Exploratory Big Data Analysis on Twitter Data - PokémonGO

Umaraj Potla
Computer Science
Georgia State University
Atlanta, Georgia
upotla1@student.gsu.edu

Shivani Bhoite
Comupter Science
Georgia State University
Atlanta, Georgia
sbhoite1@student.gsu.edu

Anil Ravuru
Computer Science
Georgia State University
Atlanta, Georgia
aravuru1@student.gsu.edu

Anit Gupta
Computer Science
Georgia State University
Atlanta, Georgia
agupta33@student.gsu.edu

Alejandro Benitez
Computer Science
Georgia State University
Atlanta, Georgia
abenitez6@student.gsu.edu

***Abstract*—Over the years, Twitter has become one of the largest social platforms and a source for wide-ranging data about what is currently going on in the world. One such trend can be seen from July 2016 until today on the most loved game PokémonGO. The purpose of our study will be to analyze the perceptions of Pokémon Go users who tweeted about the game on Twitter. Insights and visualizations would be brought based on location, hashtags, sentiment analysis, Lineal and bi-variate classification of the tweets and predictive modelling using advanced machine learning and Data mining techniques.**

I. DATA PREPROCESSING

To analyse any data, the first thing we should be doing is processing that data. The preprocessing stage includes structuring the data, filtering the useful data and extracting the useful features in the data for our analysis. As we are dealing with huge data, We should do the above-mentioned steps efficiently. As we picked PokemonGO as our topic of interest, we looked for words like 'Pokémon', 'Pokécoins' in the text field of the tweet data. Briefly, a tweet data contains the information of the tweet including time at which it was tweeted, Its source, detailed information of the user who tweeted it, location information from where it was tweeted, complete original tweet information if it is a retweet, information about all the media embedded in it and some extra information like the language, sensitivity of the tweet and number of users who liked or retweeted the tweet. Compositely there are around 390 fields embedded in a single tweet. Certainly, we do not need all this information for our analysis. Selectively we picked 19 fields to extract interesting results from our analysis.

Twitter is a worldwide platform where people from many countries tweet their opinions in different languages. Based on the language field present in the tweet, we translated all the tweets that are not in English using google translate API [2]. Although $GoogleTrans$ is free, It has limitations like blocking the IP address for repeated calls. It was quite challenging to bypass the limitations to use the API.

II. PERFORMANCE COMPARISON

We were very curious to compare the speeds of python and Spark. So we performed the above preprocessing steps using python and Spark. As it would be unfair if we compare a sequential process with Spark's inbuilt multithreading, We implemented parallel processes [6] in python to compare the speeds.

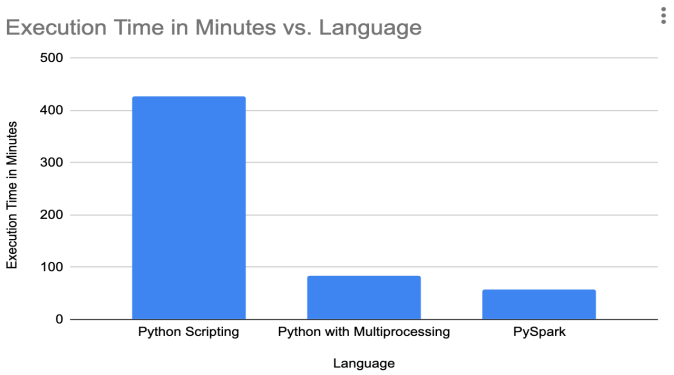

Fig. 1. Speed comparisons of different languages

System Specifications	
Python	16GB RAM Linux Machine
Python Multiprocess	8 Core, 16GB RAM Linux Machine
Spark	3 Worker nodes, 48 Core in HDInsights

From the results, we can evidently say that Spark is much faster when compared to Python. Unexpectedly, Python with multithreading performed with significant pace. Overall, Python processed the data at 14Mb/sec, Python with 16 parallel threads processed with 73Mb/sec and Spark with habove mentioned specifications processed with 105Mb/sec.

