effective Classification algorithms which help in building the fast machine learning models that can make quick predictions. It is a probabilistic classifier, which means it predicts on the basis of the probability of an object.

Iʼhe foímula foí Bayes' theoíem is

$$P(A|B) = \frac{P(B|A)P(A)}{P(B)}$$

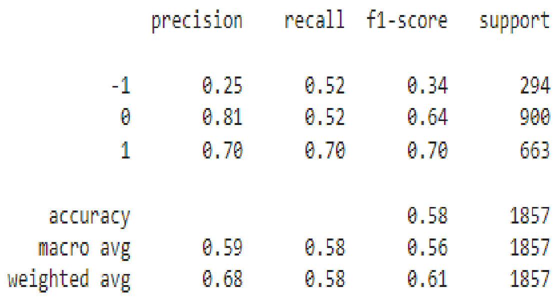
Where, P(A|B) is Posterior probability: Probability of hypothesis A on the observed event B. P(B|A) is Likelihood probability: Probability of the evidence given that the probability of a hypothesis is true. P(A) is Prior Probability: Probability of hypothesis before observing the evidence. P(B) is Marginal Probability: Probability of Evidence.

6.1.1.1 Naive Bayes in Spark

First we imported Naïve Bayes from the ml classification package then we imported the time which helps in calculating the time of execution of an algorithm. Then we used time function to start it. Then we create the Naive Bayes model as nb by smoothing as 1 which helps tackle the problem of zero probability in Naive Bayes. Then we train the model by fitting the train dataset after that we predict by using our model for test data after that we can stop time and print it then we calculate the accuracy of the predictions with the actual values. To calculate accuracy we used multi class evaluator function because here in output we have 3 classes which come under multi class. The accuracy of the Naive Bayes is 70%. Out of the Naive Bayes method is 0.7041344064 accuracy and 1.87 seconds is the time taken to run.

6.1.1.2 Naïve Bayes in ML

First we imported GuassianNB from the sklearn package then we imported the time which helps in calculating the time of execution of an algorithm. Then we used time function to start it. Then we create the GuassianNB function model nb. Then we train the model by fitting the X_train and y_train after that we predict by using our model for X_test data after that we can stop time and print it then we calculate the accuracy score function for the predictions with the actual values. Then we used the classification_report function which gives the output of precision, recall, f1-score and support values for negative, neutral and positive classes. It also gives the macroaverage and weights of the accuracy.

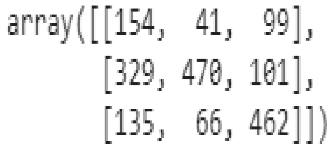


```
              precision    recall  f1-score   support

          -1       0.25      0.52      0.34       294
           0       0.81      0.52      0.64       900
           1       0.70      0.70      0.70       663

    accuracy                           0.58      1857
   macro avg       0.59      0.58      0.56      1857
weighted avg       0.68      0.58      0.61      1857
```

Fig 1



```
array([[154,  41,  99],
       [329, 470, 101],
       [135,  66, 462]])
```

Fig 1.1

Fig 1 shows Naive Bayes model output. Precision should ideally be 1 (high) for a good classifier. Precision becomes 1 only when the numerator and denominator are equal. Here category 0 and 1 are near to one so category 0 and 1 are predicted good. Category 0 and 1 FP (false positive) is less. category -1 false positive more. Recall should be high (1) for a good classifier if false negative decreases, recall will increase. So, category 1 has less false negatives others has more false negatives. F1 Score becomes 1 only when precision and recall are both 1. F1 score becomes high only when both precision and recall are high. Category 0 and 1 has more precision, recall. F1 score is the harmonic mean of precision and recall and is a better measure than accuracy. It gives 58. 48 % accuracy. It took 0.99 seconds to run.

Fig 1.1 shows the confusion matrix of the Naive Bayes model. 154 data are correctly predicted as positive. 470 data predicted correctly as negative.462 neutral data predicted

6.1.2 SVM

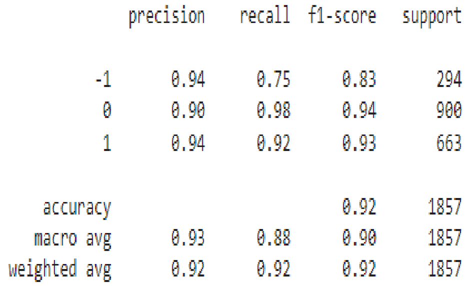
Support Vector Machine (SVM) is a supervised machine learning algorithm that can be used for both classification or regression challenges. However, it is mostly used in classification problems. In the SVM algorithm, we plot each data item as a point in n-dimensional space (where n is a number of features you have) with the value of each feature being the value of a particular coordinate. Then, we perform classification by finding the hyper-plane that differentiates the two classes very well (look at the below snapshot).

6.1.2.1 SVC in spark

First we imported SVC from the ml classification package then we imported the time which helps in calculating the time of execution of an algorithm. Then we used time function to start it. Then we create the SVC model as lsvc by maxIter as 10.It will take 10 as maximum number of iterations. Then we train the model by fitting the train dataset after that we predict by using our model for test data after that we can stop time and print it then we calculate the accuracy of the predictions with the actual values. To calculate accuracy we used multi class evaluator function because here in output we have 3 classes which come under multi class. The accuracy of the SVC is 76%. Out of the SVM method is 0.763 accuracy and 6.86 seconds is the time taken to run.

