Predicting article's shares and favorites based on title using Machine Learning

Flávio Henrique de Freitas
(Dated: September 7, 2018)

***Abstract*—Choosing a good title for an article is an important step of the writing process. The more interesting the article title seems, the higher the chance a reader will interact with the whole content. This project focus on predicting the number of shares and favorites on Twitter from FreeCodeCamp's articles based on its titles. This problem is a classification task using Supervised Learning. With data from FreeCodeCamp on Twitter and Medium, it was used machine learning methods including Support Vector Machines (SVM), Decision Trees, Gaussian Naive Bayes (GaussianNB), K-Nearest Neighbors and Logistic Regression to make the predictions. This study shows that the XXX algorithm performed better than the others reaching an accuracy of XXX.**

***Keywords*— prediction, machine learning, social media, title, performance**

I. DEFINITION

A. Project Overview

Social networks websites have become an important communication tool and source of information. The hours spent in average connected per day in the past years is up to 6 hours [1] for adults and 9 for teenagers, while 30% of this time is on social networks [2]. During a normal navigation on such platforms, users are exposed to several posts such as friends' statuses, images, news and more. With such amount of information and variety of content, the time for the user to decide to interact with the content is very small. Gitte at al. [3] suggest that we take around 50 milliseconds to make a good first impression and this has proved to be very powerful in a wide range of contexts.

Besides being a place for connecting with friends and sharing moments of the user's life, a survey has shown that social networks are also used as a source of news and information by 67% of the users [4]. Part of these posts are articles that can be read on an external website. Typically such posts show the title of the article and sometimes a small part of its content and an image.

Considering the offer of content and competition with so many interesting posts, showing a proper title for the post affects the probability that a user will check the content. This measure has a strong impact on how many readers an article will have and how much of the content will be read. Furthermore, showing the user a content they prefer (to interact) increases the user satisfaction. It is thus important to accurately estimate the interaction rate of articles based on its title.

B. Related Work

In the literature is possible to find previous studies on the area of classifying the article focused on click-baits title detection [5] [6]. Click-bait headlines normally exploits the curiosity of the reader, proving enough information to make the reader curious, but not enough to fully satisfy the curiosity. In this way, the user is forced to click on the linked content to read the whole article.

Some other studies also investigate this subject using deep learning on cross-domain sentiment analysis [7].

C. Problem Statement

When an author writes a text, it is expected that their words will influence and bring value to the readers. While writing, the title is one of the important details that needs to be taken in consideration, because this will normally be the first contact place of their work. Thus, to create a good first impression, to have more people read the article and interact with it, choosing a good title is very important.

Some of the most used platforms to spread ideas nowadays are Twitter [8] and Medium [9]. On the first one, articles are normally posted including external URLs and the title, where users can access and demonstrate satisfaction with "Favorites" or "Retweeting" (sharing) of the original post. The second one shows the full text with tags to classify the article and "Claps" (similar to Twitter's "Favorites") to show how much the users appreciate the content. A correlation between these two networks can bring us more valuable information.

The problem to be solved is a classification task using Supervised Learning: *Predict the range of favorites and shares count an article receives based on its title*

D. Evaluation Metrics

At least one evaluation metric is necessary to quantify the performance of the benchmarks and solution model. For this project, it will be used the accuracy, which is the number of correct predictions made as a ratio of all predictions made.

$$Accuracy = \frac{\text{Number of correct predictions}}{\text{Total number of predictions made}}$$

TABLE I. Sample of the data points

Title	Retweet	Favorite	Claps
ES9: JavaScript's state of the art in 2018	15	48	618
Here's another way to think about state: How to visually design state in JavaScript	10	30	2
How to understand Gradient Descent, the most popular ML algorithm	4	14	102

This metric only works well if there are similar number of samples belonging to each class. For this reason, we will divide the range of shares and favorites count in a way that respects this distribution.