III. WHY POKÉMON GO?

It is important to know why did we choose to work on this topic. We found out it was one of the major mobile application releases in 2016. It revolutionized the mobile gaming industry by bringing augmented reality in it simplest form. We want to analyze how are people reacting to it on social media. So, the only way to compare the popularity of a new app is to compare its popularity with other applications that have already gained significant popularity. Hence in order to do that we collected the list of some of the major applications that were used in 2016. After handpicking these applications, we finalized applications such as Twitter, Amazon, Snapchat, Facebook and Tinder.

A. Analysis specific preprocessing steps

We are using translated preprocessing file for this analysis. But we came across many tweets which were still not translated by the google translation API that we are using, So in order alleviate that issue entirely we are taking the help of regular expressions where we are filtering Emojis, non-English characters and any other character which is in not in proper encoding in the tweets. In the tweets we also found that there were some situations where users where tweeting about the poke coins, Pokémon sun and moon which is similar game but for Nintendo switch, some people also write pokmon so we have considered all those edge cases as well. Similarly, other mobile application names were also considered while taking into consideration the final count.

B. Use of spark – data frames and functions over Pandas

Since we were working on a project for Big Data, we made sure that our program is scalable. We even compared the runtime analysis for pandas data frames and spark data frames. So, all the operation that we did were in spark. Major operations involved were using of regex replace, where, isin, sort, groupby etc. Hence ensuring the scalability of the code to even bigger data.

C. Plotting libraries – Pygal

We wanted to implement intuitive graphs. There are multiple libraries which can be used to do plot intuitive graphs such as plotly, bokeh and seaborn etc. But Pygal has some features which makes it stand out when compared to other libraries. It enables easy integration with web applications and enables the use of java scripting on the graphs that we have developed [4]. Pygal is also highly customizable and yet also extremely simplistic which is a very rare combination. The only drawback is for Pygal there is no integration with spark. Pygal works on either pandas data frames, series and other python data types only. So, we need to convert our final analysis from spark data frames to pandas data frames.

D. Results

We were able to get the final counts after getting to complete the preprocessing. So as explained in the introduction we first compared the tweets for the different applications, but we found that Pokémon Go was ranked fourth with the listed applications. The reason was all the other applications that we were considering were had a bigger audience and customer base. On the contrary Pokémon go app was being used by the people who are interested in playing mobile games. So, we did compared with other mobile gaming applications and we found out that in year 2016 Pokémon GO was the most tweeted application.

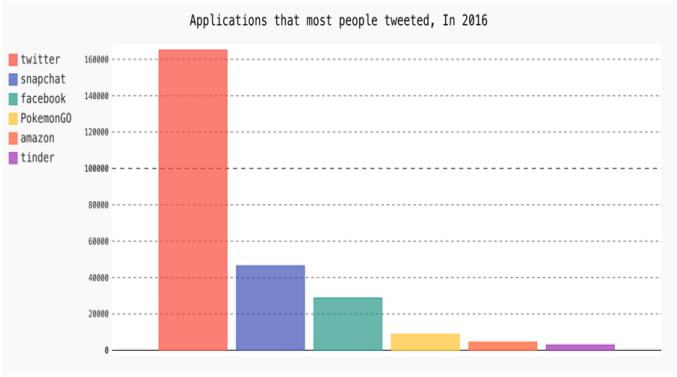

Fig. 2. Applications that most people tweeted, In 2016

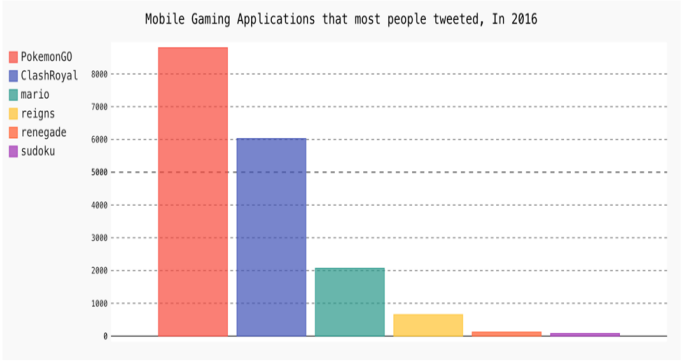

Fig. 3. Mobile Gaming Applications that most people tweeted,In 2016

E. Conclusions

We can vividly see the huge difference between people's reactions towards other games and towards Pokémon Go. The primary goal of the analysis was to justify why we choose this topic. Hence, we can see it was one of the major milestones that was achieved by Nintendo that year and analyzing people reactions on social media platforms such as twitter is worth the time.