6.1.2.2 SVC in ML

First we imported linearSVC from the sklearn package then we imported the time which helps in calculating the time of execution of an algorithm. Then we used time function to start it. Then we create the LinearSVC function model as support. Then we train the model by fitting the X_train and y_train after that we predict by using our model for X_test data after that we can stop time and print it then we calculate the accuracy score function for the predictions with the actual values. Then we used the classification_report function which gives the output of precision, recall, f1-score and support values for negative, neutral and positive classes. It also gives the macroaverage and weights avg of the accuracy.

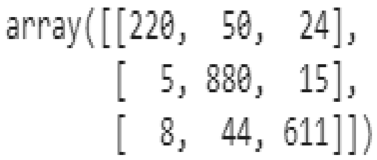


```
              precision    recall  f1-score   support

          -1       0.94      0.75      0.83       294
           0       0.90      0.98      0.94       900
           1       0.94      0.92      0.93       663

    accuracy                           0.92      1857
   macro avg       0.93      0.88      0.90      1857
weighted avg       0.92      0.92      0.92      1857
```

Fig 2



```
array([[220,  50,  24],
       [  5, 880,  15],
       [  8,  44, 611]])
```

Fig 2.1

Fig 2 shows output of SVM model. Precision should ideally be 1 (high) for a good classifier. Precision becomes 1 only when the numerator and denominator are equal. Here all category 1 is near to one so this is a good model for this dataset. Recall should be high (1) for a good classifier if false negative decreases, recall will increase. So, category 0 and 1 has less false negatives others has more false negatives. F1 Score becomes 1 only when precision and recall are both 1.
F1 score becomes high only when both precision and recall are high. Category 0 and 1 has more precision, recall. Accuracy of this model is 92.13%. It took 0.25 seconds to run.

Fig 2.1 shows the confusion matrix of the svm model using python.220 data correctly predicted as positive. 880 data correctly predicted as negative. 611 correctly predicted as neutral. Others are misclassified.

6.1.3 Decision Tree

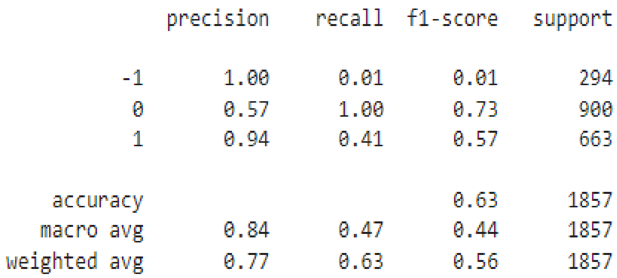
Decision tree is a type of supervised learning algorithm that can be used for both regression and classification problems. The algorithm uses training data to create rules that can be represented by a tree structure.
Like any other tree representation, it has a root node, internal nodes, and leaf nodes. The internal node represents condition on attributes, the branches represent the results of the condition and the leaf node represents the class label.

6.1.3.1 Decision Tree in spark

First we imported Decision Tree classifier from the ml classification package then we imported the time which helps in calculating the time of execution of an algorithm. Then we used time function to start it. Then we create the Decision Tree Classifier model as dt by maxdepth as 3. It will take 3 as depth. Then we train the model by fitting the train dataset after that we predict by using our model for test data after that we can stop time and print it then we calculate the accuracy of the predictions with the actual values. To calculate accuracy we used multi class evaluator function because here in output we have 3 classes which come under multi class. The accuracy of the SVC is 40 %. Out of the SVM method is 0.4023 accuracy and 7.32 seconds is the time taken to run.

6.1.3.2 Decision Tree in ML

First we imported Decision Tree classifier from the sklearn package then we imported the time which helps in calculating the time of execution of an algorithm. Then we used time function to start it. Then we create the Decision Tree function model as clf_model using 'Gini' and random_state as 42, maximum depth 3 minimum sample leafs 5. Then we train the model by fitting the X_train and y_train after that we predict by using our model for X_test data after that we can stop time and print it then we calculate the accuracy score function for the predictions with the actual values. Then we used the classification_report function which gives the output of precision, recall, f1-score and support values for negative, neutral and positive classes. It also gives the macroaverage and weights avg of the accuracy.



```
              precision    recall  f1-score   support

          -1       1.00      0.01      0.01       294
           0       0.57      1.00      0.73       900
           1       0.94      0.41      0.57       663

    accuracy                           0.63      1857
   macro avg       0.84      0.47      0.44      1857
weighted avg       0.77      0.63      0.56      1857
```

