MEF UNIVERSITY

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

Capstone Project

Esra Arı

İSTANBUL, 2018

MEF UNIVERSITY

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

Capstone Project

Esra Arı

Advisor: Prof. Dr. Özgür Özlük

İSTANBUL, 2018

MEF UNIVERSITY

Name of the project: Wrangling WeRateDogs Twitter Data To Create Interesting And Trustworthy Exploratory / Predictive Analyses And Visualization Using Different Machine Learning Algorithms
Name/Last Name of the Student: Esra Arı
Date of Thesis Defense: 11/08/2018

I hereby state that the graduation project prepared by Esra Arı has been completed under my supervision. I accept this work as a "Graduation Project".

11/08/2018
Prof. Dr. Özgür Özlük

I hereby state that I have examined this graduation project Prof. Dr. Özgür Özlük which is accepted by his supervisor. This work is acceptable as a graduation project and the student is eligible to take the graduation project examination.

11/08/2018

Director
of
Big Data Analytics Program

We hereby state that we have held the graduation examination of __________ and agree that the student has satisfied all requirements.

THE EXAMINATION COMMITTEE

Committee Member	Signature
1. Prof. Dr. Özgür Özlük	………………………..
2. Dr. Tuna Çakar	………………………..

Academic Honesty Pledge

I promise not to collaborate with anyone, not to seek or accept any outside help, and not to give any help to others.

I understand that all resources in print or on the web must be explicitly cited.

In keeping with MEF University’s ideals, I pledge that this work is my own and that I have neither given nor received inappropriate assistance in preparing it.

Esra Arı	11.08.2018	
Name	Date	Signature

EXECUTIVE SUMMARY

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

Esra Arı

Advisor: Prof. Dr. Özgür Özlük

SEPTEMBER, 2018, 33 pages

Social media usage has rapidly grown in recent years and knowledge in these environments increased due to this expansion. Therefore, doing exploratory and predictive analysis from intensive data of social media became 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.

24 points increase (%34 change) in accuracy values has been achieved by doing oversampling in the data sets which contain low event observation. At the same time, the decision tree, logistic regression and random forest algorithms are compared and it is shown that the random forest's model performance is better than the others. The algorithm works 13 points better than logistic regression, 21 points better than decision tree.

Key Words: Text-Hashing, Data Wrangling, WeRateDogs, Machine Learning, Twitter Data, Principle Component Analysis, Random Forest, Decision Tree, Logistic Regression, Azure Machine Learning Studio

ÖZET

FARKLI MAKİNE ÖĞRENME ALGORİTMALARINI KULLANARAK WERATEDOGS TWITTER HESABININ VERİLERİNİN KEŞFEDİCİ VE TAHMINSEL ANALİZLERİNİN YAPILMASI VE GÖRSELLEŞTİRİLMESİ

Esra Arı

Tez Danışmanı: Prof. Dr. Özgür Özlük

Ağustos, 2018, 33 sayfa

Son yıllarda artan sosyal medya kullanımı, bu mecralardaki bilgi birikimi arttırmıştır. Artan bu bilgi yoğunluğu sosyal medyadan veri elde etmeyi ve bununla hem keşifçi hem de tahminsel analizler yapmayı popüler hale getirmiştir. Fakat elde edilen büyük verilerin neredeyse hepsi temizlenmemiş/ham veri durumundadır. Dolayısla verinin doğru bir şekilde temizlenmesi ve incelenmesi en az keşifçi ve tahminsel analizler kadar önemlidir Bu bitirme tezi için farklı kaynaklardan kirli veriyi toplamak, değerlendirmek, temizlemek, keşifçi ve tahminsel analizler yapmak amacı ile açık kaynaklı olan WeRateDogs twitter hesabının tweetleri kullanılmıştır. Yapılan çalışma sonucunda tahminsel veri analizinde aslında en önemli ve en çok zaman alan kısımın veriyi toplama ve temizleme olduğu anlaşılmıştır. Bu projenin çıktısı olarak sadece atılan tweet'in içerdiği yazı bilgisi ile köpeğin türünün retriever olup olmadığı tahminlenmiştir.

Yapılan tahminleme sürecinde düşük olay gözlemi içeren veri setlerinde fazladan örneklem yapılarak modelin doğruluk değerini 24 puan artması sağlanmıştır. Aynı zamanda karar ağacı, lojistik regresyon ve random forest algoritmaları karşılaştırılmış, random forest'ın model performansı açısından karar ağacı modellerinden iyi olduğu görüşmüştür. Bu doğrultuda random forest modeli karar ağacı modelinden 21 puan, lojistik regresyon modelinden ise 13 puan daha iyi doğruluk değeri almıştır.

Anahtar Kelimeler: Text-Hashing, Veri İnceleme, WeRateDogs, Makine Öğrenmesi, Twitter verisi, Princible Component Analizi, Random Forest, Karar Ağacı, Lojistik Regresyon, Azure Machine Learning Studio

TABLE OF CONTENTS

1. INTRODUCTION

Big data platforms became so popular due to increasing in diversity of used tools for big data management and accumulated data. 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 is rarely clean. Therefore, big data that gathered from variety of sources and in variety of formats should be assessed according to its quality and tidiness and cleaned using R, Python and their libraries. 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 named @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 and numerators of these ratings are greater than 10 such as 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.

WeRateDogs™ @dog_rates · Apr 16
Say hello to Alice. I'm told she enjoys car rides and smells good. 12/10 would give her everything she could ever want
WeRateDogs™ @dog_rates · Apr 16
A photographer took pictures before and after he told his bunny he's a good boy. Here are the results. 13/10

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 are used:

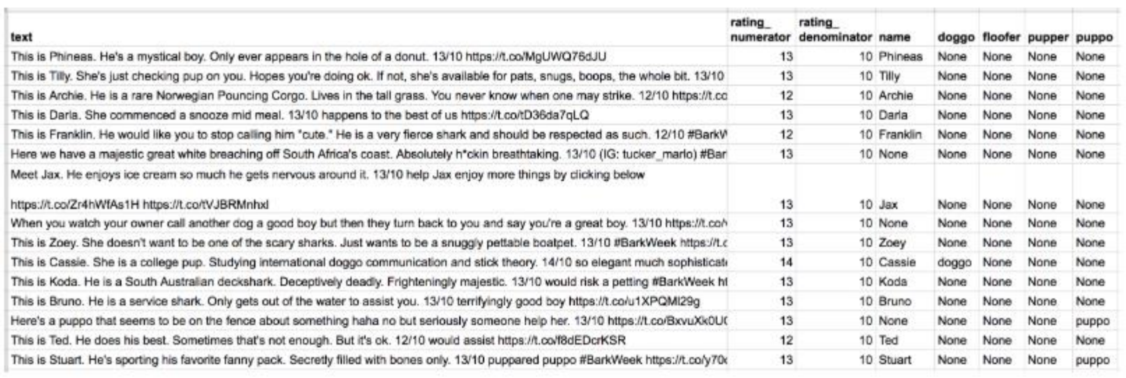
- Jupyter Notebook gives easy to understand documentation.
- The following packages (libraries) needed to be installed.
    - pandas
    - NumPy
    - requests
    - tweepy
    - json
    - re
    - seaborn
    - os
    - matplotlib.pyplot
- Microsoft Azure Machine Learning Studio

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, assessing and cleaning processes 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" 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

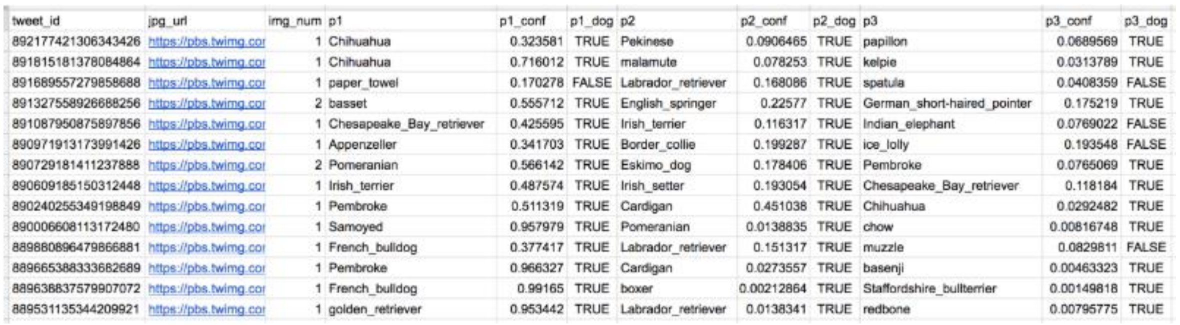
The data is programmatically extracted. Ratings are probably not fully correct.Same situation is valid for the dog names and probably dog stages too. Therefore, data is needed to be assessed and cleaned.