II. ANALYSIS

A. Data Exploration

The data used to predict how titles will perform was gathered from the accounts of the non-profit organization FreeCodeCamp on Medium [10] and Twitter [11]. On both social platforms, it was possible to get public information about how the users interacted with the content, using as "Favorites" and "Retweets" from Twitter, and "Claps" from Medium.

Correlating the number of "Favorites" and "Retweets" from Twitter with a Medium article, is an attempt to isolate the effect of number of reached readers and number of Medium "Claps". Because the more the article is shared in different platforms, the more readers it will reach and the more Medium "Claps" it will receive. Using only the Twitter statistic, it is expected that the articles reached initially almost the same number of readers (that are the followers of the FreeCodeCamp account on Twitter), and their performance and interactions are limited to the characteristics of the tweet, for example, the title of the article, that is exactly what we want to measure.

The FreeCodeCamp account was chosen, because the idea is to limit the scope of the subject of the articles and predict better the response on a specif field. The same title can perform well in one category (e.g. Technology), but not necessarily in a different one (e.g. Culinary). Also this account posts as the Tweet content the title of the original article and the URL on Medium.

After getting the articles from FreeCodeCamp written on Medium and shared on Twitter, there is a dataset of 717 data points. Table I shows some examples of such correlation and table II explains the complete list of fields of the dataset.

TABLE II. Complete description of the dataset fields

Field	Description
Title	The content of the tweet, FreeCodeCamp normally uses the title of the article from Medium and sometimes the username of the author from Twitter
Retweet Count	How many times that tweet was "Retweeted" on Twitter
Favorite Count	How many times that tweet was marked as favorite on Twitter
Medium Claps	How many times that article was marked as favorite on Medium
Medium Categories	Which tags were used to classify the article on Medium
Created at	When the tweet was posted
URL	The website of the article on Medium

B. Exploratory Visualization

This section will explore the data visualization of the existing dataset and analyze the possible metrics that will be used to understand the solution. We will identify the relationship between each one of the features with the overall performance of the article.

1. Overall Statistic

We will analyze here the high level statistics of the articles. Try to understand how many times the articles were in average re-tweeted, clapped or marked as favorites. Also understand the average title length of the articles.

TABLE III. Overall Statistic

	Favorite	Re-tweet	Claps	Text Length
count	717.00	717.00	717.00	717.00
mean	49.50	16.50	284.18	80.56
std	45.34	15.70	273.27	22.12
min	0.00	0.00	1.00	21.00
25%	20.00	7.00	6.00	65.00
50%	34.00	11.00	238.00	79.00
75%	64.00	20.00	469.00	97.00
max	298.00	125.50	997.00	146.00

From this statistic is possible to understand the order of magnitude of our dataset. Articles normally are re-tweeted and marked as favorite around tens of times and clapped hundreds of times. It is possible to check the maximum values from all the three variables, re-tweet and favorite hundreds and clap thousand of times. From these numbers we can define what is expected from our articles and the interaction with them. The length of the

text goes from 21 to 146 characters, as expected, for a tweet content.

2. *Histogram and Box plots*

In this section we will check how the multiple features are distributed.