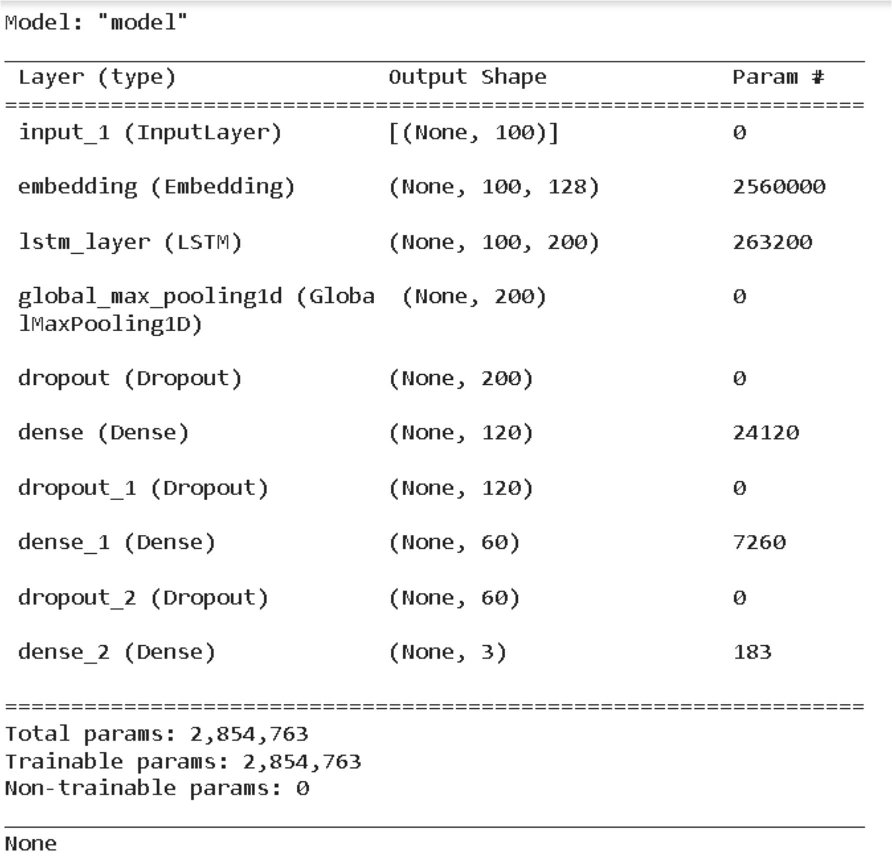
Fig 3

Fig 3 shows output of Decision Tree in python. Precision should ideally be 1 (high) for a good classifier. Precision becomes 1 only when the numerator and denominator are equal. Here -1 category is equal to one so there is no false positive in category -1. I category has less false negatives and 0 has more false negatives. Recall should be high (1) for a good classifier if false negative decreases, recall will increase. So, category 0 does not have false negatives, others have more false negatives. F1 Score becomes 1 only when precision and recall are both 1. F1 score becomes high only when both precision and recall are high. Category 0 and 1 has more precision, recall. It gives 63% accuracy and it took 3.56 seconds to run.

6.2DNN-LSTM

LSTM special kind of recurrent neural network that is capable of learning long term dependencies in data. This is achieved because the recurring module of the model has a combination of four layers interacting with each other.

LSTM has feedback connections. It can process not only single data points, but also entire sequences of data. A common LSTM unit is composed of a cell, an input gate, an output gate and a forget gate. The cell remembers values over arbitrary time intervals and the three gates regulate the flow of information into and out of the cell.



```
Model: "model"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 input_1 (InputLayer)        [(None, 100)]             0

 embedding (Embedding)       (None, 100, 128)          2560000

 lstm_layer (LSTM)           (None, 100, 200)          263200

 global_max_pooling1d (Globa  (None, 200)              0
 lMaxPooling1D)

 dropout (Dropout)           (None, 200)               0

 dense (Dense)               (None, 120)               24120

 dropout_1 (Dropout)         (None, 120)               0

 dense_1 (Dense)             (None, 60)                7260

 dropout_2 (Dropout)         (None, 60)                0

 dense_2 (Dense)             (None, 3)                 183

=================================================================
Total params: 2,854,763
Trainable params: 2,854,763
Non-trainable params: 0
_________________________________________________________________
None
```

Fig 6.1

Fig 6.1 shows the LSTM DNN model created for this dataset. It gives 48.46 % accuracy.

7. Visualization

7.1 Histogram

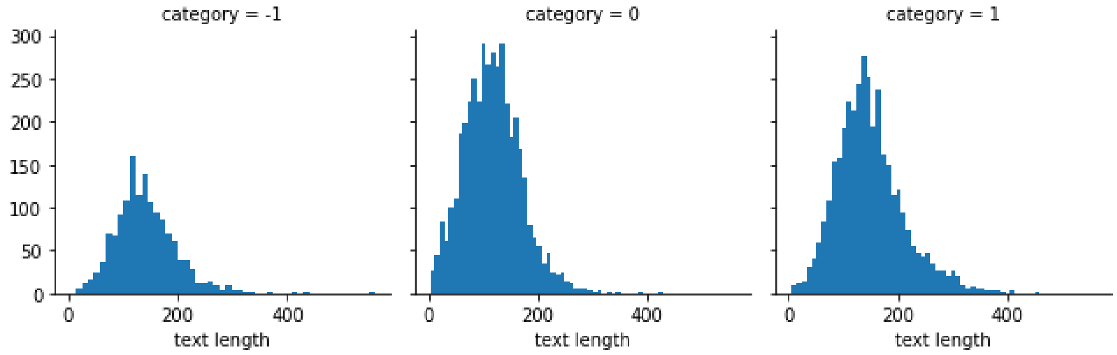


Fig 7.1

Fig 7.1 shows Histogram between text length and category. X axis is category and y axis are text length. There is no category distributed normally except category 1 (neutral). category 1 distributed normally. Positive speech has more text length.