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 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

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.



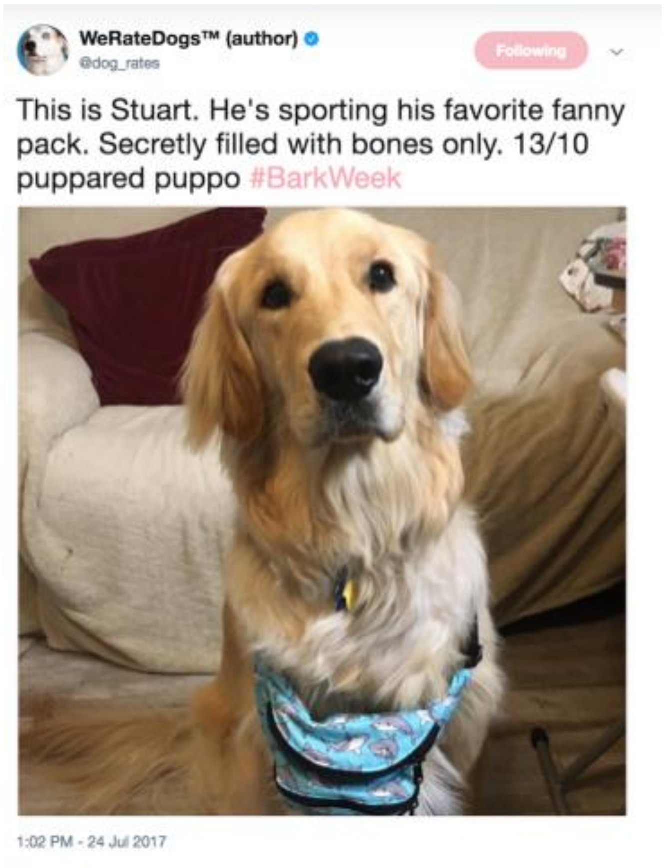
tweet_id	jpg_url	img_num	p1	p1_conf	p1_dog	p2	p2_conf	p2_dog	p3	p3_conf	p3_dog
892177421306343426	https://pbs.twimg.cor	1	Chihuahua	0.323581	TRUE	Pekinese	0.0906465	TRUE	papillon	0.0689569	TRUE
891815181378084864	https://pbs.twimg.cor	1	Chihuahua	0.716012	TRUE	malamute	0.078253	TRUE	kelpie	0.0313789	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:


WeRateDogs™ (author) @dog_rates

This is Stuart. He's sporting his favorite fanny pack. Secretly filled with bones only. 13/10 puppared puppo #BarkWeek

1:02 PM - 24 Jul 2017

Figure 4. Taken from Twitter:[2]

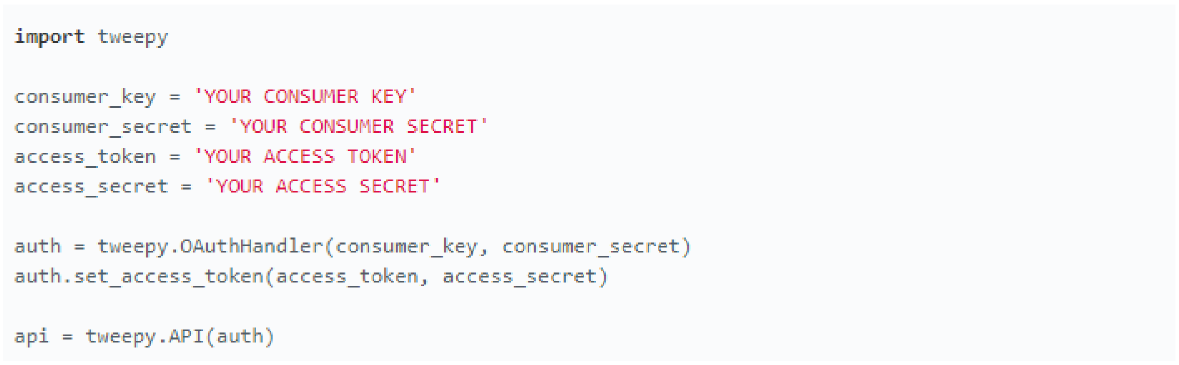
[2] Twitter. Retrieved from: https://twitter.com/dog_rates/status/889531135344209921

2. LITERATURE REVIEW

1.2 Twitter API Usage

In this project, Tweepy have been used to query Twitter's API for additional data beyond the data gathered from WeRateDogs Twitter archive. As it mentioned in introduction section, this additional data includes retweet count and favorite count.

Some APIs are completely open, like MediaWiki (accessed via the wptools library), others require authentication. The Twitter API is one that requires users to be authorized to use it. This means that before running API querying code, Twitter application is needed to be set up. And before that, a Twitter account must be signed up on behalf of an individual. In the reference section, the guide is given. Once all the setup is ready, the following code, which is provided in the references, the tweepy documentation, will create an API object that enables to gather Twitter data.



```
import tweepy

consumer_key = 'YOUR CONSUMER KEY'
consumer_secret = 'YOUR CONSUMER SECRET'
access_token = 'YOUR ACCESS TOKEN'
access_secret = 'YOUR ACCESS SECRET'

auth = tweepy.OAuthHandler(consumer_key, consumer_secret)
auth.set_access_token(access_token, access_secret)

api = tweepy.API(auth)
```

Figure 5. Phyton Code in Jupyter Notebook

Tweet data is stored in JSON format by Twitter. Getting tweet JSON data via tweet ID using Tweepy is described well in this StackOverflow answer. Note that setting the tweet_mode parameter to 'extended' in the get_status call, i.e., api.get_status(tweet_id, tweet_mode='extended'), can be useful. (Pieters, 2015)

1.2.1 Twitter's Rate Limit

Twitter's API has a rate limit. Rate limiting is used to control the rate of traffic sent or received by a server. In addition, Twitter indicates that rate limits are divided into 15 minute intervals as per Twitter's rate limiting info page is given in the references section.

In order to query all of the tweet IDs in the WeRateDogs Twitter archive takes 20-30 minutes of running time approximately. Printing out each tweet ID after it was queried and using a code timer given in the refences were both helpful for sanity reasons. Setting the wait_on_rate_limit and wait_on_rate_limit_notify parameters to True in the tweepy.api class is useful as well.

1.2.2 Writing and Reading Twitter JSON

After querying each tweet ID, its JSON data should be written to the required tweet_json.txt file with each tweet's JSON data on its own line. Then this file will be read, line by line, to create a pandas DataFrame that will be assessed and cleaned. Reading and Writing JSON to a File in Python article from Stack Abuse given in the references will be used for writing and reading Twitter Json data.

