Udacity – Data Analyst Nanodegree Program

WRANGLING WeRateDogs TWITTER DATA TO CREATE INTERESTING AND TRUSTWORTHY EXPLORATORY / PREDICTIVE ANALYSES AND VISUALIZATION USING DIFFERENT MACHINE LEARNING ALGORITHMS

Project 8

Esra Arı

İSTANBUL, 2018

EXECUTIVE SUMMARY

WRANGLING WeRateDogs TWITTER DATA TO CREATE INTERESTING AND TRUSTWORTHY EXPLORATORY / PREDICTIVE ANALYSES AND VISUALIZATION USING DIFFERENT MACHINE LEARNING ALGORITHMS

Esra Arı

Increasing usage of social media in recent years has increased knowledge in these environments. Increased information intensity has made gaining data from social media to do both exploratory and predictive analysis so popular. However, almost all of the large datasets obtained are uncleaned / raw data. Therefore, the assessing and cleaning of the data is at least as important as the exploratory and predictive analysis. The open source WeRateDogs twitter account tweets have been gathered, assessed, cleaned, analyzed and predicted for this thesis. As a result of the study, it was understood that the most important and most time consuming part of the predictive data analysis is the data gathering and cleaning. As a result of this project, probability of dog's breed whether retriever or not is predicted from the tweet's text body. The performance of the model has been increased seriously by doing oversampling in the data sets which contain low event observation. At the same time, the decision tree and random forest algorithms are compared and it is shown that the random forest's model performance is better than the decision tree models.

TABLE OF CONTENTS

1. INTRODUCTION

Application of big data platforms and tool's usage is getting so popular in the recent years with increasing number of data accumulation. Many articles have emphasized how important twitter data is actually in terms of prediction (Gayo-Avello, 2012). There are many platforms and languages to gather, asses, clean, modifiy and analysis the data. All these platforms are differentiated with each other for the purpose of usage. While some tools are good in gather and stroge data such as SQL, some tools are realy convenient in exploratory and predictive analysis.

It is important to keep in mind that real-world data rarely comes clean. Therefore; usage of R, Python and its' libraries, data will be gathered from a variety of sources and in a variety of formats, be assessed its quality and tidiness, then data should be cleaned. This is called data wrangling. Without doing data wrangling, it is impossible to make any descriptive and predictive analyses.

The dataset that will be wrangled and predicted is the tweet archive of Twitter user @dog_rates, also known as WeRateDogs. WeRateDogs is a Twitter account that rates people's dogs with a humorous comment about the dogs. These ratings almost always have a denominator of 10. Almost always greater than 10. 11/10, 12/10, 13/10, etc. because they are good dogs. WeRateDogs has over 4 million followers and has received international media coverage.

WeRateDogs' Twitter archive was downloaded to use in this project. This archive contains basic tweet data (tweet ID, timestamp, text, etc.) for all 5000+ of their tweets as they stood on August 1, 2017.

Figure 1. Image taken from Boston Magazine[1]

[1] Retrieved from: https://www.bostonmagazine.com/arts-entertainment/2017/04/18/dog-rates-mit/

1.1 Which Software should be used?

In this project the following software requirements used:

- Jupyter Notebook gives easy to understand documentation.
- The following packages (libraries) needed to be installed.
    - pandas
    - NumPy
    - requests
    - tweepy
    - json
    - re
    - seaborn
    - os
    - matplotlib.pyplot
- Microsoft Azure Machine Learning Studio

1.1 Aim of Project

Project goal is wrangling WeRateDogs twitter data to create interesting and trustworthy exploratory / predictive analyses and visualization using different machine learning algorithms. The Twitter archive is great, but it only contains very basic tweet information. During this project, additional data gathering, then assessing and cleaning have been completed for worthy analyses and visualizations. In addition to that trying different machine learning algorithms for both unsupervised sides like data extraction, dimension reduction and supervised side like random forest, decision tree increased the model performance.

1.1.1 Enhancing the WeRateDogs twitter archive

The WeRateDogs Twitter archive contains basic tweet data for all 2356 of their tweets. So far following features were generated: each tweet's text, which is used to extract rating, dog name, and dog "stage" (i.e. doggo, floofer, pupper, and puppo) to make this Twitter archive "enhanced.". The extracted data from each tweet’s text is shown below.

text	rating_ numerator	rating_ denominator	name	doggo	floofer	pupper	puppo
This is Phineas. He's a mystical boy. Only ever appears in the hole of a donut. 13/10 https://t.co/MgUWQ76dJU	13	10	Phineas	None	None	None	None
This is Tilly. She's just checking pup on you. Hopes you're doing ok. If not, she's available for pats, snugs, boops, the whole bit. 13/10	13	10	Tilly	None	None	None	None
This is Archie. He is a rare Norwegian Pouncing Corgo. Lives in the tall grass. You never know when one may strike. 12/10 https://t.co	12	10	Archie	None	None	None	None
This is Darla. She commenced a snooze mid meal. 13/10 happens to the best of us https://t.co/tD36da7qLQ	13	10	Darla	None	None	None	None
This is Franklin. He would like you to stop calling him "cute." He is a very fierce shark and should be respected as such. 12/10 #BarkW	12	10	Franklin	None	None	None	None
Here we have a majestic great white breaching off South Africa's coast. Absolutely h*ckin breathtaking. 13/10 (IG: tucker_marlo) #Bar	13	10	None	None	None	None	None
Meet Jax. He enjoys ice cream so much he gets nervous around it. 13/10 help Jax enjoy more things by clicking below https://t.co/Zr4hWfAs1H https://t.co/tVJBRMnhxl	13	10	Jax	None	None	None	None
When you watch your owner call another dog a good boy but then they turn back to you and say you're a great boy. 13/10 https://t.co/	13	10	None	None	None	None	None
This is Zoey. She doesn't want to be one of the scary sharks. Just wants to be a snuggly pettable boatpet. 13/10 #BarkWeek https://t.c	13	10	Zoey	None	None	None	None
This is Cassie. She is a college pup. Studying international doggo communication and stick theory. 14/10 so elegant much sophisticat	14	10	Cassie	doggo	None	None	None
This is Koda. He is a South Australian deckshark. Deceptively deadly. Frighteningly majestic. 13/10 would risk a petting #BarkWeek ht	13	10	Koda	None	None	None	None
This is Bruno. He is a service shark. Only gets out of the water to assist you. 13/10 terrifyingly good boy https://t.co/u1XPQMl29g	13	10	Bruno	None	None	None	None
Here's a puppo that seems to be on the fence about something haha no but seriously someone help her. 13/10 https://t.co/BxvuXk0UC	13	10	None	None	None	None	puppo
This is Ted. He does his best. Sometimes that's not enough. But it's ok. 12/10 would assist https://t.co/f8dEDcrKSR	12	10	Ted	None	None	None	None
This is Stuart. He's sporting his favorite fanny pack. Secretly filled with bones only. 13/10 puppared puppo #BarkWeek https://t.co/y70	13	10	Stuart	None	None	None	puppo

Figure 2. Extracted Data

The data is programmatically extracted. The ratings probably aren't all correct. Same is valid for the dog names and probably dog stages too. Therefore, data is needed to be assessed and cleaned.

1.1.2 Additional data via the twitter API

Back to the basic-ness of Twitter archives: retweet count and favorite count are two of the notable column omissions. Fortunately, this additional data can be gathered by anyone from Twitter's API. Therefore, this valuable data will be gathered querying Twitter's API. Details is explained in chapter 1.2..