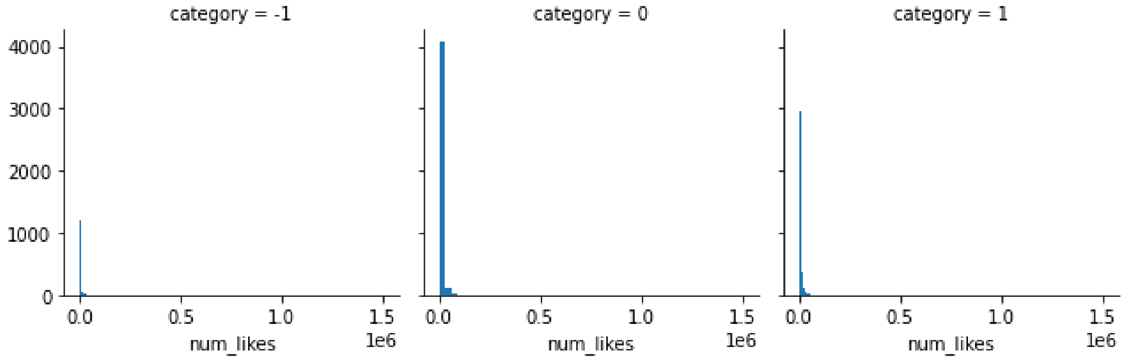


Fig 7.2

Fig 7.2 shows the relationship between category and number of likes. x axis is category and y axis are no of likes. Neutral type comments have more likes compared to other categories.

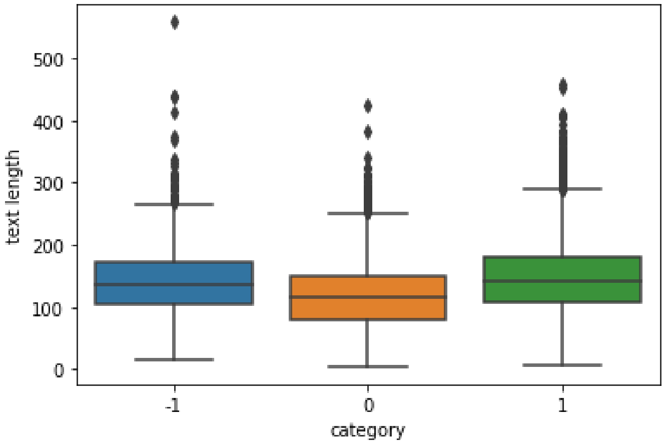


Fig 7.3

Fig 7.3 shows a boxplot between category and text length. Negative comments do not have normal distribution in text length. If a point lies inside the whisker then it is normal otherwise it is outlier. From the above fig 7.3 we can say that we have more outliers.

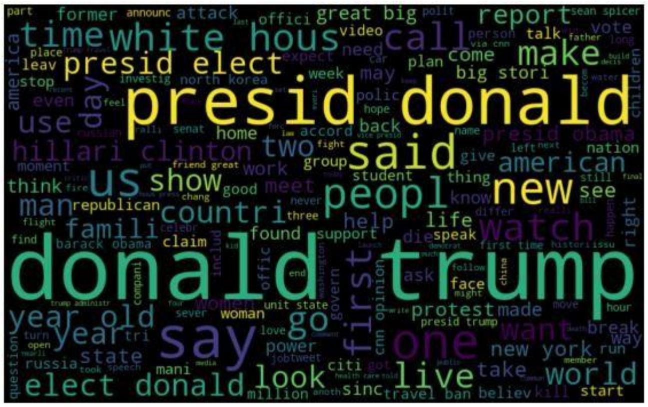


Fig 7.4

Fig 7.4 shows a world clod of commonly used words in the dataset. From fig7.4 we can say that Donald, trump is used more in the Facebook comments.

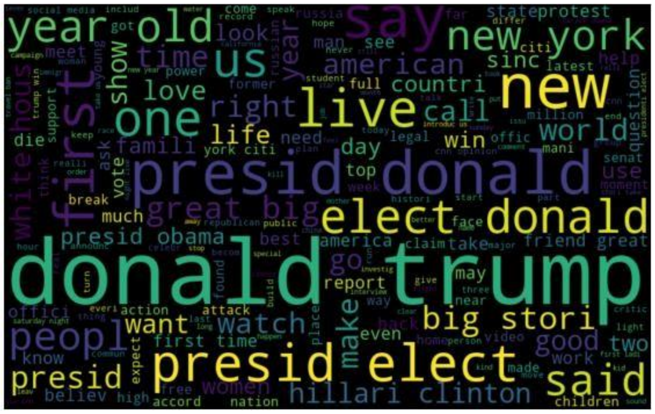


Fig 7.5

Fig 7.5 showed a word cloud of commonly used words in positive comments. From the fig 7.5 we can say it as Donald, Trump is large because max positive comments has Donald and Trump. If we wipe of the Donald and Trump because Donald and Trump are Noun then we will have elect, said, first, say, new, etc. are having more count in the comments.

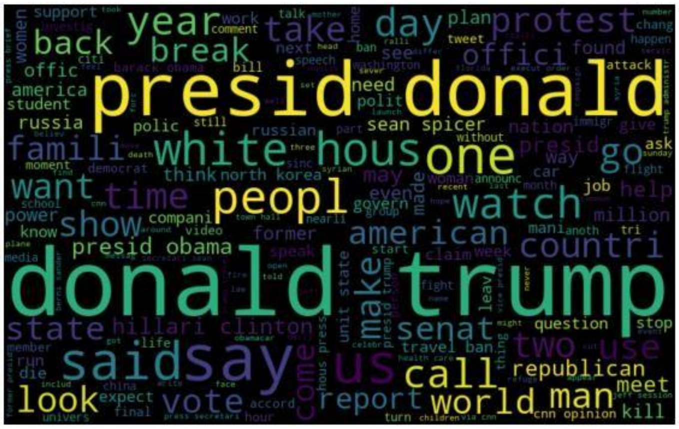


Fig 7.6

Fig 7. 6 shows a word cloud of commonly used words in negative comments. From the fig 7.6 we can say it as Donald, Trump is large because max negative comments has Donald and Trump. If we wipe of the Donald and Trump because Donald and Trump are Noun then we will have protest, break, etc. having more count in the comments.