1.3 Feature Hashing

It is stated that one of powerful machine learning algorithm for similarity search, grouping and classification is hash-based indexing (Stein, Fuzzy-Fingerprints for Text-Based Information Retrieval, July 2005). It provides an efficient and reliable ways to overcome different retrieval tasks (Stein & Potthast, 2014).

In the Azure environment, feature-hashing node converts unique tokens into integers. It operates on the strings that does not perform any linguistic analysis or preprocessing and it only works with English strings.

According to Astala, Ericson, Martens, & Petersen, the advantage of using feature hashing is representing text documents of variable-length as numeric feature vectors of equal-length, and achieve dimensionality reduction. In contrast, if the text column is used for training as is, it would be treated as a categorical feature column, with many distinct values. Having the outputs as numeric also makes it possible to use many different machine learning methods with the data; including classification, clustering, or information retrieval. Because lookup operations can use integer hashes rather than string comparisons, getting the feature weights is also much faster (Astala, Ericson, Martens, & Petersen, 2018).

1.4 Principle Component Analysis

Big datasets have become popular in the recent years; however, they are also hard to explain, work and interpret. One of the technique of reducing the dimensionality of such dataset is Principal component analysis (PCA) which helps to increase interpretability but at the same time minimize information loss (Shlens, 2015). It creates new uncorrelated variables that successively maximize variance (Jolliffe & Cadima 2016).

In the Azure Machine Learning Studio, the Principal Component Analysis works by taking a set of feature columns in the provided dataset and creating a projection of the feature space that has lower dimensionality. Identifying a feature subspace that captures most of the information in the complete feature matrix is made by PCA algorithm while using randomization techniques. Therefore, reducing the effect the transformed data matrices help to reduce the effect of noise and minimize the risk of overfitting by capturing the variance in the original data (Astala R. , Ericson, Martens, & Takaki, 2018). To get more information about the PCA approaches used in module following articles can be investigated (HALKO, MARTINSSON, & TROPP, 2010) (Karampatziakis & Mineiro, 2013).

3. 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
    - Normalizing Data (if it is necessary)
    - Trying different supervised machine learning algorithms with different parameters (Random forest, logistic regression 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
- As stated in the appendix, clearly defined data munging and data analysis efforts are attached.

3.1 Gathering Data for Project

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 in 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.

3.2 Assessing Data for 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, each three pandas data frame gathered previous section will be investigated in this section. Assessing data has 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.

3.2.1 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 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.

3.2.2 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 "'&amp;" instead of "&".
  - invalid rating_denominator (different than 10). However, rating_demoniator checked manually and found that they are true denominators, so there is no problematic extraction from text.
  - Tweets_ids with no images however this problem will be solved with using image prediction dataset. Because, prediction which do not have any image is not expected.

- 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.
- Only original ratings (no retweets) that have images are needed.
- 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.

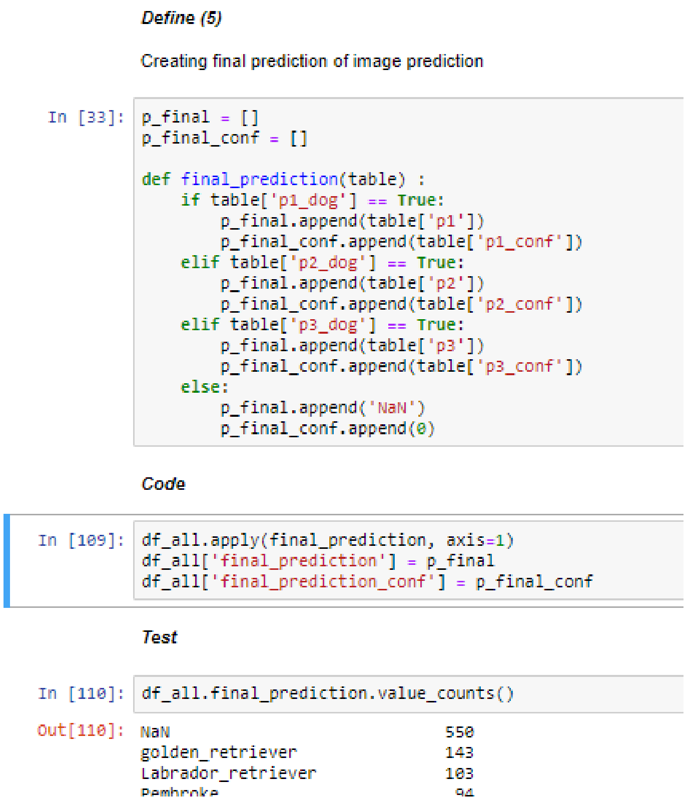
- 1222 numbers of id variable are duplicated
- id=666337882303524864 exits 4 times in the dataset with same results.
- lang indicates that the language of tweet. I wondered how "tl" lang is texted. Then, I realized id=668967877119254528 is problematic input.

Tidiness

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

3.3 Cleaning Data for Project

In this section, tidiness and quality issues were cleaned as mentioned before. 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 6. Cleaning example[3]

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

3.4 Storing

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

3.5 Exploratory Data Analysis

In the data wrangling part, data comes from three different sources is gathered, assessed and cleaned. 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 assumptions or intuitions about a data set are violated. Therefore, in this part, data is explored both quantitatively and visually. Also, it will be decided what it is going to be predicted 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 text, tweet information like number of retweeted, favorited, etc.

3.5.1 Uni-multi Variate Data Analysis

Before starting exploratory analysis, it is important to know features' explanations. Descriptions are given in the following.

- retweet_count: number of retweet count belongs to the tweet
- favorite_count: number of favorite count belongs to the tweet
- lang: language information of tweet
- created_at: tweet creation timestamp information
- tweet_id: tweet's ID which is the last part of the tweet URL after "status/": https://twitter.com/dog_rates/status/889531135344209921

- source: source information of tweet
- text: tweet’s text body
- expanded_url: tweet’s URL information
- rating_numerator: rating information of dog. This feature extracted from text body.
- rating_denominator: rating denomnator information of dog. This feature extracted from text body.
- name: name of dog. This feature extracted from text body.
- doggo: dog’s stage information which is most often used to denote a dog of medium to large size. This feature extracted from text body.
- floofer: informaton of whether dog is a very fluffy or not This feature extracted from text body.
- pupper: dog’s stage information which is most often used to denote a dog of smaller size. This feature extracted from text body.
- puppo: dog’s stage information which is most often used to denote a puppy or cute dog. This feature extracted from text body.
- jpg_url: tweet’s image URL information
- image_num: image number information
- p1: the algorithm's #1 prediction for the image in the tweet: golden retriever
- p1_conf: how confident the algorithm is in its #1 prediction: 95%
- p1_dog: whether or not the #1 prediction is a breed of dog: TRUE
- p2: the algorithm's second most likely prediction: Labrador retriever
- p2_conf: how confident the algorithm is in its #2 prediction: 1%
- p2_dog:whether or not the #2 prediction is a breed of dog: TRUE
- p3: the algorithm's third most likely prediction: Labrador retriever
- p3_conf: how confident the algorithm is in its #3 prediction: 1%
- p3_dog:whether or not the #3 prediction is a breed of dog: TRUE
- final_prediction: dog’s final breed information predicted from picture of dog. This feature created with p1, p2 and p3 features.
- final_prediction_conf: : how confident the algorithm is in dog’s final breed information predicted from picture of dog. This feature created with p1_conf, p2_conf and p3_conf features.

- new_dog_names: newly extracted dog's name information. It should be compared with name feature.
- dog_gender: gender informatio of dog. This feature extracted from text body.
- date: date information of tweet. This feature extracted from created_at feature.
- time: time information of tweet. This feature extracted from created_at feature.
- stage:dog's stage information. This feature created from doggo, floofer, pupper and puppo features.