1.1.3 Image predictions filed

Also, there is a neural network that can classify breeds of dogs. The results: a table full of image predictions (the top three only) alongside each tweet ID, image URL, and the image number that corresponded to the most confident prediction (numbered 1 to 4 since tweets can have up to four images) are given by Udacity e-learning platform.

tweet_id	jpg_url	img_num	p1	p1_conf	p1_dog	p2	p2_conf	p2_dog	p3	p3_conf	p3_dog
892177421306343426	https://pbs.twimg.cor	1	Chihuahua	0.323581	TRUE	Pekinese	0.0906465	TRUE	papillon	0.0689569	TRUE
891815181378084864	https://pbs.twimg.cor	1	Chihuahua	0.716012	TRUE	malamute	0.078253	TRUE	kelpie	0.0313789	TRUE
891689557279858688	https://pbs.twimg.cor	1	paper_towel	0.170278	FALSE	Labrador_retriever	0.168086	TRUE	spatula	0.0408359	FALSE
891327558926688256	https://pbs.twimg.cor	2	basset	0.555712	TRUE	English_springer	0.22577	TRUE	German_short-haired_pointer	0.175219	TRUE
891087950875897856	https://pbs.twimg.cor	1	Chesapeake_Bay_retriever	0.425595	TRUE	Irish_terrier	0.116317	TRUE	Indian_elephant	0.0769022	FALSE
890971913173991426	https://pbs.twimg.cor	1	Appenzeller	0.341703	TRUE	Border_collie	0.199287	TRUE	ice_lolly	0.193548	FALSE
890729181411237888	https://pbs.twimg.cor	2	Pomeranian	0.566142	TRUE	Eskimo_dog	0.178406	TRUE	Pembroke	0.0765069	TRUE
890609185150312448	https://pbs.twimg.cor	1	Irish_terrier	0.487574	TRUE	Irish_setter	0.193054	TRUE	Chesapeake_Bay_retriever	0.118184	TRUE
890240255349198849	https://pbs.twimg.cor	1	Pembroke	0.511319	TRUE	Cardigan	0.451038	TRUE	Chihuahua	0.0292482	TRUE
890006608113172480	https://pbs.twimg.cor	1	Samoyed	0.957979	TRUE	Pomeranian	0.0138835	TRUE	chow	0.00816748	TRUE
889880896479866881	https://pbs.twimg.cor	1	French_bulldog	0.377417	TRUE	Labrador_retriever	0.151317	TRUE	muzzle	0.0829811	FALSE
889665388333682689	https://pbs.twimg.cor	1	Pembroke	0.966327	TRUE	Cardigan	0.0273557	TRUE	basenji	0.00463323	TRUE
889638837579907072	https://pbs.twimg.cor	1	French_bulldog	0.99165	TRUE	boxer	0.00212864	TRUE	Staffordshire_bullterrier	0.00149818	TRUE
889531135344209921	https://pbs.twimg.cor	1	golden_retriever	0.953442	TRUE	Labrador_retriever	0.0138341	TRUE	redbone	0.00795775	TRUE

Figure 3. Tweet Image Prediction Data

So for the last row in that table:

- tweet_id is the last part of the tweet URL after "status/": https://twitter.com/dog_rates/status/889531135344209921
- p1 is the algorithm's #1 prediction for the image in the tweet: golden retriever
- p1_conf is how confident the algorithm is in its #1 prediction: 95%
- p1_dog is whether or not the #1 prediction is a breed of dog: TRUE
- p2 is the algorithm's second most likely prediction: Labrador retriever
- p2_conf is how confident the algorithm is in its #2 prediction: 1%
- p2_dog is whether or not the #2 prediction is a breed of dog: TRUE
- etc.

For instance, the #1 prediction for the image in that tweet was spot on:

Figure 4. Taken from Twitter:[2]

[2] Twitter. Retrieved from: https://twitter.com/dog_rates/status/889531135344209921

2. METHODOLOGY

Tasks in this project are given following:

- Data wrangling, which consists of:
    - Gathering data
    - Assessing data
    - Cleaning data (Missing value treatment)
    - Storing/Exporting data

- Exploratory Data Analysis
    - Analyzing data
    - Visualizing data

- Predictive Data Analysis
    - Editing Metadata
    - Missing Value Treatment
    - Feature Extraction / Feature Hashing
    - Dimension Reduction / Principle Component Analysis
    - Using different sampling techniques such as oversampling
    - Data splitting
    - Trying different supervised machine learning algorithms with different parameters. (Random forest and boosted decision tree algorithms were applied for this project on Azure network)
- Reporting on 1) data wrangling efforts and 2) data analyses and visualizations 3) prediction methodology in an executive way with Microsoft word
- In the appendix, clearly defined data munging and data analysis efforts are attached.

2.1 Gathering Data for this Project

The three pieces of data gathered as described below in a Jupyter Notebook

1. The WeRateDogs Twitter archive includes 2356 observations and 17 features.
2. The tweet image predictions, i.e., what breed of dog (or other object, animal, etc.) is present in each tweet according to a neural network. This file (image_predictions.tsv) is hosted on Udacity's servers. It has 2075 entries and 12 columns without any missing values.
3. Each tweet's retweet count and favorite ("like") count at minimum, and any additional data. Using the tweet IDs in the WeRateDogs Twitter archive, query the Twitter API which is mentioned in Twitter API section detailed for each tweet's JSON data using Python's Tweepy library and store each tweet's entire set of JSON data in a file called tweet_json.txt file. Each tweet's JSON data should be written to its own line. Then read this .txt file line by line into a pandas DataFrame with (at minimum) tweet ID, retweet count, and favorite count. It includes 3715 entries and 11 columns.

2.2 Assessing Data for this Project

After gathering each of the above pieces of data, it is required to assess them visually and programmatically for quality and tidiness issues. With this aim in this section, each three pandas data frame gathered previous section will be investigated. Assessing data was done both visually (scrolling through the data in your preferred software application) and programmatically (using code to view specific portions and summaries of the data). Both quality and tidiness issue were noted end of this section. Also, sources of low quality /dirty and messy/untidy data were mentioned shortly.

2.2.1 Sources of dirty and messy data

Dirty data is also called as low quality data or content issues. There are lots of sources of dirty data. Basically, anytime humans are involved, there's going to be dirty data. There are lots of ways in which we touch data we work with.

- user entry errors
- no data coding standards, or having standards poorly applied, causing problems in the resulting data
- integrated data where different schemas have been used for the same type of item

- legacy data systems, where data wasn't coded when disc and memory constraints were much more restrictive than they are now. Over time systems evolve. Needs change, and data changes
- no unique identifiers it should
- lost in transformation from one format to another
- programmer error
- corrupted in transmission or storage by cosmic rays or other physical phenomenon

Messy data is also called as untidy data or structural issues. Messy data is usually the result of poor data planning. Or a lack of awareness of the benefits of tidy data. Fortunately, messy data is usually much more easily addressable than most of the sources of dirty data mentioned above.

2.2.2 Noted quality and tidiness issues

df1 : WeRateDogs Twitter Dataset

It includes 2356 entries and 17 columns.

- **Quality**
  - Names column should be cleaned, there is invalid records like a, the, an, the, very, unacceptable which is start with lowercase.
  - timestamp, retweeted_status_timestamp column type should be date instead of object.
  - text includes "'&" instead of "&".
  - invalid rating_denominator (different than 10). However, I checked them manually and they are true denominators, so there is no problematic extraction from text.
  - Tweets_ids with no images however this problem will be solved when I joined with image prediction dataset. Because, I am not expecting the prediction which do not have any image.
  - Missing values expressed as "none". (name, duppo, flopper, etc.)
  - in_reply_to_status_id, in_reply_to_user_id, retweeted_status_id, retweeted_status_user_id data types should be integer instead of float.

- We only want original ratings (no retweets) that have images
- From the source column, via which channel users connected to twitter. Therefore, column should be cleaned.
- excluding any tweet that is a retweet.

Tidiness

- joining with tables df2 and df3.
- getting together stages in one column.
- adding new features like gender, etc.

df2 : Image Prediction Dataset

Great dataset which has 2075 entries and 12 columns without any missing values.

Quality

- Missed ID's exists in the dataset compare to d1
- Duplicated jpg_url
- p1, p2, p3 columns should be standardized as all lowercase and "-" expression should be removed.

Tidiness

- joining with tables df3 and df1.
- creating final dog prediction

df3 : Tweepy API Dataset

It includes 3715 entries and 11 columns.

Quality

- contributors, coordinates, place and geo features should be excluded due to high missing ratio.
- 1222 numbers of id variable are duplicated
- id=666337882303524864 exits 4 times in the dataset with same results.

- lang indicates that the language of tweet. I wondered how "tl" lang is texted. Then, I realized id=668967877119254528 is problematic input.

