Covid19 Retweet prediction

Introduction

Diffusion is the process by which information is spread from one place to another through interactions. And, during time of crisis (like Covid19 Pandemic), it is fundamental to try to understand user's behavior and interest better.

In this specific case, we are delimiting the problem to a social network site called "Twitter" and "Retweet" as a popular information diffusion mechanism.

Related work

Social network analysis and diffusion of information are two topics that have been studied for a long time. However, this specific challenge was launched in October 2020 (Covid19 context). Thus, I think is better to present recent studies.

Title	Year	URL	Summary

Prediction of Likes and Retweets Using Text Information Retrieval	2020	https://www.sciencedirect.com/science/article/pii/S1877050920304129	Twitter is one of the major social media platforms today to study human behaviors by analyzing their interactions. To ensure popularity of the tweet, the focus should be on the content of the tweet that results in numerous followings of that message with sufficient number of likes and retweets. The high quality of tweets, increases the online reputation of the users who post it. If a user can get the prediction of likes and retweets on his text before posting it on the internet, it would improve the popularity of the tweet from information sharing perspective. In this paper we employed different machine learning classifiers like SVM, Naïve Bayes, Logistic Regression, Random Forest, and Neural Network, on top of two different text processing approaches used in NLP (natural language processing), namely bag-of-words (TFIDF) and word embeddings (Doc2Vec), to check how many likes and retweets can a tweet generate. The results obtained indicate that all the models performed 10-15% better with the bag-of-word technique.

Understanding Information Spreading Mechanisms During COVID-19 Pandemic by Analyzing the Impact of Tweet Text and User Features for Retweet Prediction	2021	https://arxiv.org/abs/2106.07344	COVID-19 has affected the world economy and the daily life routine of almost everyone. It has been a hot topic on social media platforms such as Twitter, Facebook, etc. These social media platforms enable users to share information with other users who can reshare this information, thus causing this information to spread. Twitter's retweet functionality allows users to share the existing content with other users without altering the original content. Analysis of social media platforms can help in detecting emergencies during pandemics that lead to taking preventive measures. One such type of analysis is predicting the number of retweets for a given COVID-19 related tweet. Recently, CIKM organized a retweet prediction challenge for COVID-19 tweets focusing on using numeric features only. However, our hypothesis is, tweet text may play a vital role in an accurate retweet prediction. In this paper, we combine numeric and text features for COVID-19 related retweet predictions. For this purpose, we propose two CNN and RNN based models and evaluate the performance of these models on a publicly available TweetsCOV19 dataset using seven different evaluation metrics. Our evaluation results show that combining tweet text with numeric features improves the performance of retweet prediction significantly.
CIKM AnalytiCup 2020: COVID-19 Retweet Prediction withPersonalized Attention	2020	http://ceur-ws.org/Vol-2881/paper1.pdf	This paper describes the first place winning solution for the CIKM AnalytiCup 2020 COVID-19 retweet prediction challenge. The objective of the challenge is to predict the popularity of COVID-19related tweets in terms of the number of retweets, and the submitted solutions of the challenge are ranked based on Mean Squared Logarithmic Error(MSLE) on the leaderboard. The proposed deep learning model to predict retweet counts uses minimal hand-engineered features and learns to predict retweet count based on a personalized attention mechanism. As a tweet keyword may have different informativeness for different users, the personalized attention mechanism helps the deep learning model to weigh the importance of tweet keywords based on a user's interest to retweet. Additional techniques such as adding external data sets to training and pseudo-labeling are also experimented with to

			further improve the MSLE score. The final solution comprises of an ensemble of different personalized attention-based deep learning models, and the source code for the solution can be found at https://github.com/vinayakaraj-t/CIKM2020.
Word and Graph Embeddings for COVID-19 Retweet Prediction	2020	http://ceur-ws.org/Vol-2881/paper2.pdf	In this paper, we present our solution for COVID-19 retweet pre-diction challenge. The proposed approach consists of feature engineering and modeling. For feature engineering, we leverage both handcrafted and unsupervised learning features. As the provided data set is large, we implement auto-encoding algorithms to reduce feature dimension. To develop predictive models, we utilize ensemble learning and deep learning algorithms. We then combine these models to generate the final blended model. Moreover, to stabilize the predictions, we also apply bagging as well as down sampling techniques to remove the tweets where number of retweets equals to zero. Our solution is ranked first on the public test set and second on the private test set.

Using sentiment analysis to predict opinion inversion in Tweets of political communication	2021	https://www.nature.com/articles/s41598-021-86510-w	Social media networks have become an essential tool for sharing information in political discourse. Recent studies examining opinion diffusion have highlighted that some users may invert a message's content before disseminating it, propagating a contrasting view relative to that of the original author. Using politically-oriented discourse related to Israel with focus on the Israeli–Palestinian conflict, we explored this Opinion Inversion (O.I.) phenomenon. From a corpus of approximately 716,000 relevant Tweets, we identified 7147 Source–Quote pairs. These Source–Quote pairs accounted for 69% of the total volume of the corpus. Using a Random Forest model based on the Natural Language Processing features of the Source text and user attributes, we could predict whether a Source will undergo O.I. upon retweet with an ROC-AUC of 0.83. We found that roughly 80% of the factors that explain O.I. are associated with the original message's sentiment towards the conflict. In addition, we identified pairs comprised of Quotes related to the domain while their Sources were unrelated to the domain. These Quotes, which accounted for 14% of the Source–Quote pairs, maintained similar sentiment levels as the Source. Our case study underscores that O.I. plays an important role in political communication on social media. Nevertheless, O.I. can be predicted in advance using simple artificial intelligence tools and that prediction might be used to optimize content propagation.

