Sentiment Analysis, Tweet Analysis and Visualization on Big Data Using Apache Spark and Hadoop

Sujala D Shetty
Assistant Professor, BITS Pilani, Dubai Campus, DIAC, Dubai. U.A.E.
E-mail: sujala@dubai.bits-pilani.ac.in

Abstract. Big Data lets companies generate new growth opportunities and whole new types of industries that can continually change and mushroom data from the market that can merge and evaluate. In today’s world, Apache Spark and Hadoop are the most popular and effective tools used to conduct big data analysis. This paper aims to perform two types of analysis—Sentiment Analysis of tweets and Tweet Analysis. For conducting sentiment analysis, we used a dataset created by Stanford University which contains 4 fields – ID of the tweet, Sentiment of the tweet, Sentiment Source and the Tweet itself. Two machine learning algorithms performance are compared these are Logistic Regression and Random Forest Classifier and observe which algorithm provides better accuracy and why. A data visualization tool -Tableau is used to view the sentiments of tweets by filtering the text of tweets by words and hashtags. For tweet analysis, we use Python and a library called Tweepy to download tweets from the Twitter API and perform analysis using another data visualization tool called Power BI. We conduct analysis on the basis of three parameters – user followers count, retweet count and favourite count. Finally, we draw a comparison on the performance and user-friendly nature of both data visualization tools – Power BI and Tableau.

1. Introduction
Data stands for unique and distinct pieces or units of information that can be used to conduct analysis and thereby construct meaningful facts and figures. The term “Big Data” is similar to small data but is different in the aspects of “size”. The problem with having bigger chunks of data is that it requires different approaches- techniques, tools and architecture. Big data analysis aims to solve new problems or old problems in a more efficient and effective manner. Big data generates value from the storage and processing of very large quantities with digital information that cannot be analysed with traditional computing techniques. Just like many new information technologies, big data can result in huge cost reductions, substantial improvements in the time required to perform a computing task or new product and service offerings [1].

Industries are using Hadoop broadly to analyse their datasets. This is because the Hadoop architecture is based on a basic programming model (MapReduce) which makes for a scalable, modular, fault-tolerant and cost-effective computing approach. To speed up the Hadoop computing programme process, Spark was implemented by Apache. Hadoop is used in two ways by Spark: one is storage and the second is computation. Since Spark has its own device for cluster control, it only uses Hadoop for storage purposes [2].

Apache Spark, developed for rapid computing, is a lightning-fast cluster computing technology. It is based on Hadoop MapReduce and expands the MapReduce paradigm to be used successfully for more computational styles, including dynamic queries and stream processing. Spark's key feature is its in-
memory cluster computation that improves an application's processing speed. Spark is developed to cover a wide variety of workloads, including batch processes, iterative algorithms, and streaming collaborative queries. It decreases the management pressure of managing different resources, in addition to serving all these workloads in the respective system [2].

"Social networks are one of the key sources of data today. One of the most important big data echo solutions - Apache Spark and Python - would be a real-life example of working with, processing and generating information from social network data in real time. Here, we carry out data analysis on tweets and try to generate results in order to establish meaningful facts and figures. The objectives of this paper are as follows:

a. Conduct “Tweet Analysis” using a downloaded dataset using a data visualization tool called “Power BI” and to find - the total count of followers for a particular user, the number of retweets and number of favorites a particular tweet receives.
b. Conduct “Sentiment Analysis” on the downloaded dataset using two Machine learning algorithms - Random forest classifier and Logistic Regression.
c. To evaluate and compare the efficiency and performance of two ML algorithms on the dataset.
d. Perform real-time analysis by extracting live tweets from the Twitter streaming API using Tweepy in Python.
e. Uploading the live tweets dataset to conduct tweet analysis using a data analytics tool named “Tableau”.
f. Filtering the tweet dataset with respect to hashtags and keywords to generate graphs representing the positive and negative sentiments of the people.
g. Comparing the performance and user-friendly nature of two data analytics & visualization tools – Power BI and Tableau.

The paper is structured as in section 2 is discusses background, section 3 discusses methods, Section discusses about dataset, section 5 discusses results and last but not least section 6 concludes the paper.