Tidiness

- joining with tables df2 and df1.
- favorited, retweeted columns include always false inputs, therefore it should be excluded.

2.3 Cleaning Data for this Project

In this section, defined tidiness and quality issues were cleaned. As it can be seen from below picture, 19 data problems were defined, coded and tested one by one.

Define (5)

Creating final prediction of image prediction

```
In [33]: p_final = []
         p_final_conf = []

         def final_prediction(table) :
             if table['p1_dog'] == True:
                 p_final.append(table['p1'])
                 p_final_conf.append(table['p1_conf'])
             elif table['p2_dog'] == True:
                 p_final.append(table['p2'])
                 p_final_conf.append(table['p2_conf'])
             elif table['p3_dog'] == True:
                 p_final.append(table['p3'])
                 p_final_conf.append(table['p3_conf'])
             else:
                 p_final.append('NaN')
                 p_final_conf.append(0)
```

Code

```
In [109]: df_all.apply(final_prediction, axis=1)
          df_all['final_prediction'] = p_final
          df_all['final_prediction_conf'] = p_final_conf
```

Test

```
In [110]: df_all.final_prediction.value_counts()

Out[110]: NaN                   550
          golden_retriever      143
          Labrador_retriever    103
          Pembroke               94
```

Figure 5. Cleaning example[3]

[3] Detailed codes can be find from part 1 jupyter notebook.

2.4 Storing

Assed and cleaned data was stored in a CSV file with the main one named twitter_archive_master.csv.

2.5 Exploratory Data Analysis

In the data wrangling part, I gathered, assessed and cleaned data comes from three different sources. As explained in the Jupiter notebook (Part 1), most of data quality and tidiness issue was improved (19 problematic points were defined, coded and tested).

Exploratory Data Analysis (EDA) is the numerical and graphical examination of data characteristics and relationships before formal, rigorous statistical analyses are applied.

EDA can lead to insights, which may uncover to other questions, and eventually predictive models. It also is an important “line of defense” against bad data and is an opportunity to notice that your assumptions or intuitions about a data set are violated. Therefore, in this part, I will try to explore data both quantitatively and visually. Also, I will decide what I am going to predict from tweet's information in accordance with exploration. Possible prediction features outstand in the wrangling section are listed below.

- Predicting score using text, tweet information like number of retweeted, favorited, etc. and image prediction result.
- Predicting dogs' breed using using text, tweet information like number of retweeted, favorited, etc.