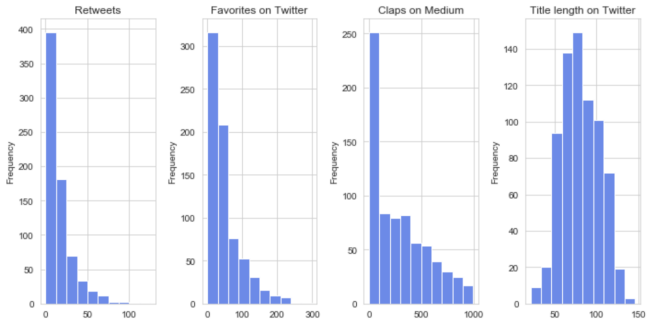

FIG. 1. Histogram

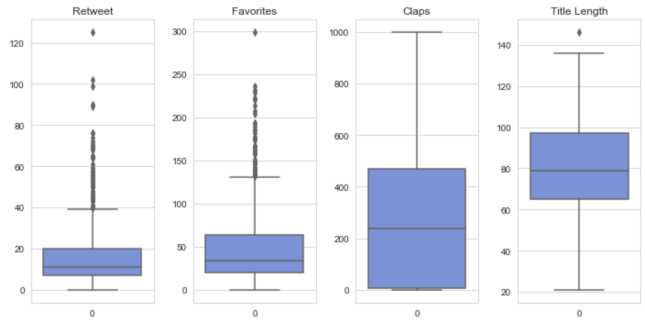

FIG. 2. Box plots

From these histograms together with the overall statistic and the box plots, we can notice that we have a Gaussian distribution for the text length and the average length is around 80 characters. Favorite, re-tweet and claps are positive-skewed, i.e. they are concentrated on the left part of the graph, meaning that a small part of the articles will over-perform about readers' interaction and biggest part of them will generate less interaction.

3. *Scatter Matrix*

Here we try to find a relationship between the multiple features that we gathered from Twitter and Medium.

We can notice for the image 3, we can notice a clear relationship between number of re-tweets and favorites. They are directed connected, it means, the more re-tweets, the more likes the article will receive and vice versa.

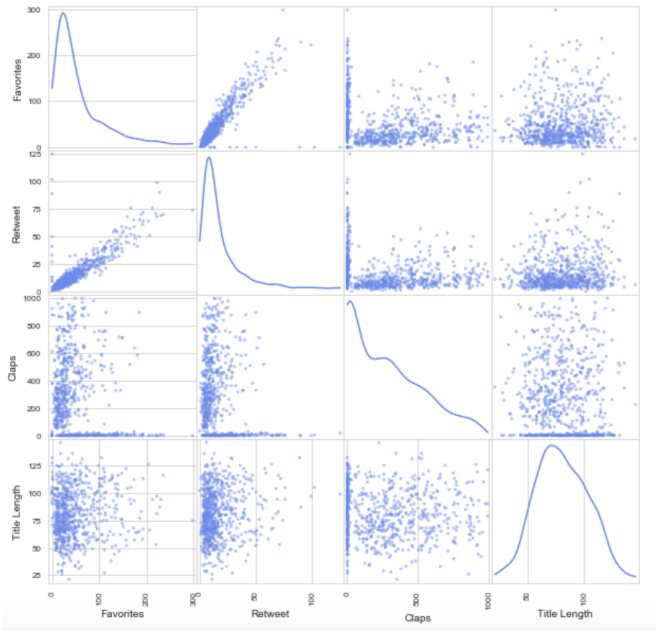

FIG. 3. Relationship between the features

4. *Title length that performed better*

Here we analyze the relationship between the length of the title with its performance. For this experiment, we just considered the 25% top performers of each feature.

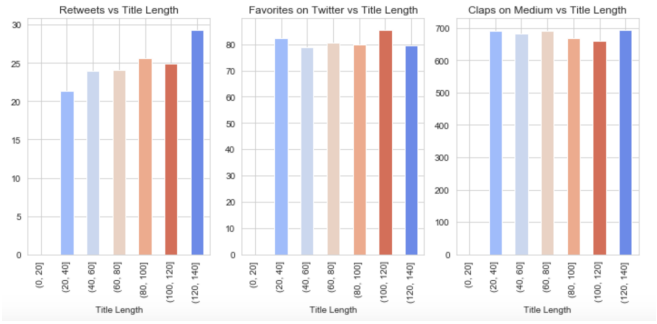

FIG. 4. Title length performance

To avoid being biased by outliers, we removed for each feature (favorites, re-tweets and claps) analysis the data points that don't fit the following formulas:

$$Outlier < Q_1 - 1.5 * IQR$$

$$Outlier > Q_3 + 1.5 * IQR$$

Where Q_1 and Q_3 are the first and third quartile and IQR is the Interquartile Range (IQR = Q_3 - Q_1).

We can notice from the graphics 4 that longer titles tend to perform a little bit better than shorter ones for

re-tweets, but for claps on Medium and favorites it seems to influence even less.