2. Background and Tools

For conducting the sentiment analysis, a Sentiment140 dataset was used which was created by Stanford University. It contains 4 fields – Item ID, Sentiment, Sentiment Source and Sentiment Text. (Here, the sentiment source is Sentiment140). The sentiment text refers to the tweet and is differentiated by two sentiments, 1- positive sentiment and 0- negative sentiment. The dataset contains 1,933 entries which are tweets. The real-time analysis was carried out by extracting tweets with the hashtag “Netflix” wherein 10,000 tweets were collected in a time period of approximately 12 seconds. However, the tweet analysis was carried out only on the top 200 entries.

A brief description on the tools that have been used to carry out the data analysis is as follows-

2.1. Apache Spark

Apache Spark is a framework for cluster computing developed for fast and general purposes. At its heart, Spark is a ‘computational engine’ responsible for designing, delivering, and controlling applications consisting of several computational tasks across many worker computers, or a Spark computing cluster built to be highly open, providing basic Python, Java, Scala, SQL, API, and large number of built-in libraries [3].

2.2. PySpark

Apache Spark is written in Scala programming language. We can work with Resilient Distributed Datasets (RDDs) which is the fundamental data structure in Spark and can be used in Python programming language [4].

- **Jupyter Notebook**: The Jupyter Notebook is the IDE for development the code [5]
- **Tweepy**: It is used to access Twitter API [6]
- **Logistic Regression**: It used to predict and classify where output variable is dichotomous in nature [7] [8]
• **Random Forest Classifier**: It is an enhanced algorithm of decisions trees. It used different decision trees. When constructing each individual tree, it uses bagging and feature randomness to try to construct an uncorrelated forest of trees whose committee forecast is more precise than that of any individual tree [9][10]

Tableau and Power BI is also used for Data visualization and presentation.

The next section i.e., Section 3 focuses on all the methods involved to conduct the big data analysis.

3. Research Methodology

3.1 Sentiment Analysis of Twitter Data

It is a method used to carry out an analysis of the sentiment of the text in terms of positive or negative. The downloaded dataset (by Stanford University) is divided into Training data and testing data. Testing data is then predicted and the accuracy of the model is calculated using two machine learning algorithms – Logistic regression and Random forest classifier. The aim here is to evaluate the performance of the algorithms as to which algorithm provides better accuracy and why [11].

**Describing the dataset**: The dataset used is “Sentiment140” which was created by Stanford University. It is a CSV file (tweets.csv) with emoticons removed. The format of the data file contains 1933 entries with 4 fields namely:

- 0 – The id of the tweet (for example: 234) – **ItemID**
- 1 - The polarity/sentiment of the tweet (0 – negative, 1- positive) – **Sentiment**
- 2 – Sentiment source (here, Sentiment140) – **SentimentSource**
- 3 – The text of the tweet (for example: cheese is fantastic #loveit) – **SentimentText**

The dataset created by Stanford University was created automatically instead of manually annotating the tweets. Their approach consists of an assumption wherein tweets with positive emoticons like the smiley face “:)” were considered as positive and tweets with negative emoticons like the sad face “:(” are taken as negative. The collection of tweets was done using the Twitter Search API by keyword search.

**System Design**: An open-source web application called as “Jupyter Notebook” was used that allowed us to code using the language “PySpark” which is an association of Python and Apache Spark.

**Spark Cluster Setup**: A spark cluster was setup in standalone mode on my laptop with the following specifications – 4 GB memory (RAM), 500 GB hard drive and Intel Pentium processor. To run a Spark cluster on Windows, we start the master and workers manually.

- **Step 1**: Run the following command in the same directory where your Apache spark is installed. `./bin/spark-class org.apache.spark.deploy.master.Master` Master will give [spark://HOST:PORT – URL] which is the address for starting the application.
- **Step 2**: Initiate the operation of one or more processes/applications i.e., a worker and connect them to the master. Open a new command prompt and run the code below. `./bin/spark-class org.apache.spark.deploy.worker.Worker spark://IP PORT`

Note: Mention the IP PORT of your master node. We repeated the same step 4 times to create 4 worker nodes and connected them to the master node.

- **Step 3**: Here, we connect application vs cluster by running the code below in command prompt. `./bin/spark-shell –master spark://IP:PORT`

**Sentiment Analysis of Twitter Data**: We now proceed with conducting sentiment analysis on our dataset (Sentiment140). The steps for the same procedure are highlighted below.