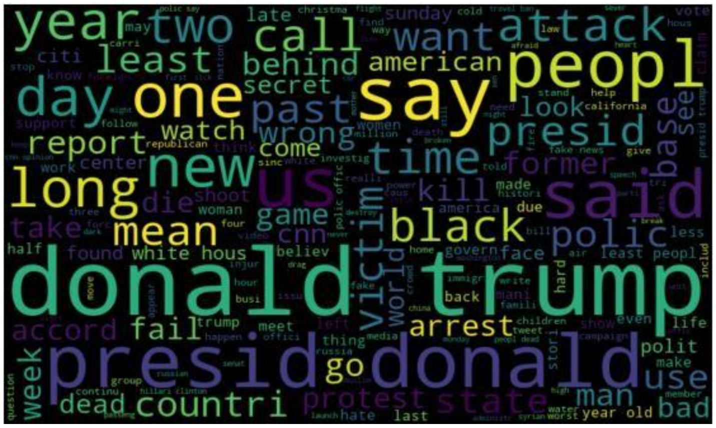


Fig 7.7

Fig 7.7 shows a word cloud of commonly used words in neutral comments. From the fig 7.7 we can say it as Donald, Trump is large because max neutral comments has Donald and Trump. If we wipe of the Donald and Trump because Donald and Trump is Noun then we will have black, long, means, one, etc. having more count in the comments.

Visualization using Tableau

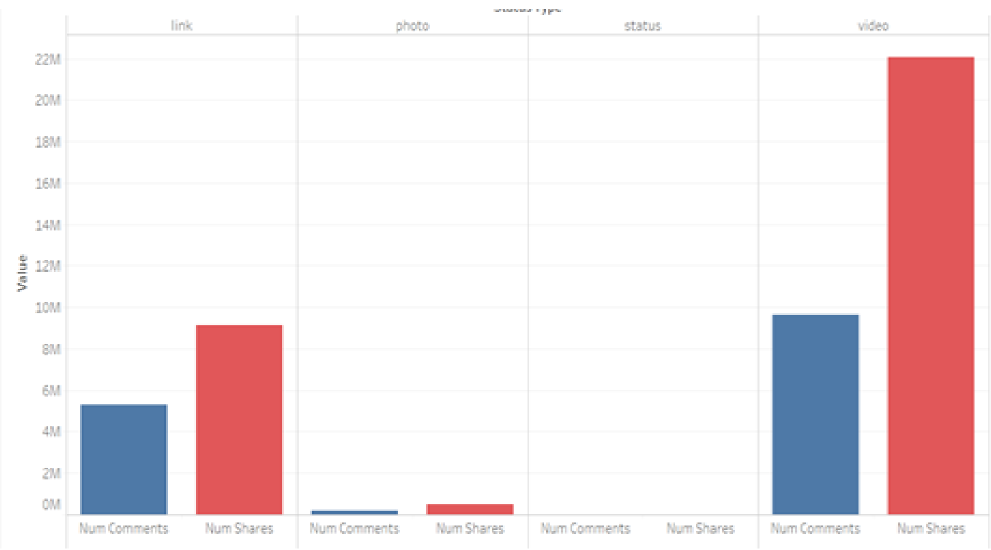


Fig 7.8

Fig 7.8 shows a bar graph of num shares and comments vs types of tweets.

Y axis is values. Blue bar graph is share, and the red box is comments.

X axis is types of tweets (Photo, video, link, status).

From the above graph we conclude that num_shares, num_comments are more for the videos compared to link, photo, status. If we want to do any ads or to attract the audience then we can use the video format.