2.5.1 Uni-multi variate data analysis

Let's remember basic information about dataset

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1625 entries, 0 to 1624
Data columns (total 34 columns):
retweet_count            1625 non-null int64
favorite_count           1625 non-null int64
lang                     1625 non-null object
created_at               1625 non-null object
tweet_id                 1625 non-null float64
timestamp                1625 non-null object
source                   1625 non-null object
text                     1625 non-null object
expanded_urls            1625 non-null object
rating_numerator         1625 non-null float64
rating_denominator       1625 non-null float64
name                     1625 non-null object
doggo                    1625 non-null object
floofer                  1625 non-null object
pupper                   1625 non-null object
puppo                    1625 non-null object
jpg_url                  1625 non-null object
img_num                  1625 non-null float64
p1                       1625 non-null object
p1_conf                  1625 non-null float64
p1_dog                   1625 non-null bool
p2                       1625 non-null object
p2_conf                  1625 non-null float64
p2_dog                   1625 non-null bool
p3                       1625 non-null object
p3_conf                  1625 non-null float64
p3_dog                   1625 non-null bool
final_prediction         1625 non-null object
final_prediction_conf    1625 non-null float64
new_dog_names            1158 non-null object
dog_gender               727 non-null object
date                     1625 non-null object
time                     1625 non-null object
stage                    1625 non-null object
dtypes: bool(3), float64(8), int64(2), object(21)
memory usage: 398.4+ KB
```

Figure 6. Information about dataset

	retweet_count	favorite_count	tweet_id	rating_numerator	rating_denominator	img_num	p1_conf	p2_conf	p3_conf	final_prediction
count	1625.000000	1625.000000	1.625000e+03	1625.000000	1625.000000	1625.000000	1625.000000	1625.000000	1.625000e+03	1625.0
mean	2493.293538	8520.427077	7.384255e+17	11.457846	10.554462	1.216615	0.605994	0.136341	6.108134e-02	0.5
std	4337.790720	12106.593738	6.833344e+16	8.254696	7.074351	0.577573	0.267350	0.101156	5.183068e-02	0.3
min	13.000000	80.000000	6.660209e+17	0.000000	2.000000	1.000000	0.044333	0.000010	2.160900e-07	0.0
25%	605.000000	2033.000000	6.769579e+17	10.000000	10.000000	1.000000	0.379055	0.054787	1.588320e-02	0.3
50%	1311.000000	4049.000000	7.106587e+17	11.000000	10.000000	1.000000	0.609715	0.120481	4.981050e-02	0.5
75%	2877.000000	10575.000000	7.931506e+17	12.000000	10.000000	1.000000	0.853684	0.197897	9.451960e-02	0.8
max	76893.000000	142654.000000	8.921774e+17	165.000000	150.000000	4.000000	0.999984	0.467678	2.734190e-01	0.9

Figure 7. Descriptive Statistics of dataset

It can be seen that there are outliers in confidence features such as p1_conf, p2_conf, etc. Because, I have already created one feature called final_prediction value shows final prediction of dogs' breed. I will not explore these variables and I will exclude them before starting the model.

	lang	created_at	timestamp	source	text	expanded_urls	name	doggo	floofer	pupper
count	1625	1625	1625	1625	1625	1625	1625	1625	1625	1625
unique	4	1625	1625	3	1625	1625	828	2	2	2
top	en	Thu Mar 23 00:18:10 +0000 2017	2015-11-24 04:17:01	Twitter for iPhone	We only rate dogs. Please don't send perfectly...	https://twitter.com/dog_rates/status/682303737...	None	None	None	None
freq	1620	1	1	1596	1	1	404	1566	1617	1454

Figure 8. Descriptive Statistics of dataset

Let's visualize distribution of numeric variables. Firstly, I would like to view all of them in one.

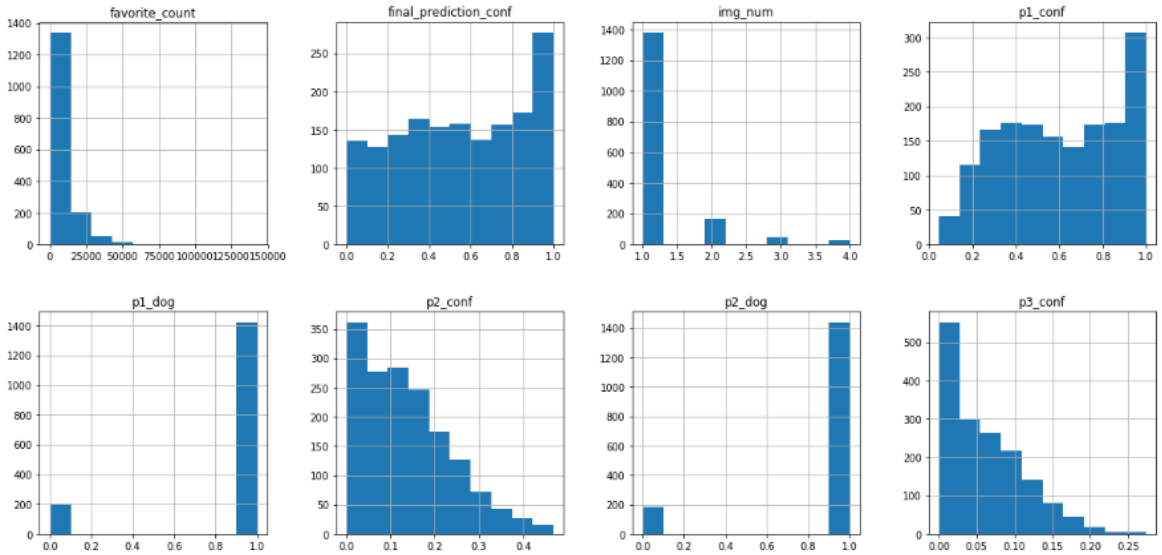

Figure 9. Distribution of numeric variables

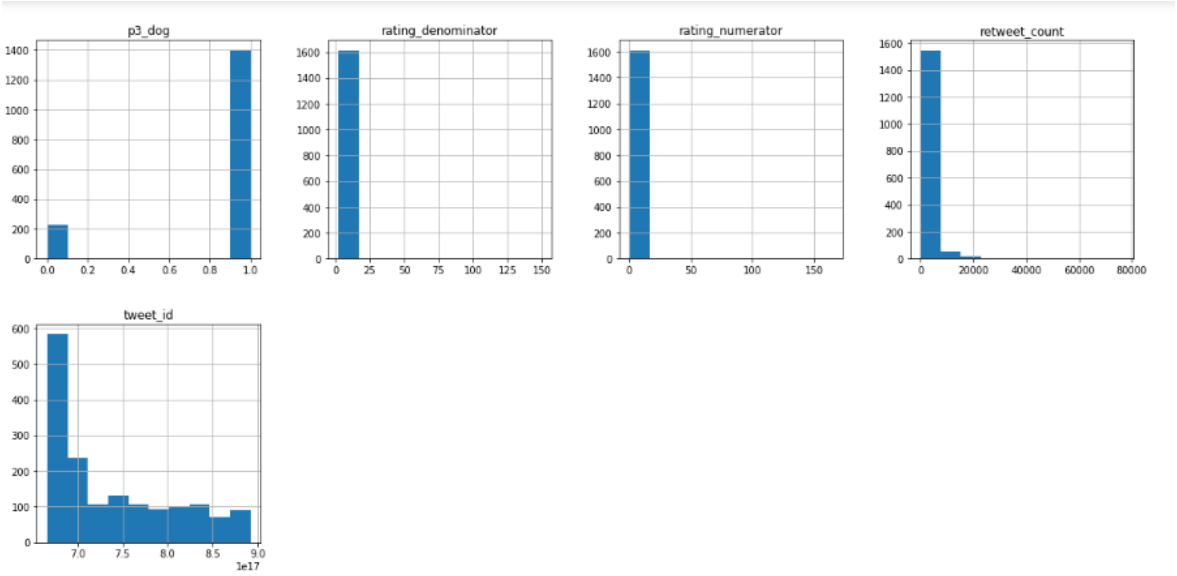

Figure 10. Distribution of numeric variables

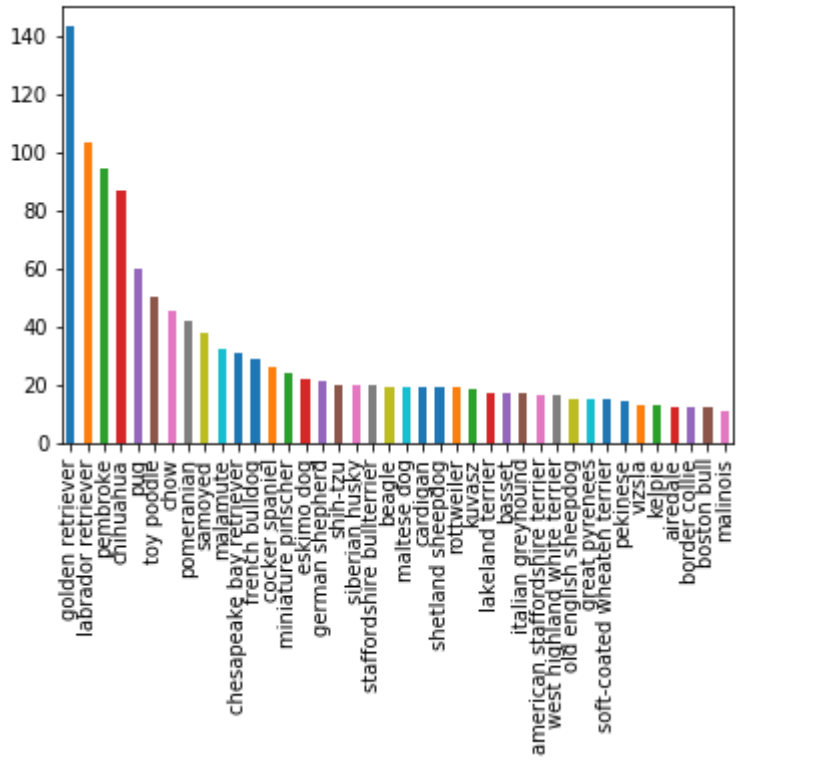

Figure 11. Investigation of predicted dogs' breed