Let's remember basic information about dataset

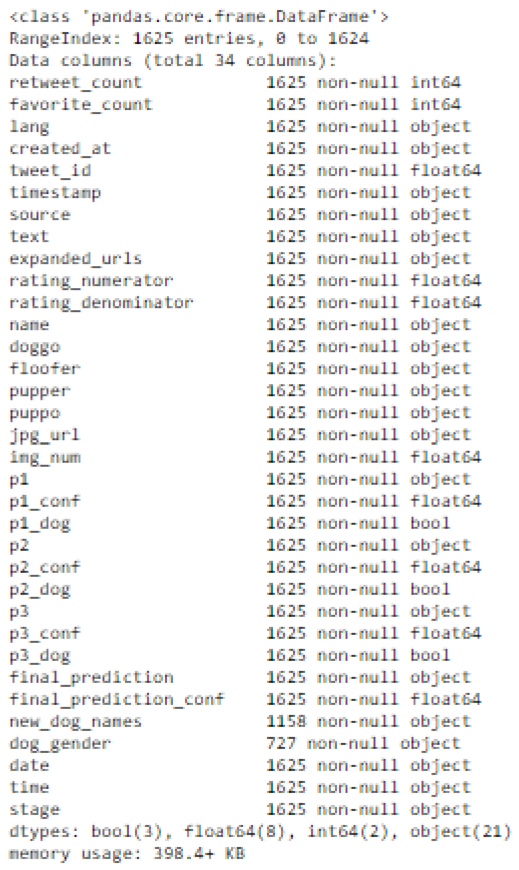
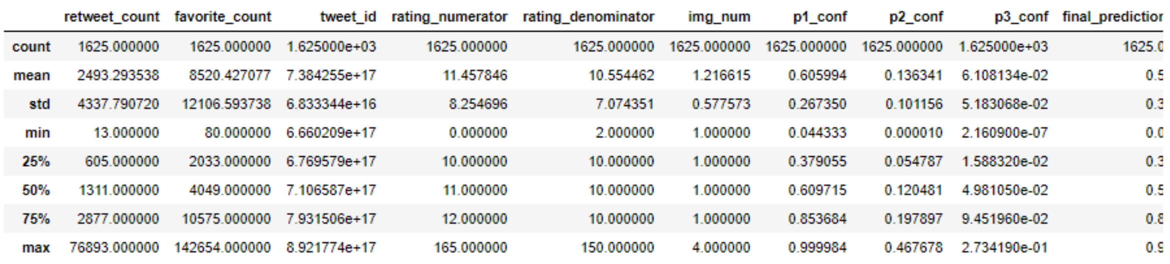

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1625 entries, 0 to 1624
Data columns (total 34 columns):
retweet_count            1625 non-null int64
favorite_count           1625 non-null int64
lang                     1625 non-null object
created_at               1625 non-null object
tweet_id                 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 7. Information about dataset

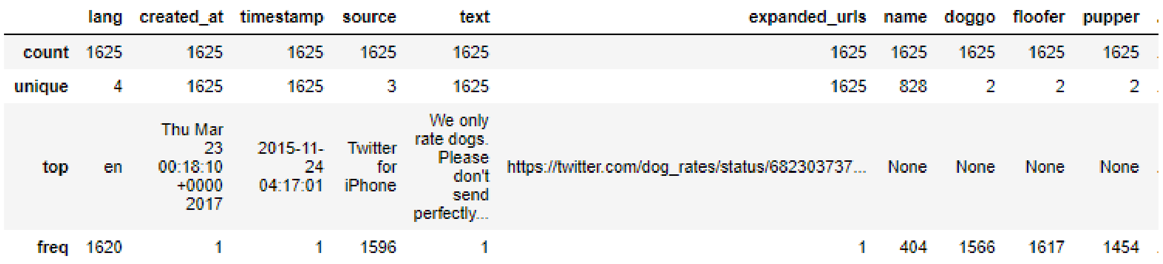
It can be seen from Figure 7., there are 31 feature and 1625 information after data cleaning. Dog_gender has highest missig ratio, secondly new_dog_names follows it.



	retweet_count	favorite_count	tweet_id	rating_numerator	rating_denominator	img_num	p1_conf	p2_conf	p3_conf	final_predictior
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 8. Descriptive Statistics of dataset

It can be seen from Figure 8., average number of retweet and favorite are 2493 and 8520. It can be say that tweeter users more pretend to favorite rather than retweet when IGR is investigated. Both rating numerator and denominator feature have absurd values like 160, 165 can be seen from max values. p1_conf, p2_conf and p3_conf will be dropped because, it has been already created one feature called final_prediction value shows final prediction of dogs' breed. Therefore, these variables will not be explored and excluded 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 9. Descriptive Statistics of dataset

Before analyzing features in detail, it is very useful to look all over to understand there is any abnormality that can affect overall analysis. As expected doggo, floofer, pupper features have two unique values; because, they are binary variables. Let's visualize features in detail.

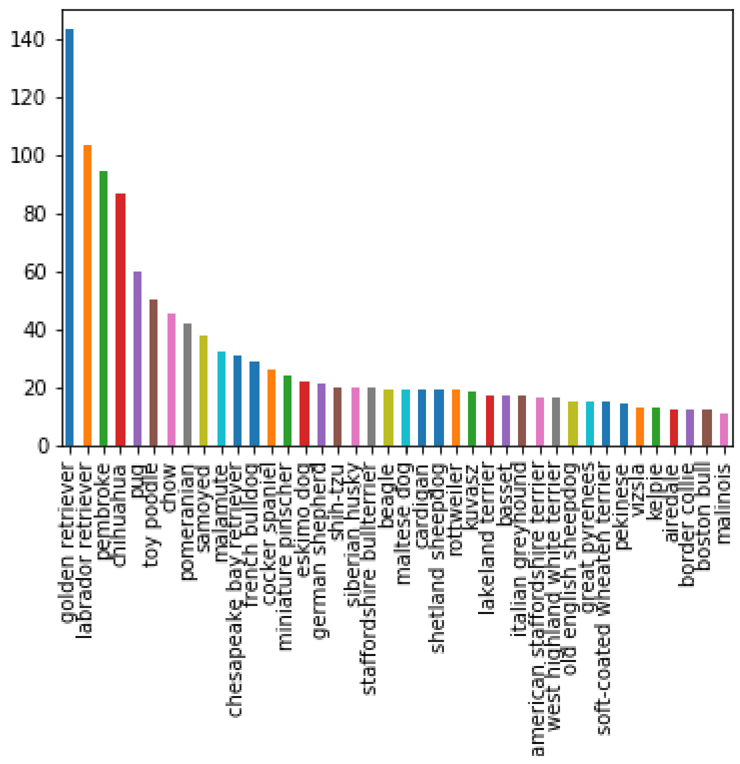


Figure 10. Investigation of predicted dogs' breed

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

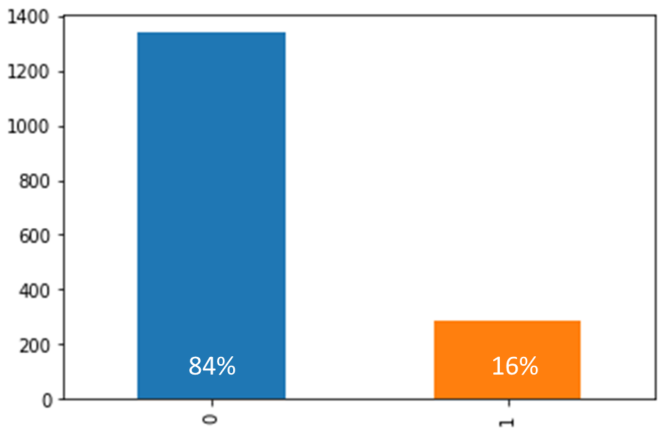


Figure 11. Retriever flag distribution

When the retrieved_flg feature is created from final_prediction, it can be seen that 16 percent of dogs belongs to retriever breed.