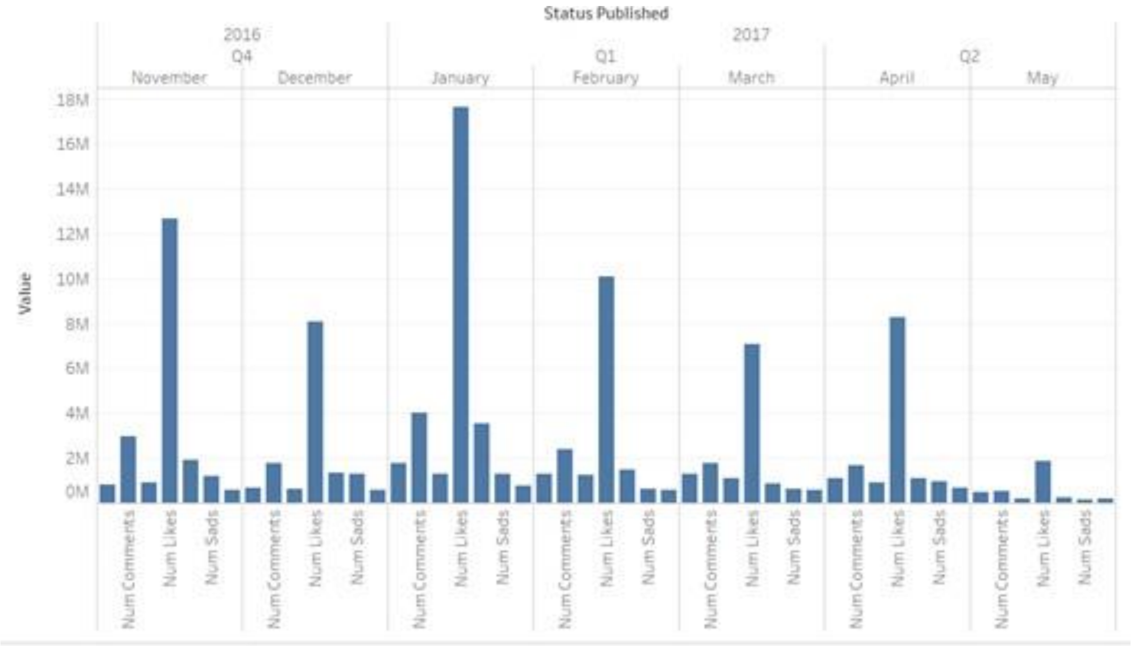


Fig 7.9

Fig 7.9 shows a bar graph of status published (no of likes, no of shares, no of sads) vs values in different months. X axis is status group by months, Y axis is values. From the above graph we can say that Dec-Jan num_likes is very high. Nov - Dec are the months of New year and Christmas so there are many interesting posts, sales, etc...

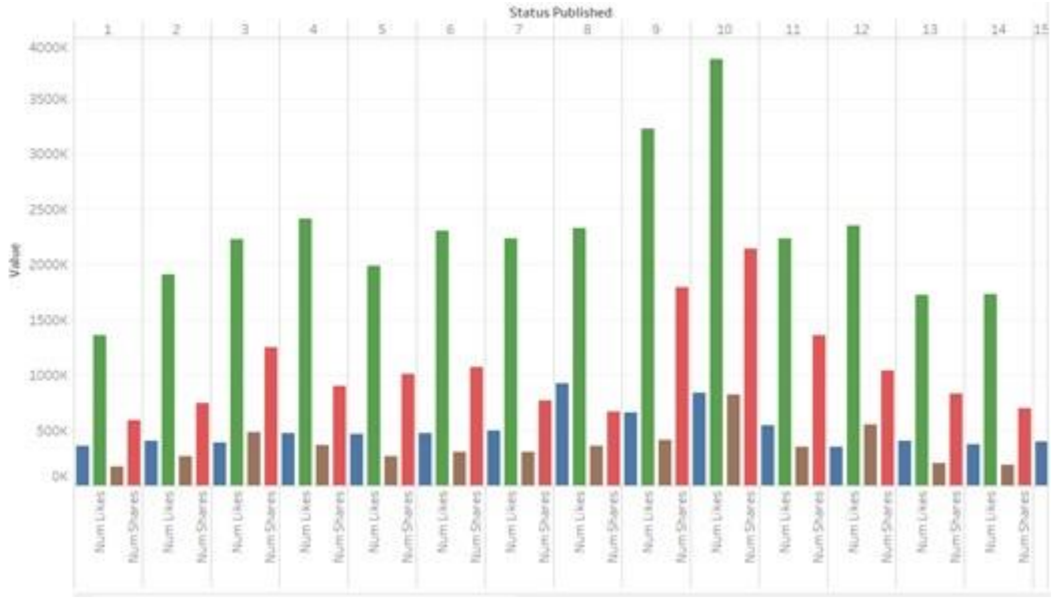


Fig 7.10

Fig 7.10 shows a bar graph of status published (no of likes, no of shares) vs values in different Days. X axis is status grouped by Days and Y axis is values.

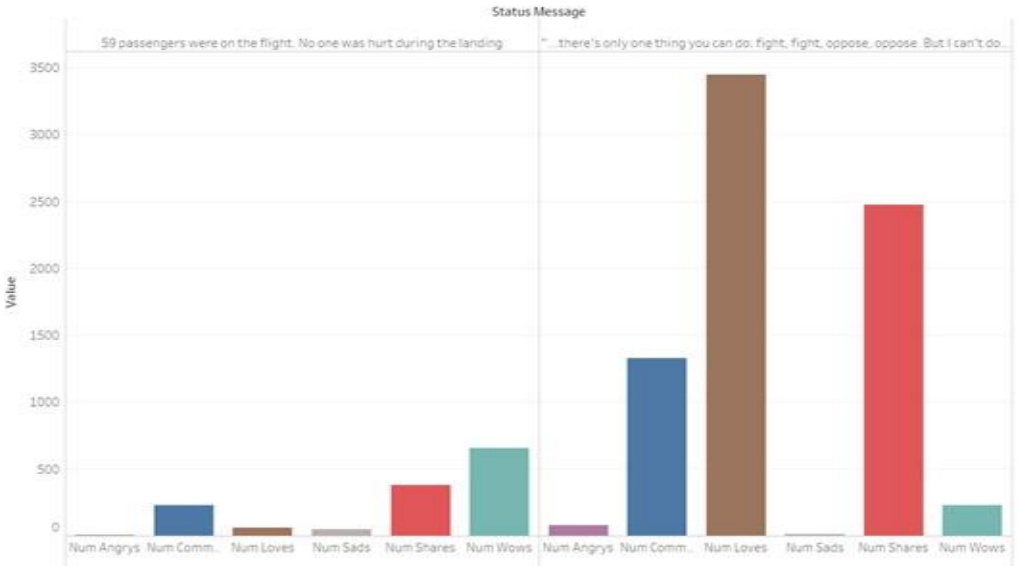


Fig 7.11

Fig 7.11 shows a bar graph of status of message (no of likes, no of comments, no of sads, no of shares, no of wows) of 2 tweets vs values. X axis is status of message, tweets and y axis is values. Tweet 2 has got more likes, comments, shares and wows compared to tweet 1.