After analyzing the title length and didn't reach any conclusion, we decided to investigate the number of words in the title.

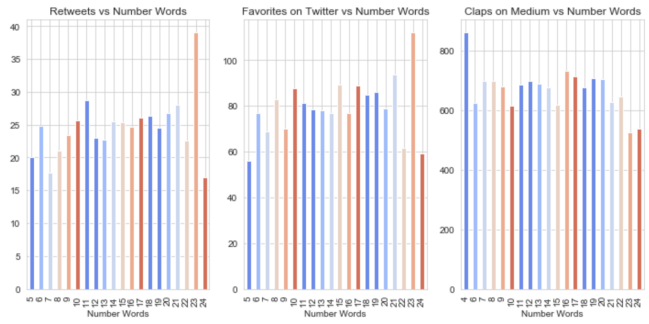

FIG. 5. Performance of the Number of words in the title

From this second experience showed on image 5, we reached the conclusion that neither number of words on the title affect considerably its performance.

5. *Categories that performed better*

Here we filtered the dataset and just analyzed the top 25% performers for each one of the features. We wanted to have a clear overview how the categories perform compared between them. The outliers were removed as explained in section II B 4.

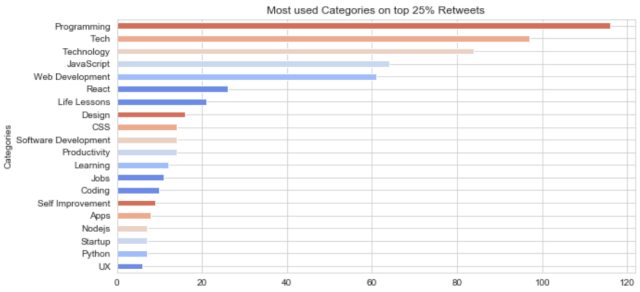

FIG. 6. Best Categories for Re-tweet

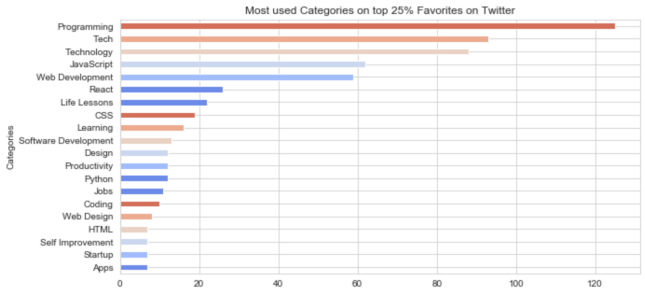

FIG. 7. Best Categories for Favorite

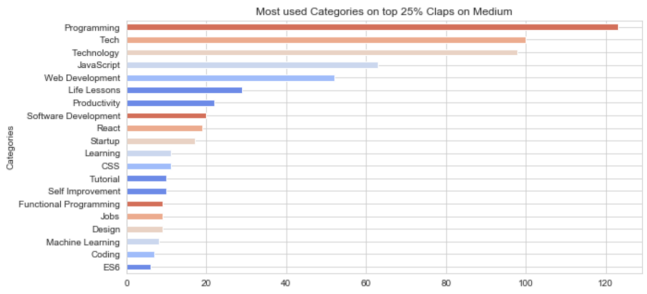

FIG. 8. Best Categories for Claps

From this statistic we notice that articles created with the following categories can increase the number of re-tweets, favorites and claps: "Programming", "Tech", "Technology", "JavaScript" and "Web Development".

6. *Words that performed better*

We repeated the same strategy of limiting the 25% performers for the words on the title of the article. We wanted to understand if there are words that can boost the interaction from the readers. The outliers were removed as explained in section II B 4.

