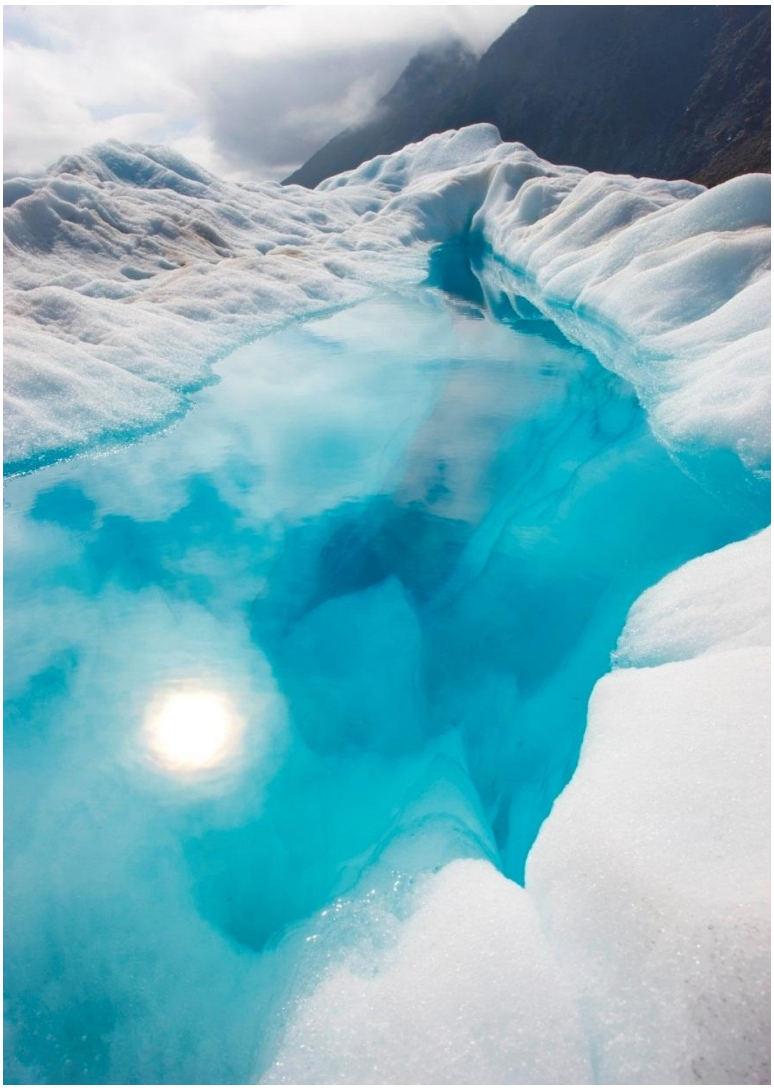

Classify Tweet by gender report

MACHINE LEARNING ASSIGNMENT 2020

Phillip Madi(1637384) Alec Mbanga(1653526) Tshegofatso Aphane(1643694)
Nhlanhla Mpele(1860226) Mahlatsi Morapedi(1882334)

| COMS3007A - Machine Learning |
| June 24, 2020 |

Github repo: https://github.com/tlaphane/Machine-Learning---Twitter-User-Gender-Classification

STEP 1: DESCRIPTION OF THE DATASET

This data is downloaded from the site https://www.kaggle.com/crowdflower/twitter-user-gender-classification.

- What are the attributes? What values do they take on?

The dataset contains the following fields:

- **unitid**: a unique id for user
- **_golden**: whether the user was included in the gold standard for the model; TRUE or FALSE
- ***unit*state**: state of the observation; one of *finalized* (for contributor-judged) or *golden* (for gold standard observations)
- ***trusted*judgments**: number of trusted judgments (int); always 3 for non-golden, and what may be a unique id for gold standard observations
- ***last*judgment_at**: date and time of last contributor judgment; blank for gold standard observations
- **gender**: one of *male*, *female*, or *brand* (for non-human profiles)
- **gender:confidence**: a float representing confidence in the provided gender
- **profile_yn**: "no" here seems to mean that the profile was meant to be part of the dataset but was not available when contributors went to judge it
- **profile_yn:confidence**: confidence in the existence/non-existence of the profile
- **created**: date and time when the profile was created
- **description**: the user's profile description
- **fav_number**: number of tweets the user has favorited
- **gender_gold**: if the profile is golden, what is the gender?
- **link_color**: the link color on the profile, as a hex value
- **name**: the user's name
- **profile*yn*gold**: whether the profile y/n value is golden
- **profileimage**: a link to the profile image