**Import modules and create spark session**: Here, we are importing different modules like Logistic Regression, HashingTF etc.
Read data file into Spark data frame: We read our dataset which is a csv file i.e. “tweets.csv” into the data frame. We have 4 fields – Item ID, Sentiment, Sentiment Source and Sentiment Text. The sentiments here denote positive (1) and negative (0) respectively. The Sentiment Text refers to the tweet data in the dataset.

Divide data into training and testing data: In machine learning we need to train our dataset, hence we divide our data into training data and testing data with 70% for training and the remaining 30% for testing. We use the function .randomSplit to do the same. Index ‘0’ denotes the training data whereas Index ‘1’ stands for the testing data. Then we perform count() function to find the total number of data rows for training data and testing data. The output shows a total of 1303 rows for training and 629 rows for testing.

Prepare training data: Here, we carry out the pre-processing stage of data. We separate the tweet into individual words using tokenizer and also remove the stop words from tweets.

Converting words feature into numerical features: Machine learning deals with numerical features rather than text features. Therefore the conversion is done using the Hashing function using Austin Appleby’s Murmur Hash algorithm. Murmur hashing is a non-cryptographic hash function which is used for hash based look-ups, it uses 3 basic operations as a whole Multiply, Rotate and XOR. We proceed to train our classifier model using training data and make use of two machine learning algorithms – “Random Forest Classifier” and “Logistic Regression”. We then compare the accuracies of both algorithms after we predict the testing data [12].

Random Forest Classifier: We train our model with respect to the same algorithm, prepare the testing data and finally calculate the accuracy of the model. The steps are provided below:

Train our classifier model using training data:
Here, we use the machine learning algorithm “Random Forest Classifier” to perform classification on our dataset. The line of code containing “model.fit” is used to train the data that we received after performing the Murmur Hash function [13].

- Prepare testing data: A similar procedure with respect to training data is followed wherein we first perform tokenization using the “tokenizer” function to separate sentences into individual words. Stop words are removed from the tweets. We then perform the conversion of text into numerical features using the Murmur Hash function.

- Predict testing data and calculate the accuracy model: Our model predicts on basis of the output we get after carrying out the Murmur Hash on the testing data. We calculate the accuracy of the model by comparing the final predictions and count the number of correct predictions. The total data present in the testing data is noted. We divide correct number of predictions by total data to calculate the accuracy of the model. The output shows us that 274 rows out of a total 570 rows were correctly predicted with an accuracy of 48.07%.

Logistic Regression: We now move ahead with logistic regression. A similar procedure is followed wherein we train our model with respect to the same algorithm, prepare the testing data and finally calculate the accuracy of the model. The steps are provided below:

- Train our classifier model using training data: In order to perform classification, we use the machine learning algorithm known as “Logistic Regression”. The line of code containing “lr.fit” is used to train the data that we received after performing the Murmur Hash function.

- Prepare testing data: A similar procedure with respect to training data is followed – tokenization, stop words removal and conversion of text to numerical features.

- Predict testing data and calculate the accuracy model: Prediction of the model is conducted in the same process carried out during the calculation of accuracy for random forest classifier. The output shows us that 622 rows out of a total 629 rows were correctly predicted with an accuracy of 98.88%.

3.2 Tweet Analysis
This part mainly focuses on collecting live tweets which contain “#Netflix” from the Twitter database using the Twitter API. This step is done using Tweepy library in Python, wherein the tweet data is first saved as a JSON file and then converted into a CSV file. We then use a data visualization tool known
as Power BI to import the tweet data and analyze the tweets and find the Retweet count, Favourite count, Total number of User Followers, etc. The data visualization part is further elaborated in Section IV.

4. Data Collection
Tweets for this review are compiled using Tweepy's python library from the Twitter website. One has to create a Twitter-Developer account in order to view the tweets. You have to include the specifics of the main use case, motive, business purpose of your use, details of the studies you intend to do, and the approaches or strategies to build the account.

Once we have our Twitter developer account an app needs to be created in their interface which generates API KEY (Consumer), API SECRET KEY(Consumer), ACCESS TOKEN, and ACCESS TOKEN SECRET, which we will use to log in through terminal.