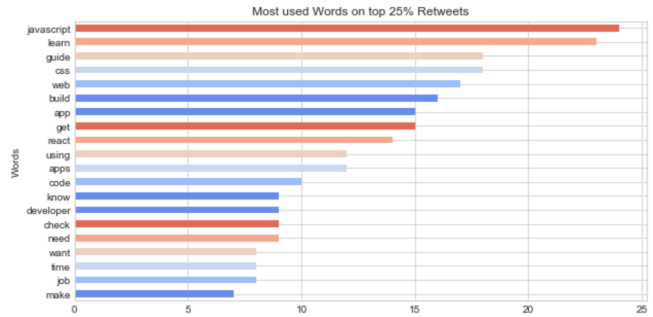

FIG. 9. Best Words for Re-tweet

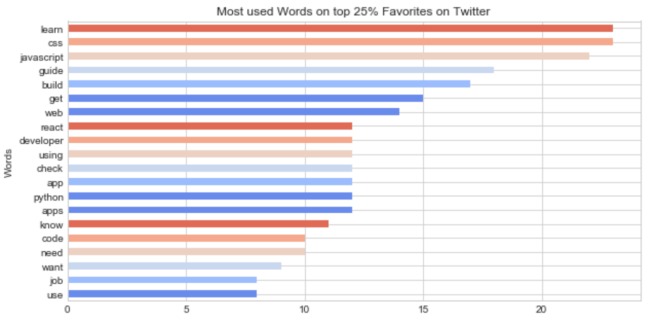

FIG. 10. Best Words for Favorite

In this lexical analysis, we can notice that some words get much more attention on the FreeCodeCamp community than others, we noticed if we want to make our ar-

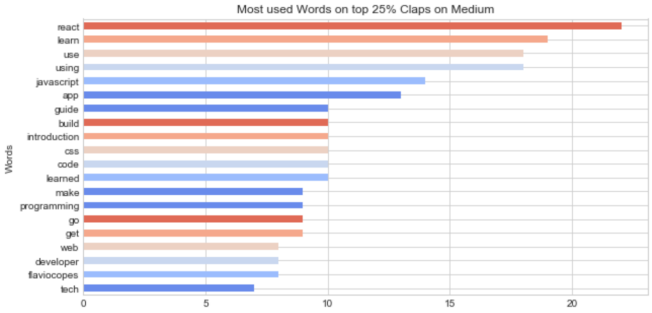

FIG. 11. Best Words for Claps

ticles reach further in numbers talking about JavaScript, React or CSS will increase this change. Using the words "learn" or "guide" to describe will also increase this probability.

C. Algorithms and Techniques

Classification is a common task of machine learning (ML), which involves predicting a target variable taking in consideration the previous data [12]. To reach such classification, it is necessary to create a model with the previous training data, and then use it to predict the value of the test data [13]. This process is called Supervised Learning, since the data processing phase is guided toward the class variable while building the model [14].

Predicting number of shares and favorites of an article can be treated as a classification problem, because the output will be discrete values (range of shares and favorites). As input, the title of the articles with each word as a token t1, t2, t3, ... tn.

For this task we will evaluate the following algorithms: Support Vector Machines (SVM), Decision Trees, Gaussian Naive Bayes (GaussianNB), K-Nearest Neighbors and Logistic Regression. In the end, it will be compared the performance of each one of them and one will be chosen. To estimate accuracy, it will be used a 5-fold cross validation, that splits the dataset in 5 parts, 4 of training and 1 of testing. The implementation of this project was made using Python, Numpy [15] and Scikit [16].

D. Benchmark

This project run the same testing and training data for multiple algorithms, the comparison between them was used to evaluate the overall performance. The overall benchmark was made comparing our data with the Logistic Regression results.

III. METHODOLOGY

A. Data Preprocessing

1. *Data cleaning*

The first part of the data processing was to clean the dataset. After downloading the tweets, we removed the ones that didn't have any URL (that points to the Medium article) or title. Data points with values of favorites, claps or re-tweets that were not positive numbers or zero were also excluded.