It can be seen that final_prediction feature which includes predicted breeds of dog has so many unique value. Therefore, this graph gives us great intuition that predicting dog's breed cannot be good model. Instead of predicting it, I can try to understand whether dog's breed retriever or not.

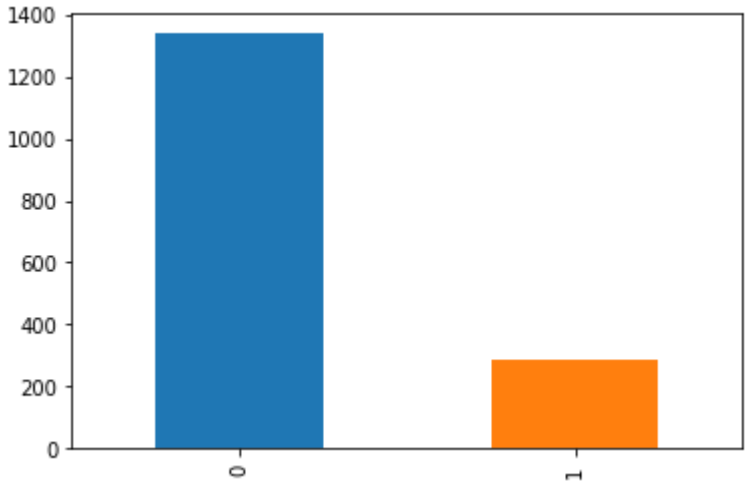

Figure 12. Retriever flag distribution

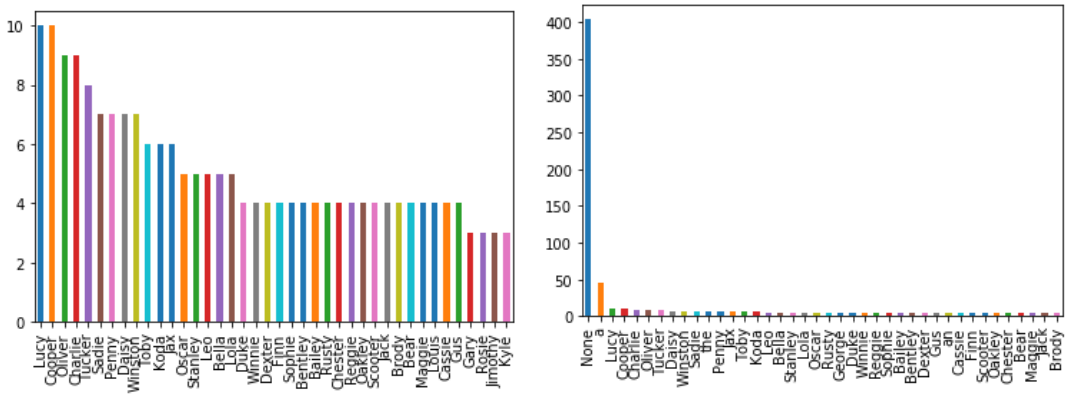

Figure 13. Comparing name and new named column created in the data wrangling part

Newly created dogs' name column includes more accurate, quality data than old name column. Therefore, I will drop name column before dive into any model.

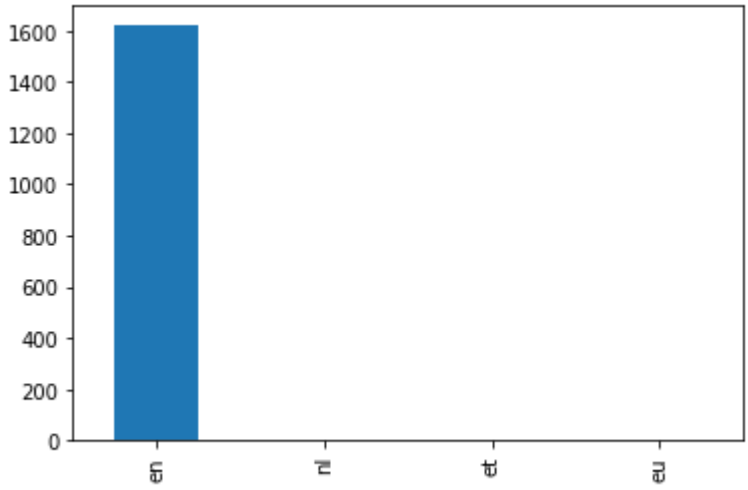

Figure 14. lang column investigation

Lang column gives the information about tweet language. It can be easily understood that most of tweets were written in English from the bar chart. Therefore, we can use text-hashing option in the further analysis during predictive analysis. In addition, I will drop this information because there is no info in it can be beneficial while doing prediction.

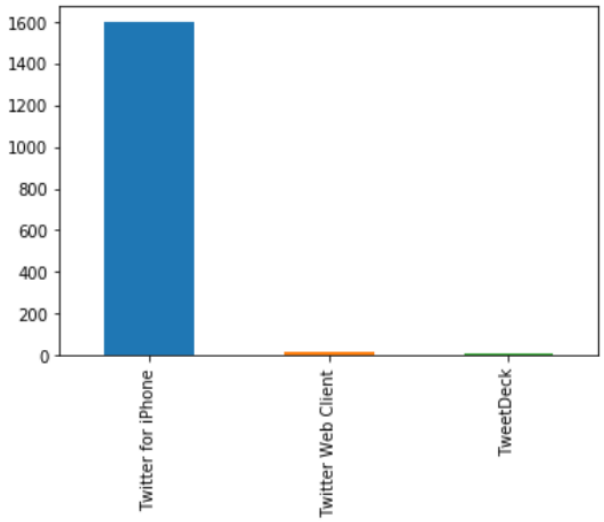

Figure 15. source column investigation

“source” column was extracted from url information column which gives us in which channel user shares the tweet. Most tweets published via twitter for IPhone, therefore like claimed in the “lang” column, this feature can be dropped as well.

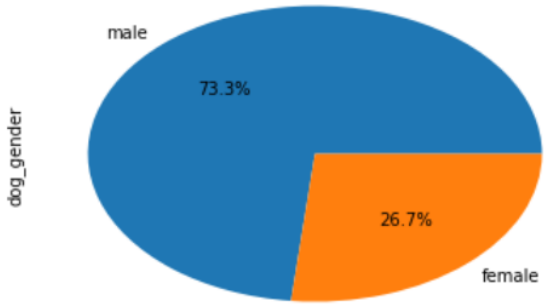

Figure 16. gender column investigation

To remember, gender column was derived from text in the tweet by manual. If text includes words like 'She', 'she', 'her', 'hers', 'herself', ‘she's’ classief as female else as male. To sum up, %73 percent of dog is male.

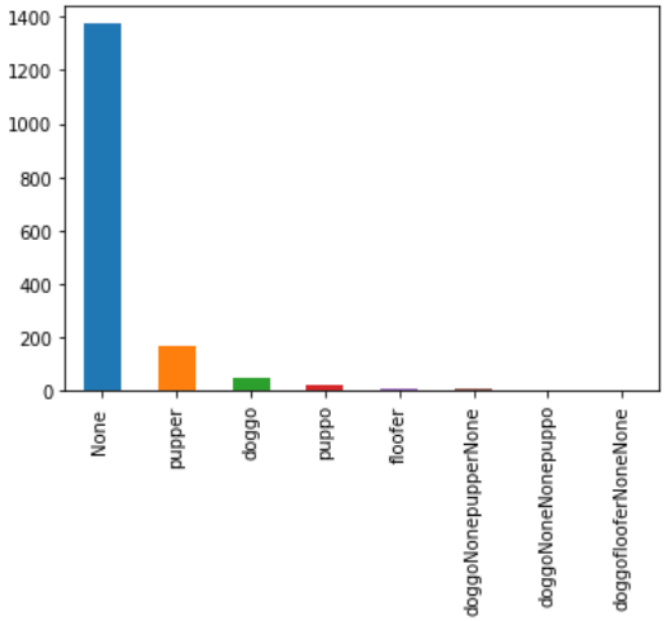

Figure 17. stage column investigation

“stages” column gives a information about dog’s stage. However, most of tweets do not includes dog’ stage information. However, even if small number of information gives this information, still it is worth to use in the prediction.

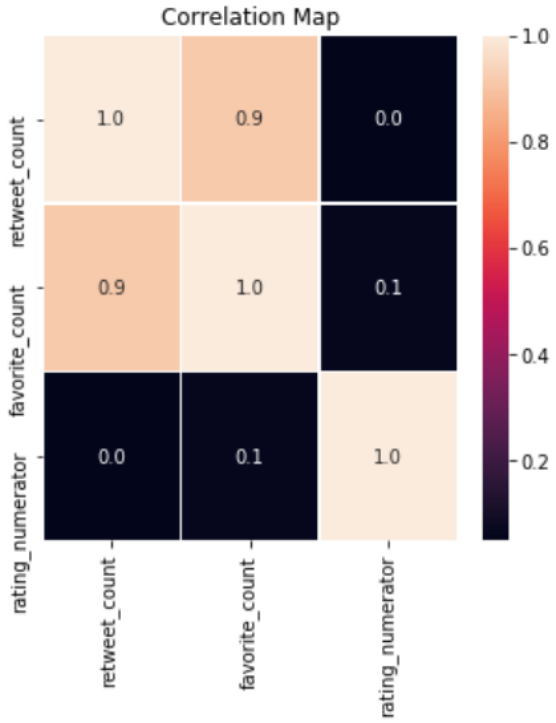

Figure 18. Correlation between numeric features

Retweet and favorite count have positive correlation with each other like expected. (0.9 positive correlation coefficient). It means that they move in the same way. However, we cannot see any relation between rating numerator gives dog' rating information. Therefore, it gives us great intuition about ratings are quite objective, they cannot be target variable for the further analysis.

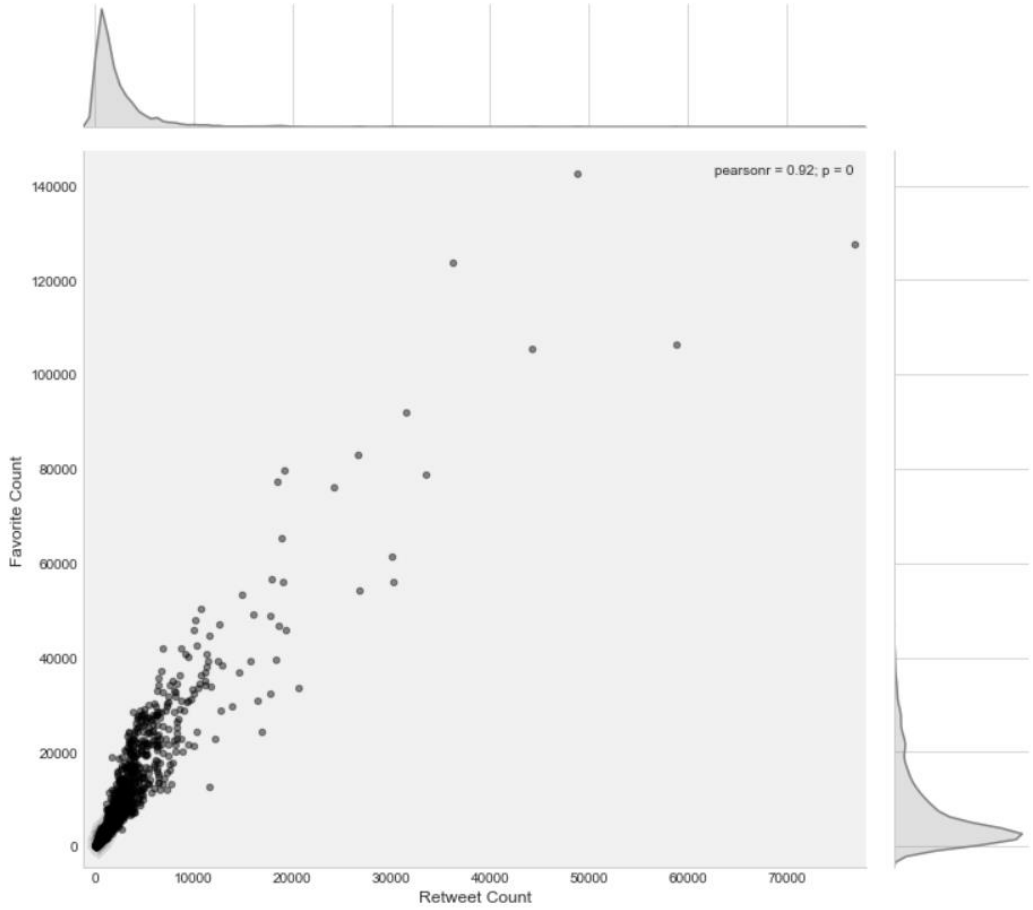

Figure 19. Correlation between retweet and favorite colums

When we look at distribution of favorite and retweet counts, most of tweets distributed between 0-20K for favorite counts and 0-10K for retweet count means have left skewed distributions. Also, outliers exist in the dataset.

5.5.2 Summary of EDA

According to univariate data analysis, some variables should be dropped due to existence of outliers, better alternatives or no information value such as p1, p1_dog, lang, name, etc.

In addition, final_prediction feature which includes predicted breeds of dog has so many unique value. Therefore, predicting dog's breed cannot be good target variable. Instead of predicting it, understanding whether dog’s breed retriever or not will be used for further analysis.

Final but not least, %73 percent of dog is male. Most of tweets distributed between 0-20K for favorite counts and 0-10K for retweet count means have left skewed distributions. Also, there is no relationship between dog's rating and favorite or retweet count.

2.6 Predictive Data Analysis

As explained in the exploratory data analysis part, there are 3 main options to make predictive analysis. First one was the predicting dog's rating, this option was ignored due to ambiguity and subjectivity of ratings shown in the analysis. Second option was the predicting dog's breed; however, this option was dropped as well because there are many unique values of dog's breed (+100) in very small number of observation (1.9K). Therefore, 3rd option which predicts whether dog's breed is retriever nor not is very good option because retriever breed is the most dominated breed in the dataset.