We create a new app with the following details:
- **App Name** – iUniApp
- **Description** - App for iUni which will be used to carry out streaming analysis of tweets for research purpose.
- **Website URL** - https://www.iuni.com

**Downloading the Tweets:**
The tweets can now be downloaded from the twitter database using the Tweepy library in Python. There are two main ways of searching for tweets with Tweepy. One is search by a hashtag (or a keyword) and the second is search by the id of a tweet (id of each tweet is unique). In our analysis, we use the first method which is search by a hashtag, here #Netflix to download the tweets.

We first import the necessary packages into Python and enter the respective API _KEY and API_SECRET generated by the app. We search using the keyword #Netflix and save the tweets in a text file called as ‘twitter.txt’. We used extended-tweet mode to make sure that we get full tweets rather than truncated ones. We download a total of 10,000 tweets in a time frame of 11.33 seconds.

**Conversion of JSON file to CSV file:**
In this stage, data received in JSON format is converted into CSV file and unwanted attributes like indices, meta data and screen name is removed.

We select the following objects- full text of the tweet, retweet count of a tweet, favorite count of a tweet, geographic location, coordinates, place from which tweet is generated, follower count of the user, time and date of tweet & id of the tweet.

We load the Jupyter Notebook CSV-file. In the read csv() feature, it is important to set the encoding parameter as 'unicode escape' since certain tweets often consist of emojis which will find it difficult to encode by default. Duplicate rows in the dataset and redundant columns are discarded.

**Cleaning the tweets:**
There may be any needless symbols in the text of a tweet that are not important to our study. We wipe our tweets, hence. If we can see, URLs (like https://t.co/XK3ZReâ80), numbers and punctuations are some of the needless text and icons to be eliminated. We do this text cleaning using the python re module.

5. Results and Discussion

5.1 Sentiment Analysis of tweets using Tableau
We load the Sentiment140 dataset created by Stanford University into Tableau in order to analyze the text of the tweets. We apply filter on the tweet data based on two parameters namely- Filter by Word & Filter by Hashtag.
5.1.1 Filter by “Word”
Filter tweets with – “cheese” and display the sentiments (1-positive, 0-negative). Next, we create a “packed bubbles graph” to differentiate between tweets that have positive and negative sentiments related to the filter “cheese”.

Bubbles can be shown with different shades represent different sentiments. For e.g., dark blue bubbles represent the negative sentiments of a tweet, for example: here tweet is “cheese is rubbish #nightmare” which has the sentiment value of 0 as it is a negative sentiment. Similarly, the light blue bubbles show the positive sentiments of a tweet, for example: the tweet is “cheese is fantastic #loveit” which has the sentiment of 1 as it is a positive sentiment.

5.1.2 Filter by “hashtag”
We filter the text of tweets with hashtags and perform an analysis of tweets based on their sentiments and number of records.

Case 1: Positive Sentiment (#loveit)
Filter the tweets with “#loveit” which is a positive word in its own; therefore we only encounter positive sentiment tweet data in our graphs and no negative sentiments. As seen in the packed bubbles graph below, we notice that the lighter colored bubbles are smaller in size when compared to the darker colored bubbles. This is because the darker bubbles contain more number of records than the lighter bubbles. All the bubbles represent the positive sentiments of the tweet, here, for example: the text is “I adore jam #loveit” which has the sentiment value 1, as it’s a positive tweet. The number of records of the same tweet is also 1; hence the size of the bubble is smaller.

The tweet “this game is brilliant #loveit” has a positive sentiment with value = 1, but the number of records for the same tweet is 2. Hence the sentiment value for each tweet is added (1+1) which gives us total sentiment value of 2.

Case 2: Negative Sentiment (#hateit)
Filter the tweets with “#hateit” and perform an analysis of tweets based on their sentiments and number of records.

5.2. Tweet Analysis with Power BI
Load the csv file created after collecting tweets using Twitter API and python by searching #Netflix into Power BI. Analysis is carried out on the first 200 rows of our dataset. We use four fields—the text of the tweet, number of retweets for a particular tweet, total number of followers of a particular user and total number of favorites a tweet receives.

Retweet Count by Tweet:
We make an Area Chart with tweet on the X axis and retweet count on the Y axis. The retweet count for a tweet text peaked at 20K in the beginning and gradually decreased with every next possible tweet. One can easily analyze the count for any particular tweet by browsing through the chart. We then find the total number of retweets, the maximum number of retweets and the average of retweets.