IV. ANALYZING POKÉMON GO POPULARITY ACROSS THE WORLD

The popularity of this app has been global. People from different countries have been tweeting about this game in 2016. On the same hand there were some other countries as well which were not that profound for this game. So in order to figure out exactly the countries which were involved in

tweeting most about this game on social media. To find out this we took advantage of the coordinates tag that was within the user info. Once we had all the coordinates, we used python library Geopy to get the countries from those coordinates and then we used base map to analyze them on the world map.

A. Analysis specific preprocessing steps

Since we received csv from the initial preprocessing had the user info and all of its tags in a single column of a data frame, we first needed to flatten the user info and make another data frame. This could have one way but since we required only coordinates, we used regular expression to get the coordinates form the user info. Since there are two type of coordinates present in the data point type and polygon type. We are only considering point type coordinates as of now. Also, we needed to set the coordinates in the correct format so as to get the data from the Geopy API call. We needed to convert the coordinates to six digits after the decimal for example 16.836747 is decoded by the API as 16° 50' 12.2892" N. If same format is not passed the API returns error.

B. GeoPy – Countries from the coordinated

There are two ways to get the country from the coordinates. The first one would be to get it from the country that is in the Place tag in twitter data but the issue is we already had very less number of coordinates and then to filter them on the basis of country would be a bad idea. The other way is to have the countries from the coordinated using some python library. So we convert our spark data frame to pandas data frame and pass the coordinated to Geopy library. Geopy makes it simple for the python developers to locate the cities, countries, coordinates of addresses, and landmarks across the globe using third party geocoders and other data sources. [1] For our analysis we are using Nominatim open street map search.

geolocator = Nominatim(user$_agent = "AnitGeoCode")$

We also need to increase the timeout limit since we are sending many coordinates. We can increase the timeout by using below command

geopy.geocoders.options.default$_timeout = 1000000$

C. Results

The results that we found out included a count of tweets per country into a spark data frame. Since mostly code is written in spark only. Results were also taken out into CSV so that we can plot using Tableau for interactive analysis. Also, on the same hand we used Base map to plot the points on the world map in the python as well.

D. Conclusion

In conclusion we can clearly see that the countries that were tweeting most about this game were USA and Latin American countries also there was very minimal participation from the Asian countries. Also important thing to note is that the there were no coordinates from Russia which means people from Russia were not posting on social media about Pokemon Go.