Some of the data points have same URL, it means, that they shared more than once on the account of Twitter. After analyzing each one of the duplicates, we noticed that there were two types of re-tweets: same URL and same title; and same URL different title. We removed the ones of the first type. For the second type, we left, because the titles were completely rewritten and it can be considered as one different data point.

2. *Assigning classes to the dataset*

For this project, we decided to classify the number of re-tweets and favorites in ranges. We wanted to make use of the properties of the Classification family of the Supervised Learning algorithms.

To avoid the Class Imbalance Problem [], we divided the dataset in similar groups, as shown in image 12.

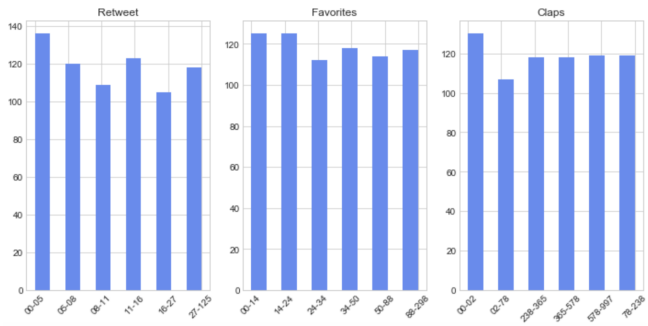

FIG. 12. Range Distribution

The defined ranges for our features as:

1. Re-tweets: 0-5, 5-8, 8-11, 11-16, 16-27 and 27-125
2. Favorites: 0-14, 14-24, 24-34, 34-50, 50-88 and 88-298
3. Claps: 0-2, 2-78, 78-238, 238-365, 365-578 and 578-997

Where the range 'x-y', means bigger than x and less equal than y. The first range item of each feature also contains the zero on the range.

VII. PROJECT DESIGN

The steps to solve the proposed problem will take as reference the one proposed by S. Raschka [17] image 13. They are:

1. **Data gathering**: This step is responsible to get the datasets that will be used on to analyze, train and test the models. This data was already gathered from Twitter and Medium and aggregated like showed in the section "Datasets and Inputs".

2. **Data pre-processing:** The dataset will be cleaned, formatted or added the missing values.

3. **Exploring Data**

   (a) **Prepare environment to run the simulations:** The environment used to make this simulation will be a Jupyter Notebook. For each of the steps, we will describe what is expected and show the Python code used for the implementation.

   (b) **Training and Testing Data Split:** It will be defined the sets for training and testing. From the overall data points that we have 719, 143 will be testing data and 576 training.

4. Training and Evaluating Models

   (a) **Model Performance Metrics Implementation:** The chosen Evaluation Metric will be implemented to analyze how the each of the models performed.

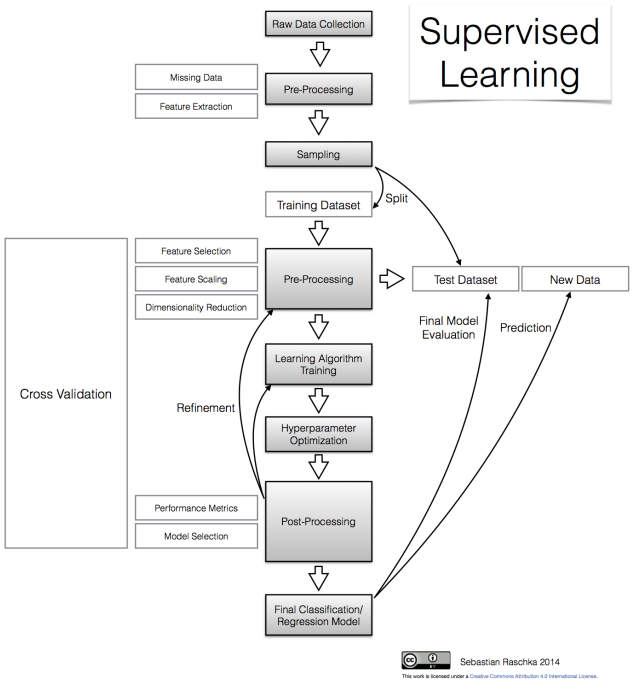

FIG. 13. Sebastians Raschka's workflow [17]

   (b) **Models Implementation:** Implement all the algorithms of supervised learning chosen to test how each of them perform for such dataset.

