MAJOR PROJECT SUMMARY

INTRODUCTION

- For the given twitter gender classification dataset we have to find the best classification algorithm.
- The Dataset contains the following fields:

1. **_unit_id**: a unique id for user
2. **_golden**: whether the user was included in the gold standard for the model; TRUE or FALSE
3. **_unit_state**: state of the observation; one of finalized (for contributor-judged) or golden (for gold standard observations)
4. **_trusted_judgments**: number of trusted judgments (int); always 3 for non-golden, and what may be a unique id for gold standard observations
5. **_last_judgment_at**: date and time of last contributor judgment; blank for gold standard observations
6. **gender**: one of male, female, or brand (for non-human profiles)
7. **gender:confidence**: a float representing confidence in the provided gender
8. **profile_yn**: "no" here seems to mean that the profile was meant to be part of the
9. dataset but was not available when contributors went to judge it
10. **profile_yn:confidence**: confidence in the existence/non-existence of the profile
11. **created**: date and time when the profile was created
12. **description**: the user's profile description
13. **fav_number**: number of tweets the user has favorited
14. **gender_gold**: if the profile is golden, what is the gender?
15. **link_color**: the link color on the profile, as a hex value
16. **name**: the user's name
17. **profile_yn_gold**: whether the profile y/n value is golden
18. **profileimage**: a link to the profile image
19. **retweet_count**: number of times the user has retweeted (or possibly, been retweeted)
20. **sidebar_color**: color of the profile sidebar, as a hex value
21. **text**: text of a random one of the user's tweets
22. **tweet_coord**: if the user has location turned on, the coordinates as a string with the format "[latitude, longitude]"

23. **tweet_count**: number of tweets that the user has posted
24. **tweet_created**: when the random tweet (in the text column) was created
25. **tweet_id**: the tweet id of the random tweet
26. **tweet_location**: location of the tweet; seems to not be particularly normalized
27. **user_timezone**: the timezone of the user

Data cleaning

- We took the data originally given to us which is regarding the information of tweets.
- Then we extracted the following columns from the datast: ('_unit_id', 'gender', 'gender:confidence', 'description', 'fav_number', 'retweet_count', 'text', 'tweet_count').
- We cleaned the text and description columns using stopwords from nltk.corpus and removed meaningless words .
- Before making the model we dropped the 'unknown' values from gender.
- Data cleaning was based on the fact that we have to determine the gender of the user therefore we have to keep all the data in which the probability of being a particular gender is 100%.

Data visualization and questions

- For knowing the most used word by each gender we must convert lines in text column to list of words. For that we are using cleaning function.
- We will convert all lines in text column to list of words using cleaning method.
- We will save list of words in column named tweets.

- Then three separate data frames are created for males, females and brands.
- Using those we will count the frequency of each word according to the gender.
- Then we will plot bar graph for each gender with words and their counts (only top 20 words).
- From that we can answer 1st question.
- Then graph is plotted to know the counts of each gender.
- For answering 2nd question we have divided all the tweet text in separated words and stored them in a list after that we have used textblob.
- Textblob is used to correct the spelling which we pass to it and we have used this functionality to check weather my passed string and corrected string is same or not then if they are same spelling is correct no typo done by user and if they are not then spelling is incorrect that this gender has created a typo so we have made another data frame consisting of words and corresponding correctness of word, both for male and female.

CLASSIFICATION ALGORITHMS

- We chose 3 classification algorithms which are **Logistic Regression, SVM** and **Random Forest classifier.**
- We trained the model using train_test_split from sklearn.model_selection assigning 'text' to x and 'gender' to y the above mentioned three algorithms and calculated the accuracy scores.
- We compared the accuracy scores of all the three machine learning models. The accuracy scores are:

  1. Logistic Regression – 59.82%

2. SVM – 59.72%
3. Random Forest Classifier – 57.2%

- Now adding content of description into text(concatenating 'description' to 'text') we recreated the training model which gave the following results.
- We compared the accuracy scores of all the three machine learning models. The accuracy scores are:
    1. Logistic Regression – 68.13%
    2. SVM – 68.54%
    3. Random Forest Classifier – 64.64%

    From the accuracy scores we can tell that SVM suits the best for the given problem.