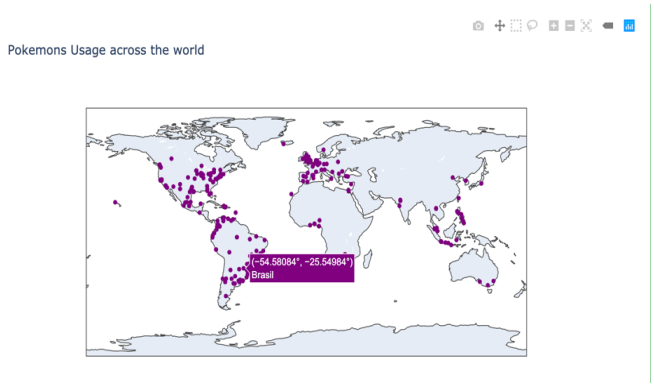

Fig. 4. Pokemon usage across the world

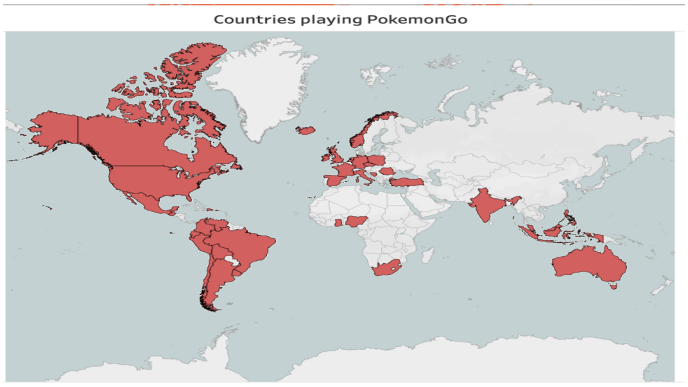

Fig. 5. Countries tweeting about Pokemon go

V. Sentiment Analysis on PokemonGo Tweets and its popularity

Pokemon Go game received both positive and negative feedback-related to the implementation using AR in this field, its vivid use, and technical glitches that the application had. Users tweeted related to PokemonGO with their opinions towards the game. As part of sentiment analysis, we have analyzed the user's take over the game. Although a tweet is a simple text with limited characters, people express their opinions precisely through hashtags. So based on that valuable text and hashtags, we would like to categorize the tweets into positive, neutral, and negative. While computing the overall polarity of the game, we will be considering the level of positivity and negativity of the tweets. We have used different visualization techniques in Tableau to observe the presence of different emotions and their corresponding valence in texts from tweets towards PokemonGO.

A. Use of NLTK - VADER library

For performing the sentiment analysis, we have used the [16] NLTK - VADER (Valence Aware Dictionary and sEntiment Reasoner) library. [13] VADER is a lexicon-based and rule-based tool for sentiment analysis. This VADER tool is precisely harmonized with sentiments expressed in social media platforms. [17] The VADER does not need any

training data. The VADER is constructed from a generalizable, valence-based, human-curated conventional emotion lexicon.

B. Application of Sentiment Analysis

Data pre-processing was performed to the text before the application of the Sentiment Analysis. Stopwords, punctuation, and empty spaces were removed [7]. Then the text underwent the process of chunking to label each word with its respective parts of speech. [8] In this context, only Noun Phrases of the words were taken into consideration. [8] The process of consideration of particular Noun Phrases is called chunking [8] [12]. These chunks are considered as input to the NLTK - VADER method. [15] After the Sentiment Analysis, the compound score for each observation is considered as the final parity score. The compound score is the aggregated value or compounded value of the positive,neutral, and negative parity of the observation. [8]

C. Conclusion

After analysis, most of the results of the comments to neutral, i.e., the compound score was zero. The comments having a parity score of less than 0 are considered as negative. The parity score for more than 0 is considered positive. Higher is the parity higher is the positivity of the comment. So after the Sentiment Analysis, it is concluded that the majority reaction of the population for the Pokemon Go was neutral or positive.

VI. Latent Dirichlet Allocation (LDA)

Topic modeling is a method for unsupervised classification of documents or text to find natural groupings called “topics” even when the characteristics of each topic is unknown [18]. Topics are created by using TF-IDF vector representations of words and clustered by their appearance in particular groups of documents. The main approach for this is to select k number of topics for the algorithm, apply the algorithm to twitter data that mention Pokemon, and analyze the most frequent words used in each topic to find a real-life meaning for each topic. This analysis will be using the LDA model from ML lib library builtin to Pyspark [5] demonstrated here [11].

A. Choosing Number of Topics

Choosing the number of topics affects the granularity of the topics as well as the overlap between key words common to multiple topics. To find the number of topics used for this analysis, perplexity score was evaluated for 2 to 200 topics run on a subset of the data to shorten training time over the iterations. Results of these experiments are shown below.

Perplexity is a measure of overlap between topics, as so the lower the perplexity score the better the separation between topics. Then two topics would be optimal for greatest separation, however, two topics would be generalizing too much for the intended purpose of running this model, which is to better understand the different context for which Pokemon and PokemonGo are being frequently tweeted about in the given month of twitter data. The number of topics was increased

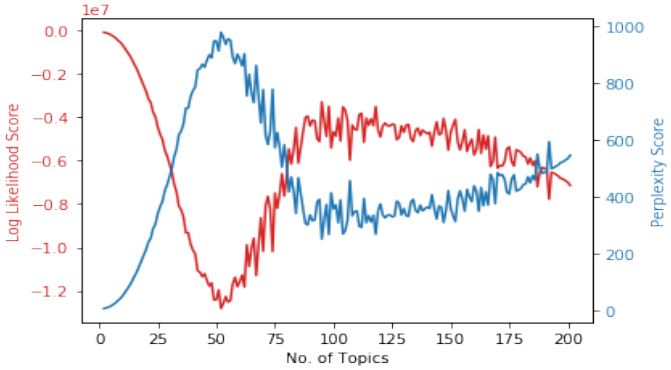

Fig. 6. LDA Training Perplexity vs No. of Topics

until overlap could be visually be seen in topic distribution chart and then chosen to be one less(4 topics).