- **retweet_count**: number of times the user has retweeted (or possibly, been retweeted)
- **sidebar_color**: color of the profile sidebar, as a hex value
- **text**: text of a random one of the user's tweets
- **tweet_coord**: if the user has location turned on, the coordinates as a string with the format "[*latitude*, *longitude*]"
- **tweet_count**: number of tweets that the user has posted
- **tweet_created**: when the random tweet (in the text column) was created
- **tweet_id**: the tweet id of the random tweet
- **tweet_location**: location of the tweet; seems to not be particularly normalized
- **user_timezone**: the timezone of the user

- **What are the targets?**

- The targets are the **gender**: one of *male*, *female*, or *brand* (for non-human profiles).
- We are predicting how well do words in tweets and profiles predict user gender?

- **How many data points do you have?**

- The whole dataset contains **20050** data points

- Sample data points (4) from the dataset:

	_unit_id	_golden	_unit_state	_trusted_judgments	_last_judgment_at	gender	gender:confidence	profile_yn	profile_yn:confidence	created	...
0	815719226	False	finalized	3	10/26/15 23:24	male	1.0000	yes	1.0	12/5/13 1:48	...
1	815719227	False	finalized	3	10/26/15 23:30	male	1.0000	yes	1.0	10/1/12 13:51	...
2	815719228	False	finalized	3	10/26/15 23:33	male	0.6625	yes	1.0	11/28/14 11:30	...
3	815719229	False	finalized	3	10/26/15 23:10	male	1.0000	yes	1.0	6/11/09 22:39	...
4	815719230	False	finalized	3	10/27/15 1:15	female	1.0000	yes	1.0	4/16/14 13:23	...

...	profileimage	retweet_count	sidebar_color	text	tweet_coord	tweet_count	tweet_created	tweet_id	tweet_location	user_timezone
...	https://pbs.twimg.com/profile_images/414342229...	0	FFFFFF	Robbie E Responds To Critics After Win Against...	NaN	110964	10/26/15 12:40	6.587300e+17	main; @Kan1shk3	Chennai
...	https://pbs.twimg.com/profile_images/539604221...	0	C0DEED	ÛÏIt felt like they were my friends and I was...	NaN	7471	10/26/15 12:40	6.587300e+17	NaN	Eastern Time (US & Canada)
...	https://pbs.twimg.com/profile_images/657330418...	1	C0DEED	i absolutely adore when louis starts the songs...	NaN	5617	10/26/15 12:40	6.587300e+17	clcncl	Belgrade
...	https://pbs.twimg.com/profile_images/259703936...	0	C0DEED	Hi @JordanSpieth - Looking at the url - do you...	NaN	1693	10/26/15 12:40	6.587300e+17	Palo Alto, CA	Pacific Time (US & Canada)
...	https://pbs.twimg.com/profile_images/564094871...	0	0	Watching Neighbours on Sky+ catching up with t...	NaN	31462	10/26/15 12:40	6.587300e+17	NaN	NaN

- State what you are trying to predict with the data?

- We are predicting how well do words in tweets and profiles predict user gender?

STEP 2: DATA PROCESSING AND VISUALISATION.

- How was the data pre-processed?

In NewProcessedData.ipynb:

- Replaced missing values with the most frequent value on that feature.
- Changed continues values to discrete categorical values.

In ProcessedData.ipynb:

- Creation of bag of words.
- Got top 4000 words which will act as our features of each sentence.
- Frequency of our features in each datapoint.
- Creation of Sparse Matrix, Design Matrix.

- How were the inputs/targets normalised?

- How was the data split into training/validation/test data?

- The full set of the data was split into 3/5 for **training** and 1/5 for **validation** and 1/5 for **test** data.

STEP 3: ALGORITHM IMPLEMENTATION

Logistic Regression:

Logistic regression is one of the quickest algorithm. We can apply batch gradient descent to the problem of finding the coefficients for the logistic regression model as follows:

Given each training instance:

1. Calculate a prediction using the current values of the coefficients. Learning rate 0.2 and we do 500 iterations
2. Calculate new coefficient values based on the error in the prediction.

The process is repeated until the model is accurate enough (e.g. error drops to some desirable level) or for a fixed number iterations. You continue to update the model for training instances and correcting errors until the model is accurate enough and cannot be made any more accurate. It is often a good idea to randomize the order of the training instances shown to the model to mix up the corrections made.