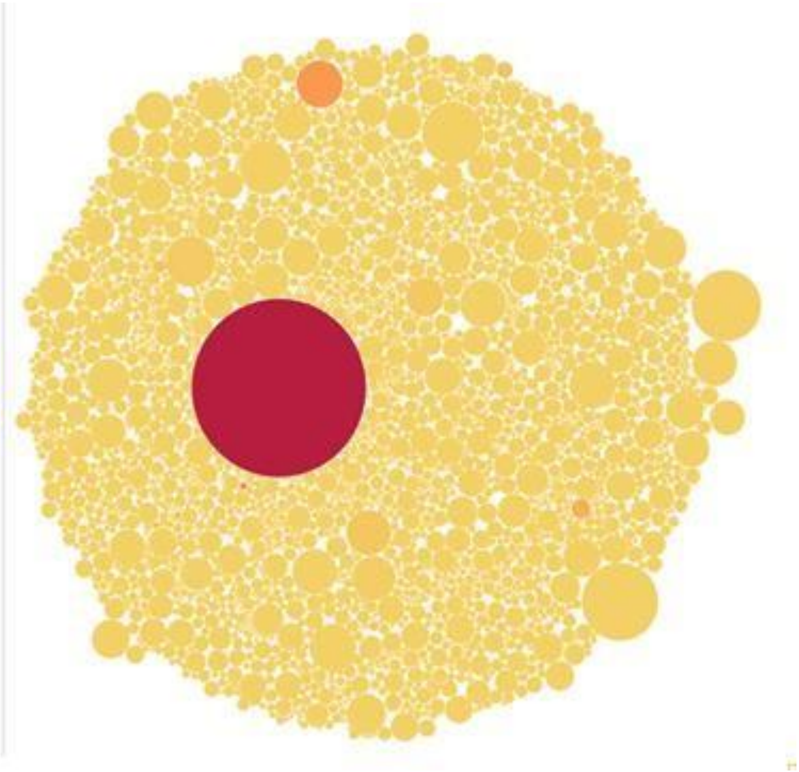


Fig 7.12

Fig 7.12 shows bubble charts. Bubble color shows no.of anger and size shows no.of love of tweets.

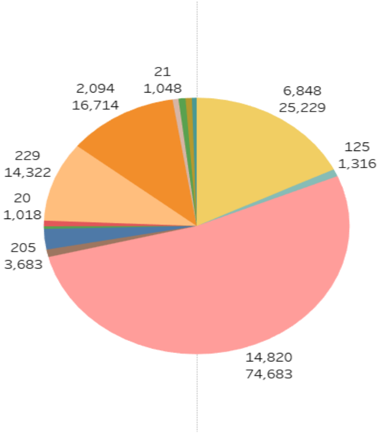


Fig 7.13

Fig 7.13 shows a pie chart.

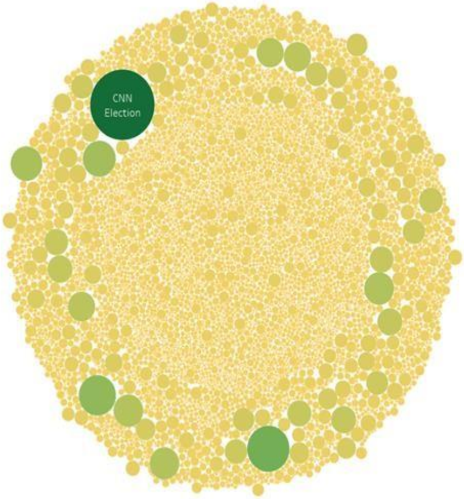
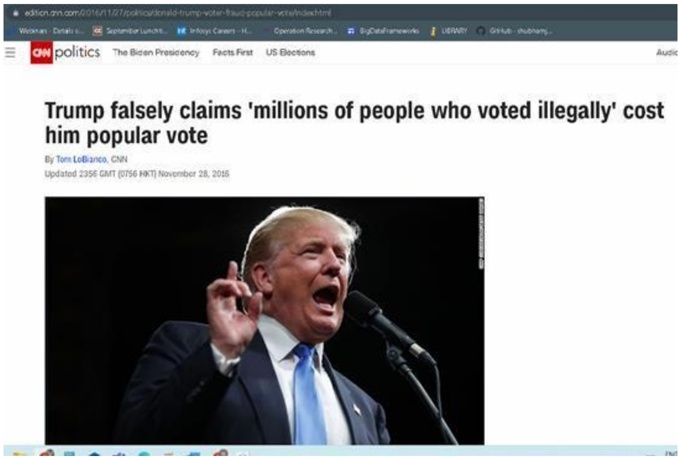


Fig 7. 14



Trump falsely claims 'millions of people who voted illegally' cost him popular vote

By Tom LoBianco, CNN

Updated 2356 GMT (0756 HKT) November 28, 2016

Fig 7.15

Fig 7.14 shows Dashboard, it is directly connected to the web in such a way When we click on some point then it will directly go to chrome and it opens that point related page like Fig 7.15.

Already in the word cloud we can see that we have large word or no.of count is big is Donald Trump and max comments or conversations is about Donald Trump

9. Results

Accuracy comparison between pyspark ML algorithm and ML algorithm.