B. Results and Conclusions

The topic distribution chart for lda using 4 topics contains the results of the analysis and is attached as a html file [3]. Topics are represented by salient key words prevalent in each single topic. These words may be found in other topics but should be prominent in the respective topic.

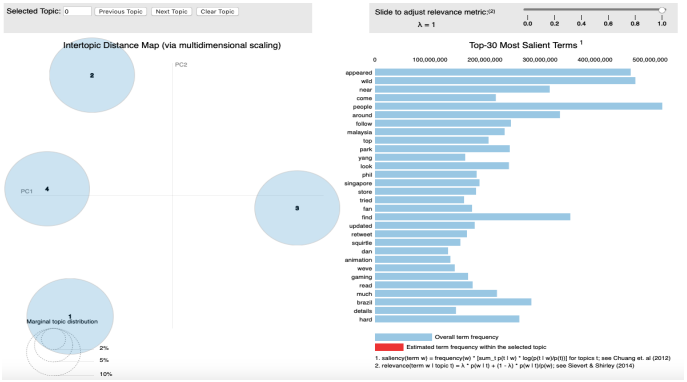

Fig. 7. LDA Topic Distribution

Topics Top Ten Salient Words Per Topic:

Topic 1

Wild, Appeared, Catch, Around, Near, Tube, Video, Free, Park, Playing

Topic 2

Via, News First, Malaysia, Video, Playing, Play, Amp, Get, Poke

Topic 3

Team, People, News, App, Like, You’re, Come, New, Game, Look

Topic 4

Play, New, Tube, Playing, Find, App, Follow, Top, Take, Video

While not obvious from the words themselves, we can interpret the topics' meanings from how the words would probably be used in tweets. Topic 1 are tweets referring to playing the game and tweeting out what Pokemon they have found and their location while playing. Pokecoins, the in-game currency, is also listed in this topic (not top ten) lending to the idea that this topic are people playing the game. Topic 2 are Pokemon news videos being spread through tweets popularized in Malaysia. Topic 3 are people talking about the novelty of the new game/app potentially in news articles. Topic 4 are tweets related to social media as it contains words like 'follow', 'retweet', 'giveaway' and 'trends'. These are words that social media celebrities might say to appeal to their fans and fans of the game.

C. Further Research

This analysis chose 4 topics to develop an understanding of Pokemon tweets. This is not necessarily the best value for number of topics, but we did find meaningful results. Underlying subtopics may arise by increasing the number Topics used in the algorithm and allowing for more overlap of the topics. Dynamics topic modeling may also prove useful in understanding the context of Pokemon tweets over time, however this requires data across a significant time period to monitor changes in topics.

VII. Identifying Most Frequent And Rarest Pokemon Characters Tweeted

In the Pokemon Go game, there are approximately 810 Pokemon characters that can be caught by the players. However, the rarity of these Pokemon varies and the availability of Pokemon is also dependent on the geography and climate of a particular region. Every location might not have all the Pokemon characters available in the game. Users usually tweet about the Pokemon character they caught and the scores along with screenshot images of the Pokemon. In this study, we identified the top 10 most frequent Pokemon characters and 10 most rarest Pokemon characters tweeted about from the tweets text. This type of analysis could be useful to know the popularity of the Pokemon characters being talked about by people around the world over Twitter.

A. Methodology

List of Pokemon characters available on Pokemon Go game [10] was used to identify the number of tweets each Pokemon character was tweeted about by the users. The top ten and the last ten frequent characters were used to visualize their counts.

B. Results And Conclusions

Fig. 7 shows the top ten most frequent Pokemon characters tweeted in Twitter in the year 2016, October. 'Pikachu' being the highest frequent Pokemon character, it can be implied that Pikachu was caught the most in the Pokemon Go game and gain most popularity by people around the world, followed by 'Brionne', 'Torracat', 'Rowlet' and 'Squirtle' as the top five most popular Pokemon characters.