Methods

To achieve the objectives of the analysis, it is always necessary to start with an understanding of the context and the available data.

Regarding the data, we can create two segments: categorical and quantitative features (in addition to the label which is a quantitative value). In this same way, we can divide the feature engineering process in two groups: quantitative and categorical.

Quantitative feature engineering

Also, within this group we have made the subdivision as follows:

- User-based

    - count_td: Count of tweets per day
    - count_tdu: Count of tweets per day per user
    - Both “count_td” and “count_tdu” were scaled using MinMaxScaler: The rescaled value for a feature E is calculated as, Rescaled(ei)=ei−EminEmax−Emin∗(max−min)+min(1). For the case Emax==Emin, Rescaled(ei)=0.5∗(max+min)
- Time-based
    - Weekend: Specify if the event took place on a weekend or not
    - am_of_day: specify if the event took place “ante meridiem”
    - pm_of_day: specify if the event took place “post meridiem”
- Content-based
    - pos_sentiment: positive evaluation of the tweet, coming from the sentiment field.
    - neg_sentiment: negative evaluation of the tweet, coming from the sentiment field.
    - number_entities: number of entities, found in the entities field (taking into account the specified delimiter).
    - number_mentions: number of mentions, found in the mentions field (taking into account the specified delimiter)
    - number_hashtags: number of hashtags, found in the hashtags field (taking into account the specified delimiter)
    - number_urls: number of urls, found in the urls field (taking into account the specified delimiter)
    - sentiment_overall_trinary_Score: the positive polarity score (from 2 to 5), the negative polarity score (from -2 to -5) and scores (1 and -1) are considered to be neutral. So, the overall trinary score is as follow: the overall positive (score = 1), negative (score = -1) and neutral (score = 0).

Categorical feature engineering: Not implemented

Experimentation

For this project, the experiments that can be mentioned have arisen to meet needs, a product of intuition or by applying techniques or methodologies taught in previous sessions.

- Development environment: Due to the limitations of my laptop (resources), I have been forced to experiment with a full web environment . For this, I have used databricks community edition as machine learning platform and a brand new google drive account as primary data source (databricks community edition has limitations to use S3 buckets)
- Dara ingestion: to increase the performance of operations on spark dataframes I implement the following process: first create a dataframe from original CSV file, then save the dataframe to parquet file and finally create a dataframe from parquet file (Column-oriented). So, at the end, I am querying against a Spark DataFrame based on Parquet.
- Also, for this project you have imposed the personal constrain of not converting the pyspark dataframes to pandas dataframes. So that all analysis is done with spark

- Linear regression model hyper Param: ParamGrid for Cross Validation was implemented

Results

- Training time and best results go hand in hand with the complexity of the model.
- R squared metric represents the percentage of variance explained and something surprising is that this metric was high in almost all the variations tested (which is not strange for a multiple regression). But since Spark does not have the calculation of "predicted R-squared" and "adjusted r squared", and it did not build its own method for its calculation, we cannot say with certainty if there is overfitting

Discussions

the project has many pending tasks:

- Implement another machine learning models and deep learning models, in addition to linear regression
- Categorical feature engineering
- Implement NLP

Source Code

Since I have configured a full web environment, the source code can be viewed through this url:

https://github.com/lveagithub/bts-rta-2020/blob/main/AssigmentMachineLearning/Predict_num_retweets.ipynb

Conclusions

- Since one of my project goals is to try to solve the problems only with Spark, without using libraries like Pandas, I have been able to check the following:
    - Based on my current experience, Spark lacks statistical and plotting libraries that can be found in Python. Which makes us often feel the temptation to use certain libraries such as Pandas or seaborn, which are more advanced in this regard.
    - Knowing how to work with Big Data is essential, and you must be careful about converting data structures such as Pyspark dataframe to Pandas dataframe (in my opinion it is not a good practice, although it is widely used).
    - Experimentation with big data goes hand in hand with horizontal scaling.
    -
- Machine learning model:
    - Feature engineering is really important, but the complexity of the model leads to overfitting. For that reason, in addition to regularization, it was necessary to implement

cross validation by splitting the dataset into a set of *folds* which are used as separate training and test datasets. But the drawback of this process is the considerable time involved in training the model.

- Similar projects in which this type of analytical model can be applied:
  - The Covid19 pandemic is not over, it would be interesting to apply it with data from other countries.
  - There is another pandemic called fake news, it would be interesting to apply this type of model (with the mentioned pending tasks) to determine the behavior of users and to be able to create early alerts that prevent the dissemination of invalid information.