• Total Number of retweet count = 63,000
• Maximum number of retweet count = 4,118
• Average of retweet count = 210.79

Favorite Count by Tweet:
A pie chart is used to showcase the number of favorites for a particular tweet. The different colors in the pie chart represent different tweets in the dataset. We then find the total number of favorites, the maximum number of favorites and the average of total number of favorites.

• Total Number of favorite_count = 106
• Maximum number of favorite_count = 4
• Average of retweet_count = 0.35

**User Followers:**
We make a gauge chart to find the sum of user followers, the maximum number of followers for a particular user and the average of user followers.
• Sum of User followers = 17,66,355 or approx. 2M
• Max of user_followers_count = 270K
• Average of user_followers_count = 5.89K

5.3 **Overall Visualization of Data**

**Comparing the performance of Random Forest Classifier and Logistic Regression:** It is observed that Logistic regression yields a better accuracy i.e., 98.88% when compared to Random Forest Classifier which gives us an accuracy of 48.07% which is a difference of almost 50%. This is because we have a “Categorical variable dataset” and the number of variables that cannot be controlled at the production level but can be controlled at the analysis level is less than the number of explanatory variables which are independent, here sentiment value of tweets (0,1). Random forest has a higher true and false positive rate as the number of explanatory variables increases in a dataset.

**Comparing the performance and user friendly nature of Power BI and Tableau:** The comparison of the two data analytics and visualization tools is done on the basis of the following factors:

**Price:** If you are on the lookout of an affordable option, Power BI is your answer. It is free, easy to download and you only need a Microsoft account to complete the process. On the other hand, Tableau is not free but offers a free trial (1 year) for teachers and students.

6. **Future Work**
Future work involves collaborating with "Spark NLP," an open-source library for the processing of natural languages, developed on top of Apache Spark and Spark ML. Integrating with ML Pipelines offers a simple API. The Spark NLP library, which contains Scala and Python APIs for use by Spark, is written in Scala. It has no reliance on any other library of NLP or ML. The library includes the ability to train, modify and store models as a native extension of the Spark ML API so that they can run on a cluster or other machines or save for later.

7. **References**
[1]. Nasrin Irshad Hussain & Pranjal Saikia, Big Data presentation, *Kaziranga University*, Assam, 2014.
[2]. Apache Spark Introduction available at https://www.tutorialspoint.com/apache_spark/apache_spark_introduction.html, 2016
[3]. Survo Banerjee, Apache Spark Overview available at https://www.kdnuggets.com/2018/07/introduction-apache-spark.html, 2016
[4]. PySpark Introduction available at https://www.tutorialspoint.com/pyspark/pyspark_introduction.html, 2015
[5]. Mike Driscoll, Jupyter Notebook Introduction available at https://realpython.com/jupyter-notebook-introduction/, 2014
[6]. Jason Ridgen, Tweepy: a Python library for Twitter API available at https://medium.com/@jasonrigden/tweepy-a-python-library-for-the-twitter-api-9d0537dcebd4, 2018
[7]. Power BI Introduction available at https://www.tutorialspoint.com/power_bi/power_bi_introduction.htm, 2016
[8]. Sales, Tableau overview and Differentiation with MS Excel available at https://www.newgenapps.com/blog/what-is-tableau-overview-application-tableau-vs-excel, 2017
[9]. James Lani, Understanding Logistic Regression available at https://www.statisticssolutions.com/what-is-logistic-regression/, 2015
[10]. Will Koehrsen, An Implementation and Explanation of Random Forest in Python available at https://towardsdatascience.com/understanding-random-forest-58381e0602d2 , 2018
[11]. Arora, Amandeep Singh, Linesh Raja, and Barkha Bahl. "Data centric security approach: A way to achieve security & privacy in cloud computing." Proceedings of 3rd International Conference on Internet of Things and Connected Technologies (ICIoTCT). 2018.
[12]. Abdul Ghaffar Shoro & Tariq Rahim Soomro, Big Data Analysis: Ap Spark Perspective, Global Journal of Computer Science and Technology: C Software & Data Engineering, 2015
[13]. Matei Zaharia, Reynold S. Xin, Patrick Wendell, Tathagata Das, Michael Armbrust, Ankur Dave, Xiang, Josh, Shivaram Venkataraman, Michael J. Franklin, Ali Ghodsi, Joseph Gonzalez, Scott Shenker and Ion Stoica, Apache Spark: A Unified Engine for Big Data Processing, Communications of the ACM, November 2016.