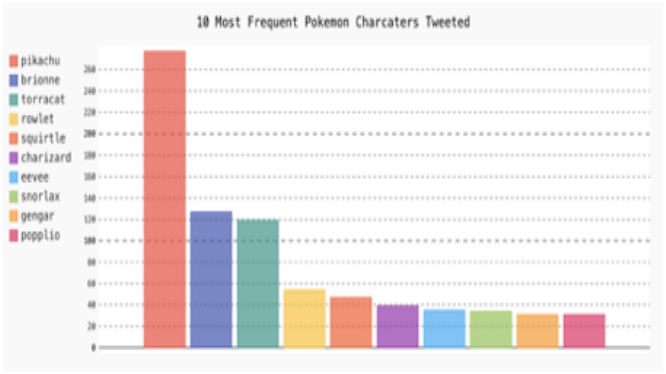

Fig. 8. Most Frequent Pokemon Characters Tweeted

Fig. 8 shows the top ten most rarest Pokemon characters tweeted in Twitter in the year 2016, October. 'Duskull', 'Zangoose', 'Hoopa', 'Zekron' and 'Wailord' are the most rarest Pokemon characters. It can be implied that these characters were caught the rarest in the Pokemon Go game and gained least popularity by people around the world.

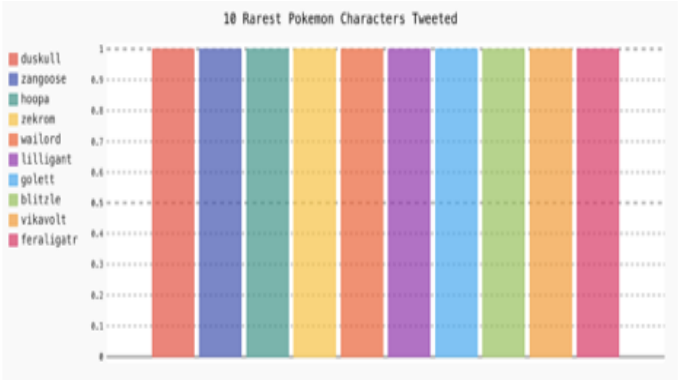

Fig. 9. Most Rarest Pokemon Characters Tweeted

VIII. Time Series Analysis

Time series analysis and prediction modeling are one of the research areas which has attracted attention over the last few decades. The time series analysis technique deals with time-series data or trend analysis. Time series data means that data is in a series of particular periods or intervals [14]. Aim of this modeling technique is studying the past observations of a time series and develop an appropriate model to predict the inherent structure of the series. The fitted model is then used to predict the future values for the series, i.e., to make forecasts. Time series analysis is also known as a method of predicting the future by understanding the past [19].

A. Seasonal Autoregressive Integrated Moving Average (SARIMAX) Model

SARIMAX models are a type of general time series model and are used to analyze and forecast data that have an additional seasonal component. Values for p, d, and q are

derived to make the time series stationary. A series which is made stationary has a constant mean and variance. Below is a general explanation of a SARIMAX model.

$$\text{ARIMA}\ \underbrace{(p,d,q)}_{\text{Non-seasonal part of the model}}\ \underbrace{(P,D,Q)_m}_{\text{Seasonal part of the model}}$$

Fig. 10. SARIMAX

p value of AR model implies the output variable at time t is sum of past values. *d* value of the I part is the number of non-seasonal differences applied to the series which is needed to make it stationary. The SARIMAX model takes past values as training input, and gets the differnce from current value. *q* value from the MA model is a moving average of previous error terms. The Seasonal portion *(P, D, Q)m* has the same structure as the non-seasonal parts [9].

B. Training The Model

To train the SARIMAX Model, tweets with created at dates between 01st October, 2016 to 25th October, 2016 was considered which is 2/3rd of the data. To test the model, tweets between 26th October to 31st October was considered. order = (4,1,1) and seasonal$_order$ = $(2, 1, 1, 24) was used to get seasonal order of 24 hours$.