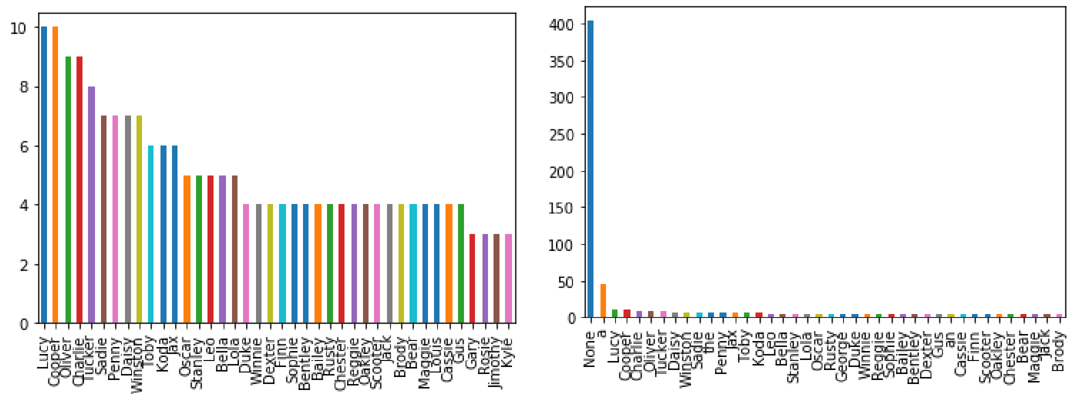


Figure 12. Comparing name and new named column created in the data wrangling part

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

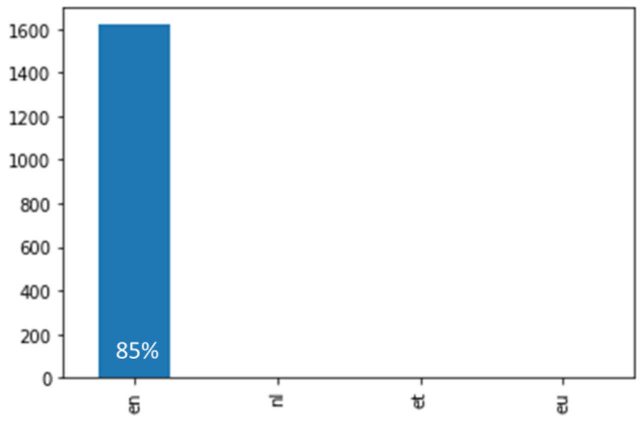


Figure 13. lang column investigation

Lang column gives the information about tweet language. It can be easily understood that most of tweets (85%) were written in English from the bar chart. Therefore, text-hashing option can be used in the further analysis during predictive analysis. In addition, this information is dropped because there is no info in it can be beneficial while doing prediction.

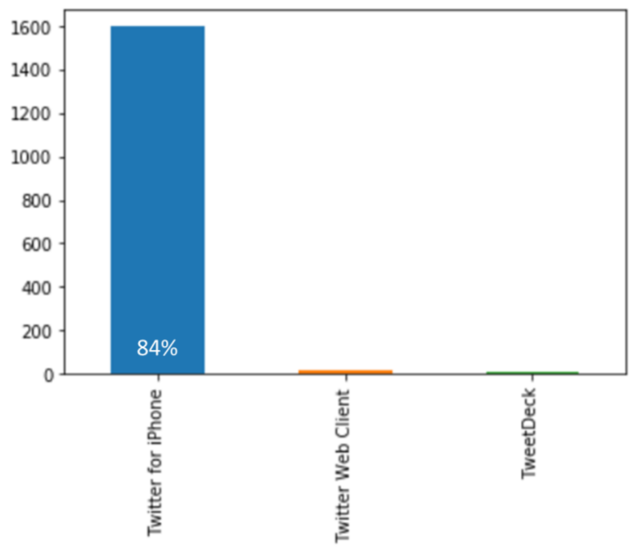


Figure 14. source column investigation

“source” column was extracted from url information column which gives 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.

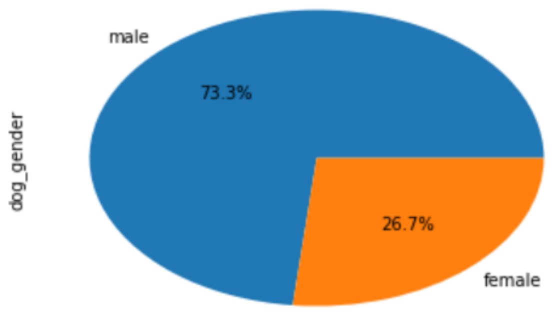


Figure 15. gender column investigation

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.

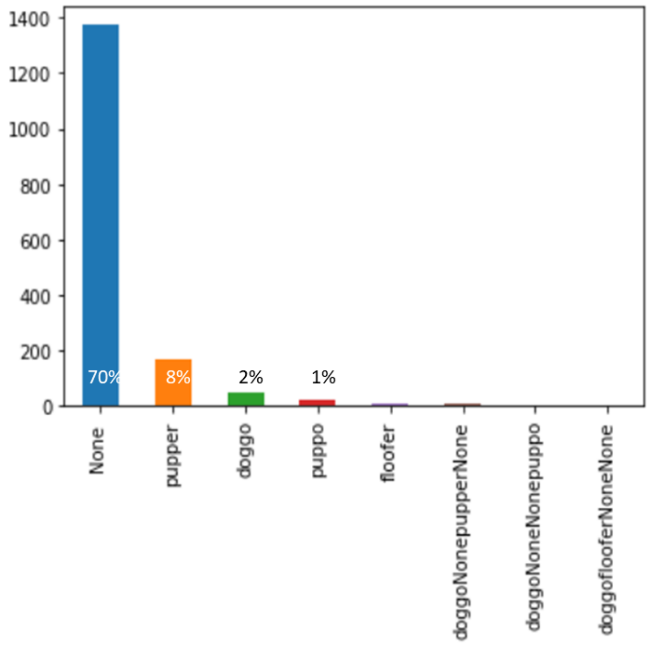


Figure 16. stage column investigation

"stages" column gives an information about dog's stage explained in detail at the beginning of chapter. However, most of tweets do not includes dog' stage information. Even if small number of information gives this information, still it is worth to use in the prediction.

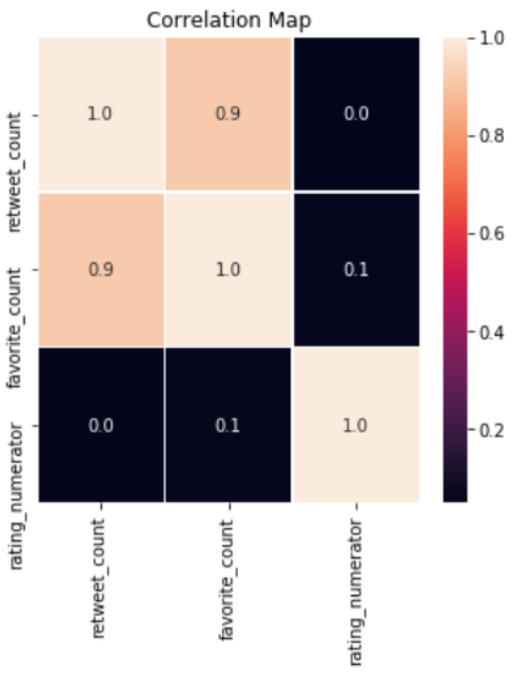


Figure 17. Correlation between numeric features

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, there is no relation between rating numerator gives dog' rating information. Therefore, it gives great intuition about ratings are quite objective, they cannot be target variable for the further analysis.