With this aim, following modelling steps have been completed on Microsoft Azure Machine Learning Studio. Overall experiment picture can be seen at below.

- Uploading cleaned dataset
- Editing metadata (correcting data types and properties)
- Doing Feature-hashing
- Reducing dimension with PCA,
- Selecting candidate model inputs
- Dividing two pipelines one for oversampling data, second one for normal process
- Splitting train-test
- Building models using 2 different machine learning algorithms with different parameters optimizing them with Tune Model Hypermeters node.
- Scoring both train and test datasets
- Comparing results

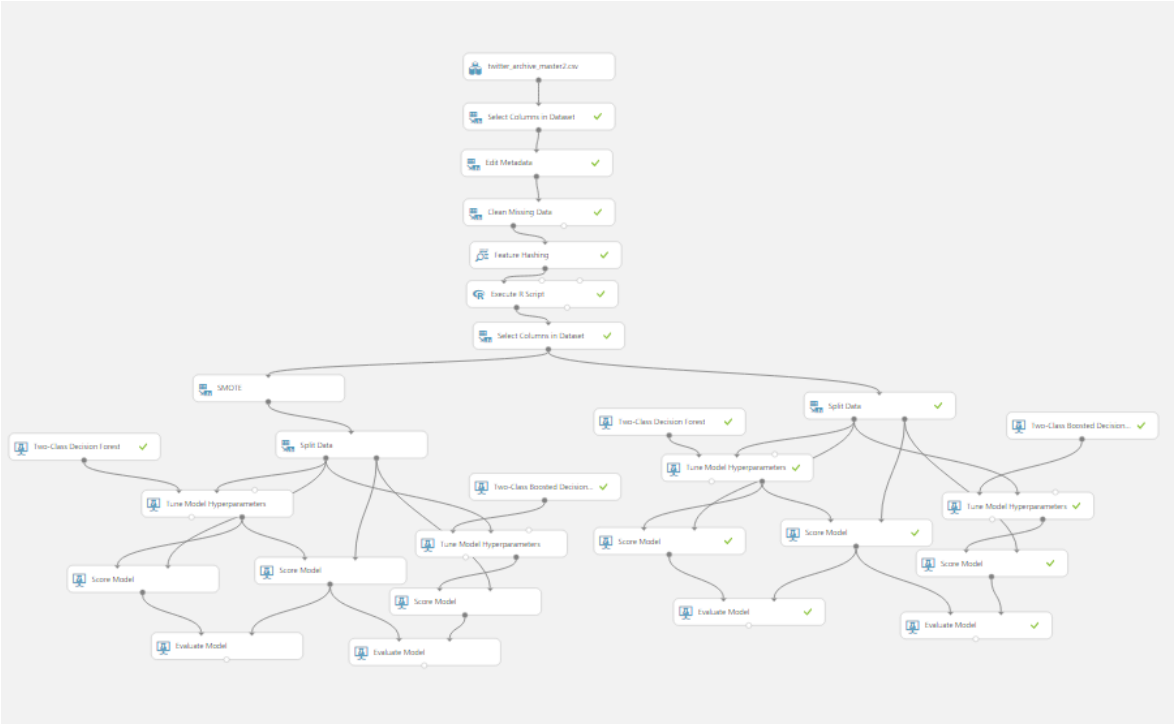

Figure 20. Overall experiment picture

Modelling started with uploading cleaned csv file into environment. According to results of exploratory data analysis, some variables were dropped and modelling continued with following variables. Also, retriever_flag feature stated as label.

Feature Name	Explanation
tweet_id	Tweet id information of tweet
source	Souce information of tweet
retweet_count	Number of retweet count belongs to tweet
favorite_count	Number of favorite count belongs to tweet
text	Tweet's text body
rating_numerator	Rating information of dog's. This feature extracted from text body
final_prediction	Dog's breed information predicted from picture of dog
new_dog_names	Dog's name information. This feature extracted from text body
stage	Dog's stage information. This feature extracted from text body
dog_gender	Dog's gender information. This feature extracted from text body
date	Date information of tweet
time	Time information of tweet
retriever_flag	Shows whether dog's breed retriever or not

Figure 21. Features' explanations

Missing data cleaned with replacing missing values with probabilistic PCA node. After cleaning was finished and data type of each features was controlled, feature-hashing node applied on text column to extract additional data from tweet's text body. With the

help of this node, 87 additional features were extracted. However, starting a model with these all variables lead to model to be overfitting. Therefore, doing dimension reduction was required at this time. Using a R code, 87 variables reduced to 10 variable with PCA to overcome overfitting. Same process duplicated with 40 variables; however, overfitting was observed means there was great differentiation in model performance between train and dataset. After this step of this project, 2 pipelines were determined according to sampling method. First one was continued with oversampling method due to dataset is low event portfolio. Second one continued without doing oversampling. Apart from oversampling methodology, same procedures were applied for these 2 pipeline. Data splitted into train and test datasets with stratified sampling and 0.5 fraction. Because both observation count and event count are so low, 0.5 ratio was determined for train-test splitting. With the help of Tune Model Hypermeters node, different parameter option has been tried for both two-class decision forest and two-class boosted decision tree algorithms.

In the first pipeline, oversampling methodology has been applied. With this aim, event rate increase 4.3 times increased, and population was prepared which have equivalent amount of event rate and non-event count. As explained at above, with Tune Model Hypermeters node two-class boosted decision tree and two-class decision forest models' parameters has been optimized. Optimized random forest algorithm shown in blue line whereas red line indicates the optimized decision tree on ROC curve. It can be clearly seen that random forest has greater performance compare to decision tree looking at area under ROC curve.

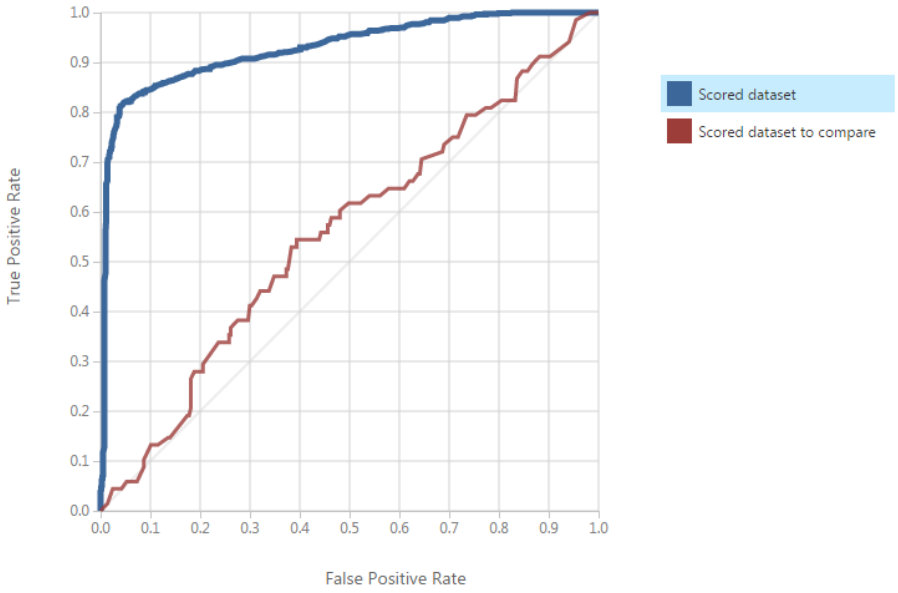

Figure 22. Comparision of random forest and decision tree for 1st pipeline