Naïve Bayes:

Naïve Bayes is one of the most accurate algorithms. The classifier is given with some features (being a male or female) and it must decide if tweet a male or female (these are two classes). It uses Bayes theorem formula to calculate the probability of each class under the features, and it assigns the class with the highest probability to the input. To provide the required probabilities (90%, 10%, 80%...) it uses the training set. For example, it counts the people in the training set that are males and find they contribute 10% of the sample. In the other words, it tries to build the probability distribution of the features for each class based on the training data

Neural Networks:

We choose the Neural Network because is deals better with nonlinearity. Our data deals with strings that are converted into numerical values. For some given features that represent the top 100 most occouring words in our data, the algorithm will classify for each user whether its a male or a female. The Implementation is at its simplest form which takes the the dataset (this dataset contains features and number of users in matrix form) as its input, then we created a bias vector and weight vector which needs to be updated for a given number of epoches. We choice the epoches to iterate 1000 because is its the default number generally used, however we could have iterated more than that to train our dataset, though it could have taken longer. As stated that the algorithm is at its simplest form, we have chosen that it should just contains only one hidden layer. The learning rate we chose best sits between overfitting and underfitting as we played around values which best fits our data. The problems we encountered was updating our weights simultaneouly thus we found that our testing accuracy decreased due to the small mumber of data points and simultaneously updating our weights. Our main priority was seeing the error sum for each iteration decreasing thus it decreases in a sequential form yet it the results are effective to the weights.
The algorithm learns from our data effectively and we are confident enough that the algorithm does better than other tested algorithms.

Decision trees:

Started by changing features that were continuous and made them to be discrete. Features that could not be grouped into small discrete values were deleted because they were slowing down the computation of the algorithm. Then by starting the algorithms computation we calculated the entropy and the gain of each feature then we split the data base on the feature with maximum gain. We continued doing this until the depth of the tree was 6 or if branched data in a tree had the same target value

STEP 4: A BRIEF DISCUSSION AND ANALYSIS OF RESULTS FROM THE VARIOUS ALGORITHMS.

- Some features had continuous values and I changed those features to have discrete values because most classification algorithms can easily work with discrete values and it will also help to increase accuracy. For example in our dataset you can see that the "gender:confidence" feature takes any continues value between 0 and 1, so to make it categorical we added a "discrete-gender:confidence" feature which grouped continuous values between 0-0.1 to 'gender:confidence_1', 0.1-0.2 to 'gender:confidence_2', 0.2-0.3 to 'gender:confidence_3',....., 0.9-1 to 'gender:confidence_9'. So now we have just 9 categories inside gender:confidence instead of thousands of continuous values between 0 and 1. So on our algorithms we can either use continued 'gender:confidence' or 'discrete-gender:confidence'. In most of our algorithms, dividing the data set into training and testing data sets produced low accuracies and we introduced validation dataset and the accuracy increased. The ratio 60%, 20%, 20% was used to split the data into training, testing and validating datasets respectively.
- The Neural Network seems to have performed the best.

1. Decision Tree Results:

   At first the computation would take longer (>1hour) to compute results and after removing features that had many unique values and that could not be grouped into small discrete values then the algorithm started computing faster.

Model Accuracy:

```
Decision Tree Algorithm classification accuracy on Training data 45.35680529300567
Decision Tree Algorithm classification accuracy on Validation data 47.8908188585608
Decision Tree Algorithm classification accuracy on Testing data 41.445270988310305
Confusion Matrix:
```

	pred female, male
Actual female	[12, 1644]
Actual male	[9, 1158]

2. **Neural Networks:**

   Model Accuracy:

```
Neural Network classification accuracy on Training data: 59.5108695652174
Neural Network classification accuracy on Validation data: 62.934089298369955
Neural Network classification accuracy on Testing data: 59.000708717221826
Confusion Matrix:
               pred female, male
Actual female       [1219, 437]
Actual male          [720, 446]
```

3. **Logistic Regression**:

Model Accuracy:

```
Logistic Regression classification accuracy on Training data 54.21786389413988
Logistic Regression classification accuracy on Validation data 58.32742735648476
Logistic Regression classification accuracy on Testing data 58.64635010630759
Confusion Matrix:
              pred female, male
Actual female         [1653, 3]
Actual male           [1164, 2]
```