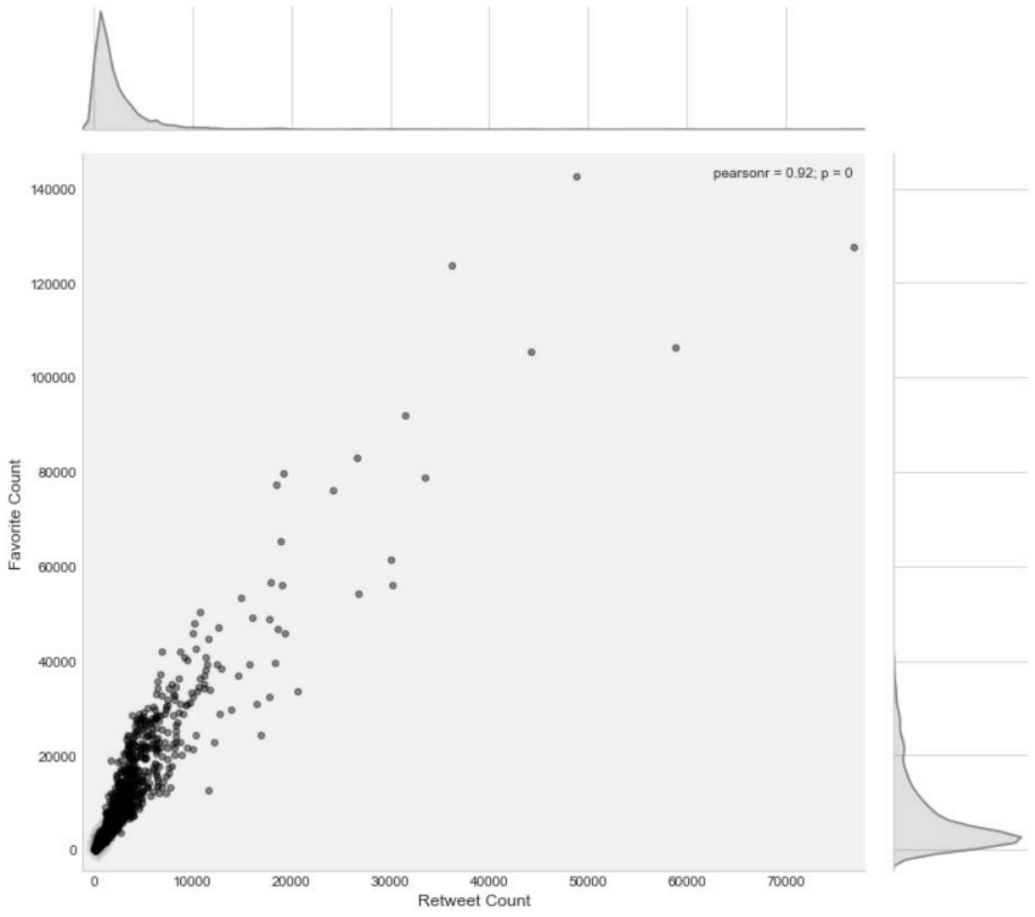


Figure 18. Correlation between retweet and favorite colums

When distribution of favorite and retweet counts inspected, 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.

As a result, %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. Tweeter users tend to favorite the tweet instead of retweet mostly. 16% percent of dogs belong to retriever breed. Also, there is no relationship between dog’s rating and favorite or retweet count.

3.6 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, which is 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 in the Figure 23.

- 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

- Normalizing data if it is necessary (applied for logistic regression)
- Building models using 3 different machine learning algorithms with different parameters optimizing them with Tune Model Hypermeters node (two-class decision forest, two-class boosted decision tree, two-class logistic regression)
- Scoring both train and test datasets
- Comparing results

Modelling started with uploading cleaned csv file into the Azure machine learning studio 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 19. Features' explanations

Missing data cleaned with replacing missing values with probabilistic PCA node. Especially, regression algorithms are not working with missing values; therefore, cleaning missing values one the most important part in modelling. 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 variables 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 test dataset. Therefore, moving forward with 10 variables was decided in this stage. After doing PCA, 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 pipelines. Data split into train and test datasets with stratified sampling and 0.5 fraction. Because both observation and event count are so low, 0.5 ratio was determined for train-test splitting. Before train the model, z-score transformation applied for transformation required algorithms like logistic regression. With the help of Tune Model Hypermeters node, different parameter option has been tried for two-class decision forest, two-class boosted decision tree and two-class logistic regression algorithms. All process summarized in the following Figure 20.

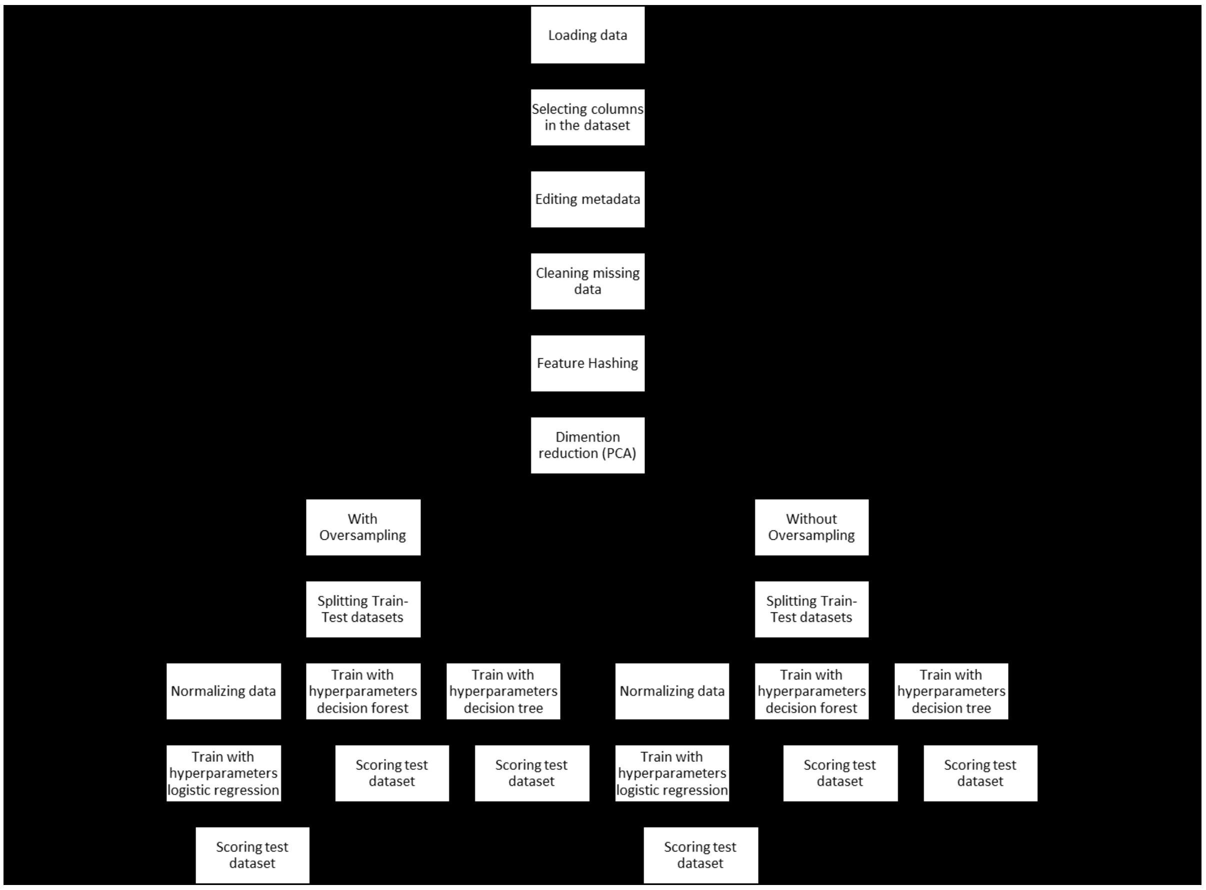
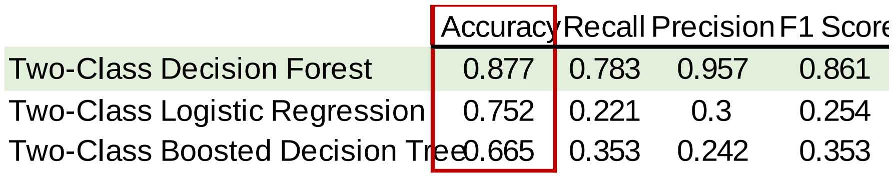