Figure 24 shows score bucket distributions of selected random forest model. When the threshold value were optimized; 0.89, 0.97, 0.82, 0.89 values are achieved for accuracy, precision, recall and F1 score respectively.

True Positive	False Negative	Accuracy	Precision	Threshold	AUC
628	142	0.897	0.968	0.56	0.948

False Positive	True Negative	Recall	F1 Score
21	790	0.816	0.885

Positive Label	Negative Label
1	0

Score Bin	Positive Examples	Negative Examples	Fraction Above Threshold	Accuracy	F1 Score	Precision	Recall	Negative Precision	Negative Recall	Cumulative AUC
(0.900,1.000]	0	0	0.000	0.513	0.000	1.000	0.000	0.513	1.000	0.000
(0.800,0.900]	0	0	0.000	0.513	0.000	1.000	0.000	0.513	1.000	0.000
(0.700,0.800]	15	0	0.009	0.522	0.038	1.000	0.019	0.518	1.000	0.000
(0.600,0.700]	391	2	0.258	0.769	0.689	0.995	0.527	0.690	0.998	0.001
(0.500,0.600]	274	124	0.510	0.863	0.863	0.844	0.883	0.884	0.845	0.130
(0.400,0.500]	83	365	0.793	0.685	0.754	0.608	0.991	0.979	0.395	0.554
(0.300,0.400]	7	163	0.901	0.586	0.702	0.541	1.000	1.000	0.194	0.754
(0.200,0.300]	0	39	0.925	0.562	0.690	0.526	1.000	1.000	0.145	0.802
(0.100,0.200]	0	118	1.000	0.487	0.655	0.487	1.000	1.000	0.000	0.948
(0.000,0.100]	0	0	1.000	0.487	0.655	0.487	1.000	1.000	0.000	0.948

Figure 23. Random forest probablity distiribution and treshold selection

After decided that random forest is the best model, I wanted to compare model performance on train and test dataset to understand there is any overfitting in the model. It can be seen from figure 25, model performance on train dataset is better than test dataset. However, performances are quite similar to each other like expected.

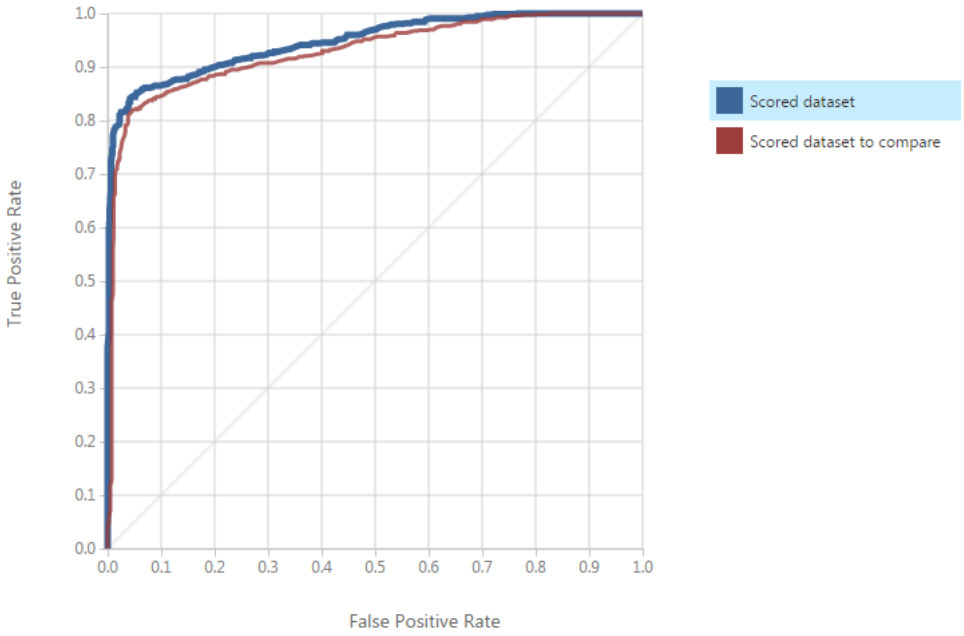

Figure 24. Train-Test comparision for random forest model

When we dive into second pipeline which was processed without using any special sampling methodology, it can be seen that random forest model still performs better than decision tree; however there are some problematic issues in the graph. We will understand why lines moves this way while looking at score distribution. In addition, it is nice to remember that train-test splitting and Tune Model Hypermeters node using are still same in this pipeline as well.

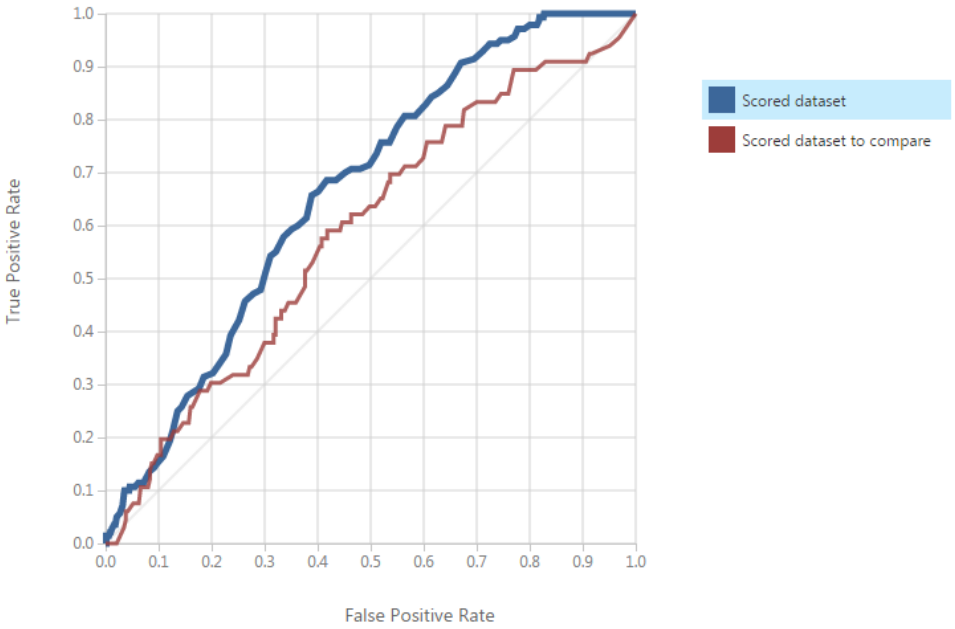

Figure 25. Comparision of random forest and decision tree for 2nd pipeline

When we look at score distributions of random forest, it can be seen that most of observations are summed in the 0.1-0.2 range. Therefore, it strongly shows that model cannot separate these observations which means that model cannot perform well. Even model has 0.77 accuracy ratio, recall and precision values are so bad in optimum threshold which as arranged by modeler.

True Positive	False Negative	Accuracy	Precision	Threshold	AUC
30	110	0.776	0.226	0.18	0.664

False Positive	True Negative	Recall	F1 Score
103	707	0.214	0.220

Positive Label	Negative Label
1	0

Score Bin	Positive Examples	Negative Examples	Fraction Above Threshold	Accuracy	F1 Score	Precision	Recall	Negative Precision	Negative Recall	Cumulative AUC
(0.900,1.000]	0	0	0.000	0.853	0.000	1.000	0.000	0.853	1.000	0.000
(0.800,0.900]	0	0	0.000	0.853	0.000	1.000	0.000	0.853	1.000	0.000
(0.700,0.800]	0	0	0.000	0.853	0.000	1.000	0.000	0.853	1.000	0.000
(0.600,0.700]	0	0	0.000	0.853	0.000	1.000	0.000	0.853	1.000	0.000
(0.500,0.600]	0	0	0.000	0.853	0.000	1.000	0.000	0.853	1.000	0.000
(0.400,0.500]	0	0	0.000	0.853	0.000	1.000	0.000	0.853	1.000	0.000
(0.300,0.400]	0	0	0.000	0.853	0.000	1.000	0.000	0.853	1.000	0.000
(0.200,0.300]	5	13	0.019	0.844	0.063	0.278	0.036	0.855	0.984	0.000
(0.100,0.200]	135	677	0.874	0.274	0.289	0.169	1.000	1.000	0.148	0.516
(0.000,0.100]	0	120	1.000	0.147	0.257	0.147	1.000	1.000	0.000	0.664

Figure 26. Random forest probablity distiribution and treshold selection

When we look at random forest model's performance in the train dataset, same situation also observe in this dataset as well.

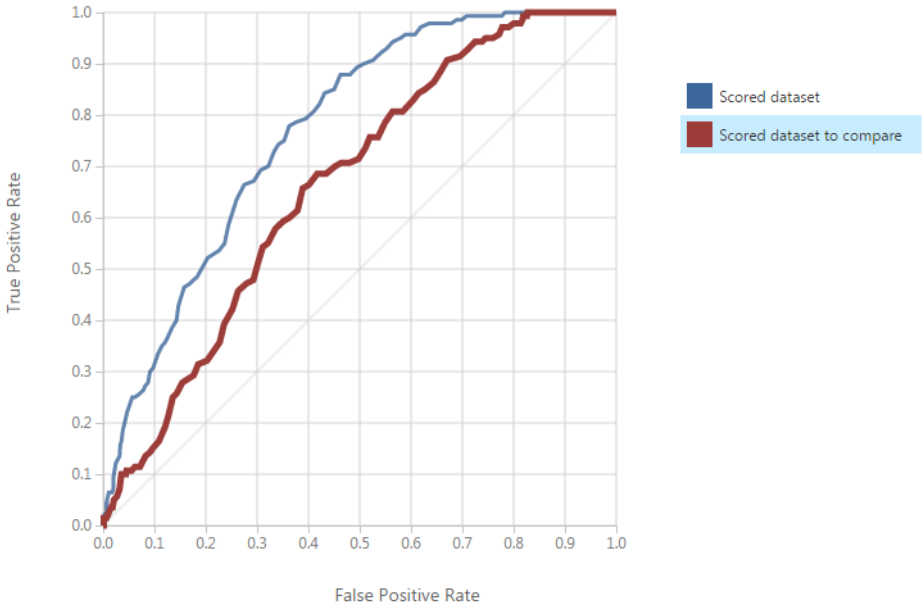

Figure 27. Train-Test comparision for random forest model

3. RESULT AND IMPROVEMENT POINTS