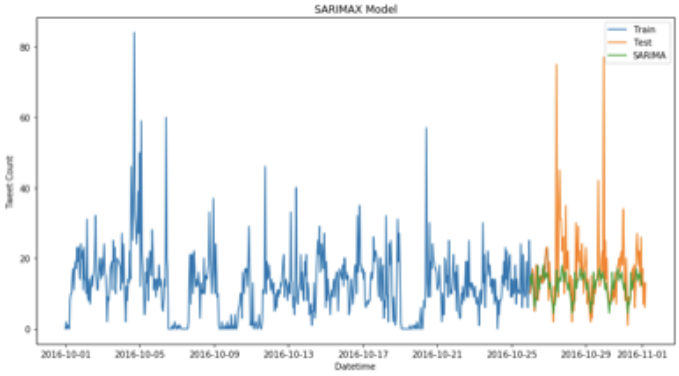

Fig. 11. SARIMAX Model

C. Results

Fig. 12 is the SARIMAX model summary showing that model was built as SARIMAX (4, 1, 1) x(2, 1, 1, 24). AIC and BIC values are 4173 and 4213 respectively. Standard error values ranges from 0.032 to 0.034 for AR model and 0.015 for MA model. Fig. 11 shows the SARIMAX model fit on Twitter data in time-series graph model.

Statespace Model Results

Dep. Variable:	datetime_new	No. Observations:	624
Model:	SARIMAX(4, 1, 1)x(2, 1, 1, 24)	Log Likelihood	-2077.756
Date:	Wed, 20 Nov 2019	AIC	4173.511
Time:	02:17:05	BIC	4213.069
Sample:	10-01-2016	HQIC	4188.911
	- 10-26-2016		
Covariance Type:	opg		

	coef	std err	z	P>|z|	[0.025	0.975]
ar.L1	0.2854	0.034	8.367	0.000	0.219	0.352
ar.L2	0.1958	0.034	5.822	0.000	0.130	0.262
ar.L3	0.0768	0.033	2.294	0.022	0.011	0.142
ar.L4	0.1467	0.032	4.626	0.000	0.085	0.209
ma.L1	-0.9834	0.015	-63.976	0.000	-1.013	-0.953
ar.S.L24	-0.0328	0.049	-0.670	0.503	-0.129	0.063
ar.S.L48	-0.0898	0.048	-1.855	0.064	-0.185	0.005
ma.S.L24	-0.9962	0.975	-1.021	0.307	-2.908	0.916
sigma2	52.2633	49.668	1.052	0.293	-45.085	149.612

Ljung-Box (Q):	26.95	Jarque-Bera (JB):	1692.40
Prob(Q):	0.94	Prob(JB):	0.00
Heteroskedasticity (H):	0.55	Skew:	1.57
Prob(H) (two-sided):	0.00	Kurtosis:	10.61

Warnings:
[1] Covariance matrix calculated using the outer product of gradients (complex-step).

Fig. 12. SARIMAX Model Summary

With the above fit model, predictions was made for dates 26th October, 2016 to 31st October 2016 for each hour to forecast the frequency of Pokemon tweets. Fig. 13 shows the prediction graph.

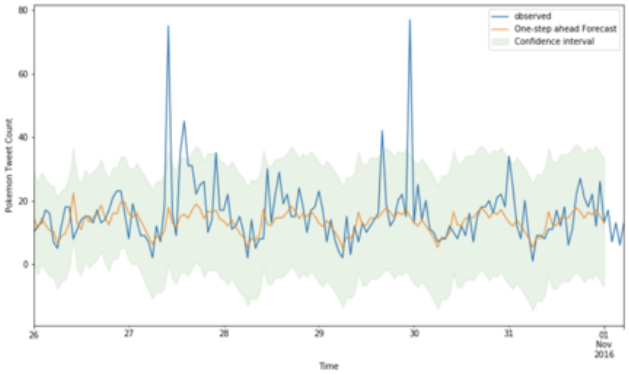

Fig. 13. SARIMAX Prediction

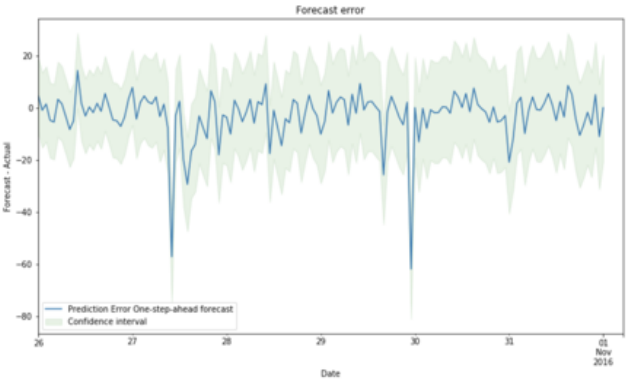

Fig. 14. SARIMAX Error

Fig. 15 shows Error rate of the SARIMAX model prediction. Error rate is calculated as the difference of prediction mean and the actual test values. The confidence interval, in light green color, is pretty high.