Loading data
Selecting columns in the dataset
Editing metadata
Cleaning missing data
Feature Hashing
Dimention reduction (PCA)
With Oversampling
Without Oversampling
Splitting Train-Test datasets
Splitting Train-Test datasets
Normalizing data
Train with hyperparameters decision forest
Train with hyperparameters decision tree
Normalizing data
Train with hyperparameters decision forest
Train with hyperparameters decision tree
Train with hyperparameters logistic regression
Scoring test dataset
Scoring test dataset
Train with hyperparameters logistic regression
Scoring test dataset
Scoring test dataset
Scoring test dataset
Scoring test dataset

Figure 20. Overall modelling process flow diagram

As it can be seen from Figure 20, project divided into 2 pipeline after dimension reduction step. In the first pipeline, oversampling methodology has been applied. With this aim, event rate increased 4.3 times which means that population was prepared as having equivalent amount of event rate and non-event count. As explained as above, with Tune Model Hypermeters node two-class boosted decision tree, two-class logistic regression and two-class decision forest models' parameters has been optimized. When the algorithms are ready, test data set is scored to compare the model performance as can be seen from Figure 21. With 0.87 accuracy value, two-class decision forest algorithm is the best model among these three algorithms. It also has meaningful recall, precision and F1 score value.



	Accuracy	Recall	Precision	F1 Score
Two-Class Decision Forest	0.877	0.783	0.957	0.861
Two-Class Logistic Regression	0.752	0.221	0.3	0.254
Two-Class Boosted Decision Tree	0.665	0.353	0.242	0.353

Figure 21. Models comparison

Before deciding decision forest is best algorithm for first pipeline, it is important to see ROC curves of these algorithms. Logistic regression and boosted decision tree have quite similar ROC curve; therefore, I only compared decision forest and tree algorithms' ROC curves. At Figure 22. It can be seen that ROC curve of optimized decision forest algorithm shown in blue line whereas red line indicates the optimized decision tree. It can be stated that random forest has greater performance compare to decision tree looking at area under ROC curve.

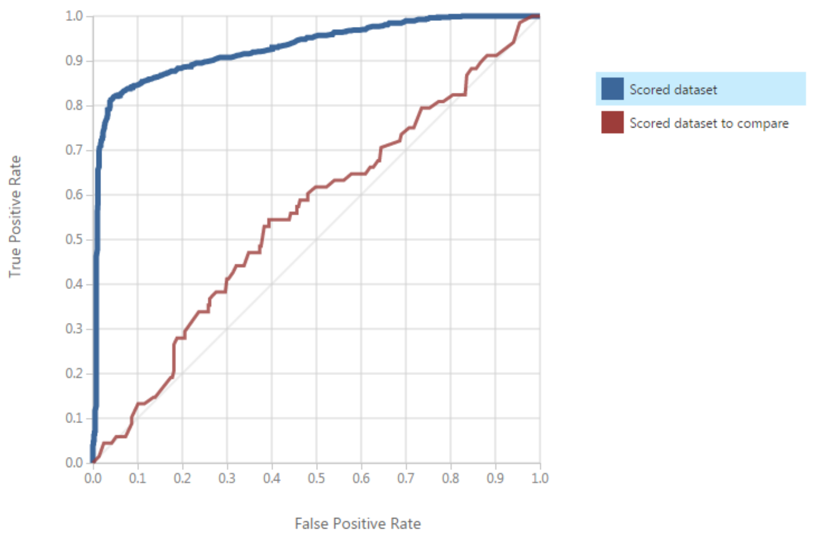


Figure 22. Comparison of decision forest and tree for first pipeline

After decided that random forest is the best model for first pipeline, it is important to observe and compare performance of model on both train and test dataset to control whether there is any overfitting in the model. Blue line represents performance on train data set, red line represents performance on test data set. As expected, blue line is always above the red line which means that performance of train data set is greater than performance on test data set. However, performances are quite similar to each other which proves no overfitting.

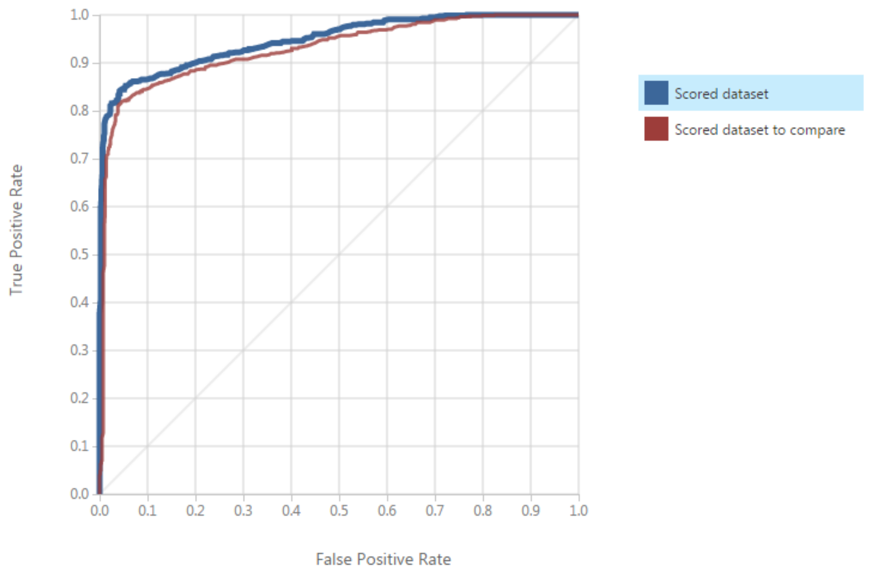
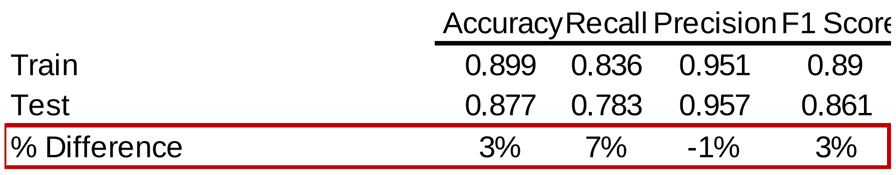


Figure 23. Train-Test comparision for random forest model

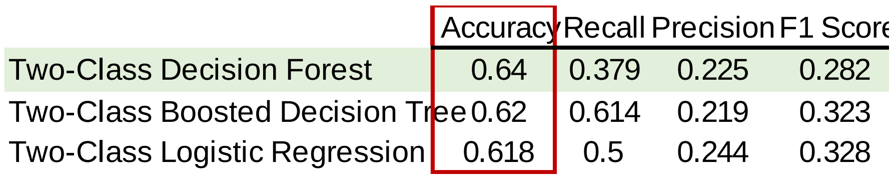
When model performance parameters compared on both test and train data set, from Figure 24. it can be seen that there is no significant difference which two points accuracy value decrease in test data set. (3% change)



	Accuracy	Recall	Precision	F1 Scor
Train	0.899	0.836	0.951	0.89
Test	0.877	0.783	0.957	0.861
% Difference	3%	7%	-1%	3%

Figure 24. Decision forest model comparison on oversampled data set

When the second pipeline examined which was processed without using any special sampling methodology, it can be seen that decision forest model still performs better than decision tree and logistic regression algorithms. However, there are very low recall, precision and F1 score is observed.