During this project, I realized that most important and time-consuming part was collecting and cleaning the data. Real-word data is mostly so untidy; therefore, there are many procedures to make data tidy and clean. Pyhton is the one of the great tool to gather, asses and clean the data. Also, Jupyter Notebook environment helps to document the project in easy and understandable format.

In addition, I realized that before dive into predictive modelling how EDA is important to understand data and gain insight from it. When it comes to predictive data analysis part, unsupervised learning algorithm as much as important as supervised learning. While using data extraction methodology, many variables are gathered. Most important dimensions are created with principle component analysis. Also, it is clearly seen that making oversampling helps to increase model performance significantly especially on the low event dataset as I used in this project. It has been observed that the model performance of random forest algorithm is clearly better than the model performance of decision tree.

Final but not least, I was only able to use two different supervised machine learning algorthm in this project; however, it is really important to try different machine learning algorithms such as neural networks, logistic regression, etc..

References

Astala, R., Ericson, G., Martens, J., & Petersen, T. (2018, 01 17). *https://docs.microsoft.com.* Microsoft Azure: https://docs.microsoft.com/en-us/azure/machine-learning/studio-module-reference/feature-hashing adresinden alındı

Astala, R., Ericson, G., Martens, J., & Takaki, J. (2018, 01 24). *Microsoft.* Microsoft Azure: https://docs.microsoft.com/en-us/azure/machine-learning/studio-module-reference/principal-component-analysis adresinden alındı

Dernoncourt, F. (2013, May 02). *Stackoverflow.* Stackoverflow: https://stackoverflow.com/questions/7370801/measure-time-elapsed-in-python adresinden alındı

Gayo-Avello, D. (2012, April 28). A Balanced Survey on Election Prediction using Twitter Data. Department of Computer Science - University of Oviedo, Spain.

HALKO, N., MARTINSSON, P., & TROPP, J. (2010, Dec 14). *FINDING STRUCTURE WITH RANDOMNESS:PROBABILISTIC ALGORITHMS FOR CONSTRUCTING APPROXIMATE MATRIX DECOMPOSITIONS.* Cornell University Library: https://arxiv.org/pdf/0909.4061.pdf adresinden alındı

Jim, E. (2015, 11 06). *Pyhton.* wiki.python.org: https://wiki.python.org/moin/HandlingExceptions adresinden alındı

Jolliffe, I. T., & Cadima, J. (2016, 04 13). *Principal component analysis: a review and recent developments.* NCBI: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC4792409/ adresinden alındı

Karampatziakis, N., & Mineiro, P. (2013, Oct 24). Combining Structured and Unstructured.

Matheson, A. (2017, 04 18). *Boston Magazine*. Boston Magazine: https://www.bostonmagazine.com/arts-entertainment/2017/04/18/dog-rates-mit/ adresinden alındı

Pieters, M. (2015, Feb 7). *stackoverflow*. stackoverflow: https://stackoverflow.com/questions/28384588/twitter-api-get-tweets-with-specific-id adresinden alındı

Robinson, S. (2016, 08 17). *Stackabuse*. Reading and Writing JSON to a File in Python: https://stackabuse.com/reading-and-writing-json-to-a-file-in-python/ adresinden alındı

Roesslein, J. (2018, July 03). tweepy Documentation.

Serrano, L. (2017, March 20). *Youtube*. Youtube: https://www.youtube.com/watch?v=2-Ol7ZB0MmU adresinden alındı

Shlens, J. (2015, December 10). A Tutorial on Principal Component Analysis. San Diego, La Jolla, CA 92093-0402.

SlickRemix. (2018). *SlickRemix*. https://www.slickremix.com/docs/how-to-get-api-keys-and-tokens-for-twitter/ adresinden alındı

Stein, B. (July 2005). Fuzzy-Fingerprints for Text-Based Information Retrieval. *Proceedings of the 5th International Conference on Knowledge Management (I-KNOW 05)* (s. 572–579). Graz: In Klaus Tochtermann and Hermann Maurer.

Stein, B., & Potthast, M. (2014, May 17). *Applying Hash-based Indexingin Text-based Information Retrieval*. Retrieved from ResearchGate: https://www.researchgate.net/publication/228543039