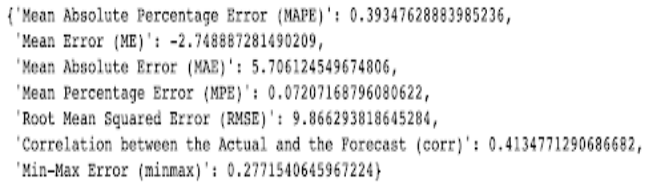

Fig. 15. SARIMAX Error Rate

Mean Error rates of the SARIMAX model is as shown below. Mean Absolute Error was 5.7 and Correlation between actual and forecast value is 41 percent. Overall, mean percentage error of the model is 7.2 percent which gives an accuracy of 93 percent.

D. Further Research

In this study, SARIMAX model was built without smoothing. The current SARIMAX model prediction has quite high confidence interval. Different levels of seasonal order and AR, MA models can be built to identify the best prediction model having least AIC and BIC scores.

REFERENCES

[1] Geopy . pypi. Accessed: 2019-11-8.
[2] Google translate api. https://pypi.org/project/googletrans/. Accessed: 2019-12-10.
[3] Lda visualization. https://stackoverflow.com/questions/41819761/pyldavis-visualization-of-pyspark-generated-lda-model. Accessed: 2019-11-10.
[4] Pygal svg graphs with flask tutorial. Accessed: 2019-11-8.
[5] Spark clustering. https://spark.apache.org/docs/latest/ml-clustering.htmllatent-dirichlet-allocation-lda. Accessed: 2019-11-2.
[6] Threading interface. https://docs.python.org/2/library/multiprocessing.html. Accessed: 2019-12-10.
[7] Nagesh Singh Chauhan. Natural language processing in apache spark using nltk (part 1/2), February 2017. Accessed: 2019-11-8.
[8] Nagesh Singh Chauhan. Natural language processing in apache spark using nltk (part 2/2), February 2017. Accessed: 2019-11-8.
[9] Robert R.F. DeFilippi. Sarima modelling for car sharing — basic data pipelines — applications with python pt. 1, May 2018. Accessed: 2019-12-06.
[10] the free encyclopedia From Wikipedia. Pokémon go, November 2019. Accessed: 2019-11-10.
[11] Soumya Ghosh. Topic modelling with latent dirichlet allocation (lda) in pyspark. https://medium.com/@connectwithghosh/topic-modelling-with-latent-dirichlet-allocation-lda-in-pyspark-2cb3ebd5678e, March 2018. Accessed: 2019-11-2.
[12] Guru99. Pos (part-of-speech) tagging chunking with nltk, February 2017. Accessed: 2019-11-8.
[13] K. M. Badran H. Elzayady and G. I. Salama. Sentiment analysis on twitter data using apache spark framework, November 2018. Accessed: 2019-11-8.
[14] PhD James Lani. Time series analysis, December 2019. Accessed: 2019-12-06.
[15] Ricky Kim. Sentiment analysis with pyspark, 2018.
[16] Nikolaos Nodarakis. Large scale sentiment analysis on twitter with spark, November 2016. Accessed: 2019-11-8.
[17] C.J. Hutto Eric Gilbert Georgia Institute of Technology. Vader: A parsimonious rule-based model for sentiment analysis of social media text, November 2014. Accessed: 2019-10-5.
[18] Julia Silge and David Robinson. Text mining with r. https://www.tidytextmining.com/topicmodeling.html, November 2019. Accessed: 2019-11-4.
[19] P. Sanguanbhoki T. Raicharoen, C. Lursinsap. Application of critical support vector machine to time series prediction. In *Circuits and Systems, 2003*, May 2003.