	Accuracy	Recall	Precision	F1 Score
Two-Class Decision Forest	0.64	0.379	0.225	0.282
Two-Class Boosted Decision Tree	0.62	0.614	0.219	0.323
Two-Class Logistic Regression	0.618	0.5	0.244	0.328

Figure 25. Models comparison

In addition to that, there are some problematic issues in the Figure 26. which indicates that there is no good classification of target. 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.

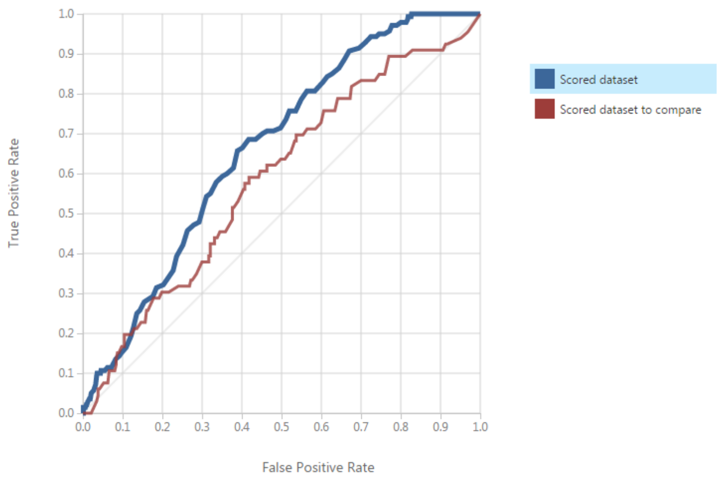
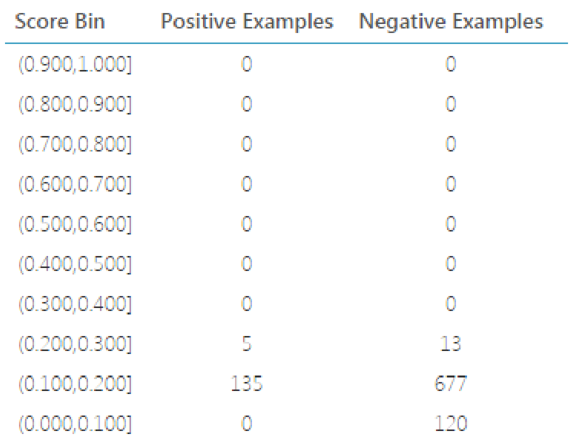


Figure 26. comparison of decision forest and tree for second pipeline

Score distributions of random forest investigated to analyze the problem in detail. It can be seen from the Figure 27, 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 0.64 accuracy ratio, recall and precision values are so bad in optimum threshold which as arranged by modeler.



Score Bin	Positive Examples	Negative Examples
(0.900,1.000]	0	0
(0.800,0.900]	0	0
(0.700,0.800]	0	0
(0.600,0.700]	0	0
(0.500,0.600]	0	0
(0.400,0.500]	0	0
(0.300,0.400]	0	0
(0.200,0.300]	5	13
(0.100,0.200]	135	677
(0.000,0.100]	0	120

Figure 27. Random forest probability distribution and threshold selection

When random forest model's performance examined in the train dataset, same situation also observed in train dataset as well at the Figure 25.

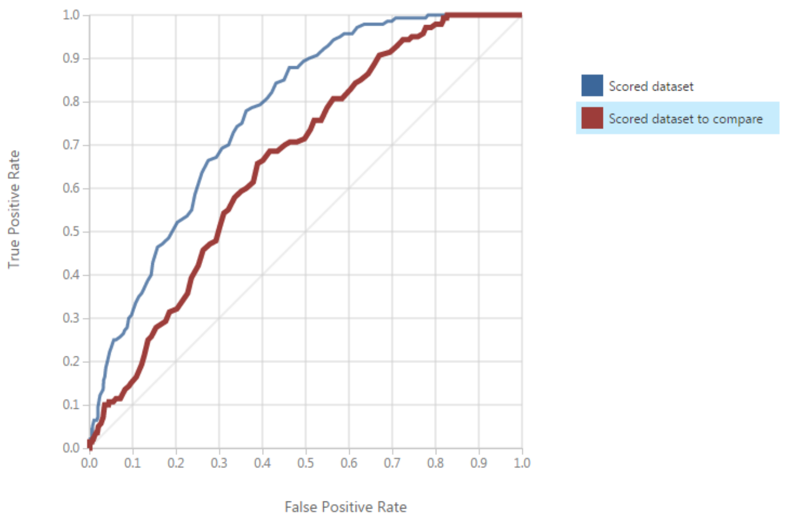


Figure 28. Train-Test comparison for decision forest model

In addition, model performance parameters were investigated, there are significant accuracy value difference in train and test data set which is 9 points decrease in test data set (%14 change) implies there is overfitting.

	Accuracy	Recall	Precision	F1 Scor
Train	0.731	0.586	0.293	0.39
Test	0.64	0.379	0.225	0.282
% Difference	14%	55%	30%	38%

Figure 29. Decision forest model comparison on train - test data set

To summarize, this exercise is a good example of how model performance can be increased by using oversampling methodology. Without oversampling 37%, 21% and %8 performance decreases have been observed at the Figure 30. with decision forest, logistic regression and decision tree algorithms respectively. It is also means that 37% accuracy value increase has been achieved with oversampling.

	Accuracy w/oversampling	Accuracy wo/oversampling	% Difference
Two-Class Decision Forest	0.877	0.64	37%
Two-Class Logistic Regression	0.752	0.62	21%
Two-Class Boosted Decision Tree	0.665	0.618	8%

Figure 30. With/out oversampling accuracy comparison

4. RESULT AND IMPROVEMENT POINTS

Most important and time consuming part of the project was collecting and cleaning the data. Real-word data is mostly untidy; therefore, there are many procedures to make data tidy and clean. Python 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 generated from the data which has limited number of features. Most important dimensions are created with principle component analysis which increases the model performance. Also, it is clearly seen that making oversampling helps to 24 points (%34 change) increase in model performance especially on the low event dataset as it is used in this project. It has been observed from Figure 34. that the model performance of random forest algorithm is clearly better than the model performance of decision tree and logistic regression for both two pipelines. It is important to keep in mind that random forest is like bootstrapping algorithm with Decision tree model which means that random forest tries to build multiple decision tree model with different sample and different initial variables. Therefore, random forest gives much higher model performance when compared to simple decision tree or regression models in many scenarios; because, it captures the variance of several input variables at the same time and enables high number of observations to participate in the prediction.

	Accuracy w/oversampling	Accuracy wo/oversampling
Two-Class Decision Forest	0.877	0.64
Two-Class Logistic Regression	0.752	0.62
Two-Class Boosted Decision Tree	0.665	0.618

Figure 34. With/out oversampling accuracy comparison

Final but not least, three different supervised machine learning algorithm used in this project; however, it is really important to try different machine learning algorithms such as neural networks, support vector machine 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