   (c) **Model Performance Metrics:** Evaluate all the models and make a comparison between their performance.

   (d) **Model Tuning:** Tune the algorithms to try to find the best parameters.

   (e) **Reiterate:** Reiterate the previous steps and check how the performance is evolving.

5. **Choosing the Best Model:** Decide the best model to make the desired prediction.

REFERENCES

[1] *eMarketer Report. (2017). US Time Spent with Media: eMarketer's Updated Estimates for 2017. Accessed 9 Aug. 2018. Available at: https://www.emarketer.com/Report/US-Time-Spent-with-Media-eMarketers-Updated-Estimates-2017/2002142.*

[2] *Common Sense Media. (2015). The Common Sense Census: Media Use by Tweens and Teens. Accessed 9 Aug. 2018. Available at: https://www.commonsensemedia.org/research/the-common-sense-census-media-use-by-tweens-and-teens.*

[3] *Lindgaard, Gitte Fernandes, Gary Dudek, Cathy M. Brown, Judith. (2006). Attention web designers: You*

have 50 milliseconds to make a good first impression! Behaviour and Information Technology, 25(2), 115-126. Behaviour IT. 25. 115-126. 10.1080/01449290500330448.

[4] *Shearer, E., Gottfried, J. (2017). News use across social media platforms 2017. Accessed 9 Aug. 2018. Available at: http://www.journalism.org/2017/09/07/news-use-across-social-media-platforms-2017/*.

[5] *Chen, Yimin, Niall J. Conroy, and Victoria L. Rubin. "Misleading Online Content: Recognizing Clickbait as "False News"" ResearchGate. ACM WMDD, 9 Nov. 2015.*

[6] *Lex, Elisabeth, Andreas Juffinger, and Michael Granitzer. "Objectivity Classication in Online Media." ResearchGate. Proceedings of the 21st ACM Conference on Hypertext and Hypermedia, 13 June 2010.*

[7] *Glorot, Xavier, Antoine Bordes, and Yoshua Bengio. "Domain Adaptation for Large-Scale Sentiment Classification: A Deep Learning Approach." WUSTL. Proceedings of the 28 Th International Conference on Machine Learning, 2011.*

[8] *Twitter. Accessed 13 Aug. 2018. Available at: https://www.twitter.com*.

[9] *Medium. Accessed 13 Aug. 2018. Available at: https://www.medium.com*.

[10] *freecodecamp. Accessed 13 Aug. 2018. Available at: https://medium.freecodecamp.org/*.

[11] *freecodecamp.org. Accessed 13 Aug. 2018. Available at: https://twitter.com/freecodecamp*.

[12] *N. Abdelhamid, A. Ayesh, F. Thabtah, S. Ahmadi, W. Hadi. MAC: A multiclass associative classification algorithm J. Info. Know. Mgmt. (JIKM), 11 (2) (2012), pp. 125001-1-1250011-10 WorldScinet*.

[13] *I.H. Witten, E. Frank, M.A. Hall. Data Mining: Practical Machine Learning Tools and Techniques. Morgan Kaufmann, Burlington, MA (2011)*.

[14] *Thabtah, S. Hammoud, H. Abdeljaber, Parallel associative classification data mining frameworks based mapreduce, To Appear in Journal of Parallel Processing Letter, March 2015, World Scientific, 2015.*

[15] *NumPy. Accessed 14 Aug. 2018. Available at: http://www.numpy.org/*.

[16] *scikit-learn. Accessed 14 Aug. 2018. Available at: http://scikit-learn.org/stable/*.

[17] *S. Raschka. Predictive modeling, supervised machine learning, and pattern classification. Accessed 14 Aug. 2018. Available at: https://sebastianraschka.com/Articles/* $2014_i ntro_s upervised_l earning.html$.