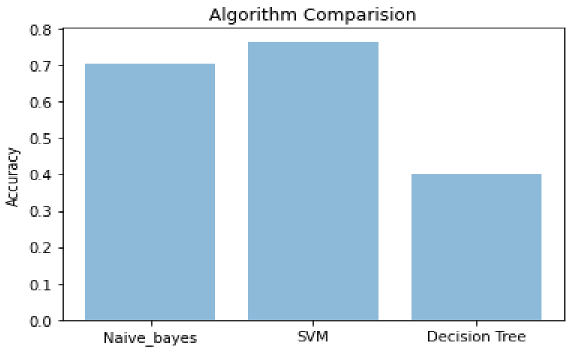


Fig 9.1

Fig 9.1 shows Accuracy comparison in pyspark ML algorithm. x axis is Algorithms and y axis is accuracy.

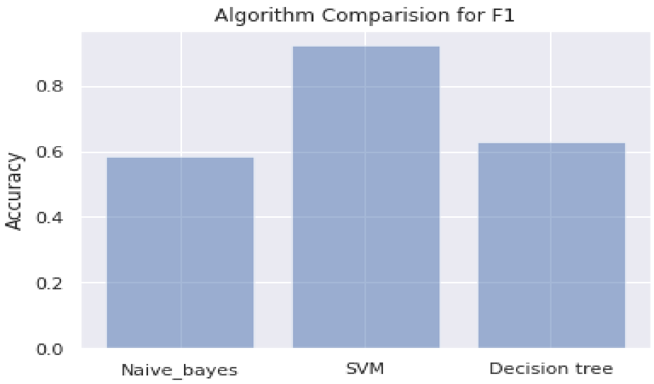


Fig 9.2

Fig 9.2 shows Accuracy comparison of ML algorithm. x axis is Algorithms and y axis is accuracy.

From the above fig9.1 and fig9.2 SVM accuracy is more than other accuracies. By Comparing both figures we can say that spark is better for Naïve Bayes and ML is better for SVM and Decision Tree.

Run time comparison between pyspark ML algorithm and ML algorithm.

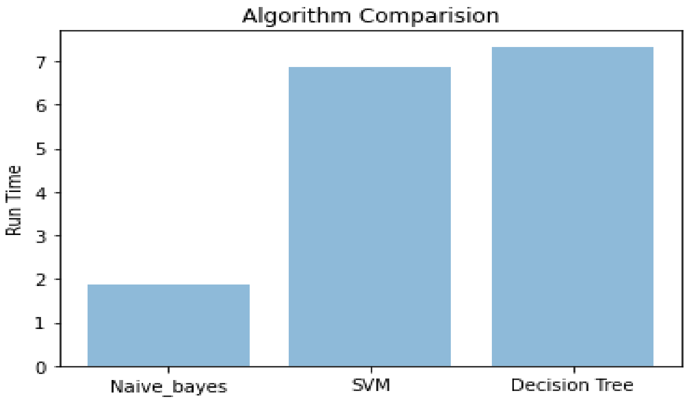


Fig 9.3

Fig 9.3 shows Run time comparison of different models. X axis is algorithm and y axis is Run time.

Runtime for the ML algorithms such as Naive Bayes is 0.99938344955444, SVM is 0.25925755500793457, Decision tree is 3.569929361343384

From the above runtimes of ML and pyspark algorithms we can say that ML has taken less time compared to pyspark.

10 Conclusion and future work

By comparing Accuracy of python ML algorithms and ML algorithms using pyspark. SVM has high accuracy in both methods. So SVM is the best model for this dataset. Comparing normal and pyspark ML models there is not much difference. Both give accuracy with point difference.

By comparing Run time of python ML models and pyspark ML models. Python ML algorithms has taken less time due to the size of the dataset which is 10000. It will be different if we use a large dataset. From this we can say that spark works worst for the small dataset we can use spark for large dataset to get good accuracy and less runtime.

In future we will try to use different Deep learning models, different ML models with pyspark

11.References

[1] ***Alvaro Ortigosa, José M. Martín, Rosa M. Carro***, "Sentiment analysis in Facebook and its application to eLearning" https://doi.org/10.1016/j.chb.2013.05.024.

[2] ***Akhtar, Nadeem, Hira Javed, and Geetanjali Sengar***. "Analysis of Facebook social network." https://ieeexplore.ieee.org/abstract/document/6658034

[3] ***Nguyen, K., & Tran, D. A.*** **"**An analysis of activities in Facebook" https://ieeexplore.ieee.org/abstract/document/5766497

[4] ***Troussas, C., Virvou, M., Espinosa, K. J., Llaguno, K., & Caro, J***. "Sentiment analysis of Facebook statuses using Naive Bayes classifier for language learning." https://ieeexplore.ieee.org/abstract/document/6623713

[5] ***Ahkter, J. K., & Soria, S***. "Sentiment analysis: Facebook status messages". *https://nlp.stanford.edu/courses/cs224n/2010/reports/ssoriajr-kanej.pdf*

[6] ***Sandoval-Almazan, R., & Valle-Cruz, D.*** "Facebook impact and sentiment analysis on political campaigns" https://dl.acm.org/doi/abs/10.1145/3209281.3209328?casa_token=qdyINRMfItYAAAAA:pyy2PNT_uDDwhZAup-d_pMJgTFfaQIBdcc8Ns7dqQOj7_iFDXZ5_10BYIH44xvL18D1wOrUBpbt9FNY