ML Major Project

**Summary of the project**

The aim of the project is gender classification from tweets on social media by building an ensemble model by using any 3 classification algorithms. Now we will look into the steps that were done while making this project.

The following was done in the project

1) Exploratory data analysis (with visualization) and Data Cleaning

2) Feature Selection and text pre processing

3) Ensemble Machine learning Model creation (3 Classification Algorithms)

4) Accuracy calculation

5) Testing the dataset with inbuilt ensemble models from dataset

Important columns are :- gender, description , text

Where gender is dependant variable and (description , text) are independent variables

**Exploratory data analysis (with visualization) and Data Cleaning**

In this step EDA was performed on the dataset and also two questions were asked on the dataset. Firstly the rows that have the target variable (gender) as male and female were only selected. Also the rows that do not have gender:confidence value as 1 were removed. Later the following questions were asked on the dataset. The dataframe with gender as male were stored in df1 and dataframe with gender as male were stored in df2

Q1) What are the most common emotions/words used by Males and Females?

Q2) Which gender makes more typos in their tweets?

To solve the following questions firstly the dataset was cleaned by removing all the punctuations in the text and stopwords were removed . The data for gender male is stored in Male and for females is stored .

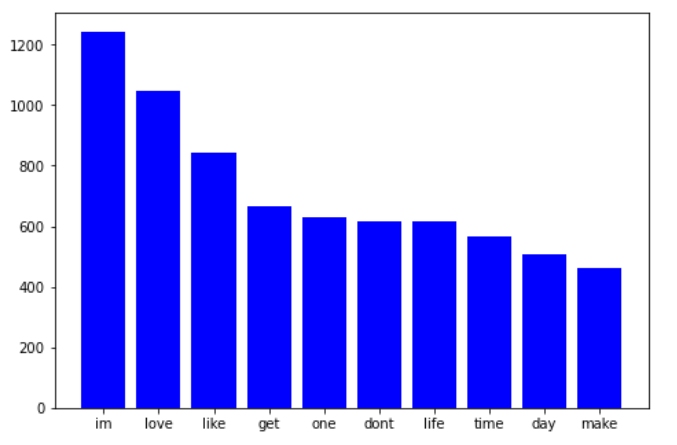
To solve Q1 a function is created that creates a dictionary of the words used by the specific gender as keys and there counts as values. The dictionary of words used by males is stored in male\_dict and for females were stored in female\_dict. The top 3 words used by males were **like**, **httpstco**, **Im** with a count of **253 250** and **242** respectively. Similarly top 3 words used by females were **Im** , **like** , **get** with a count of **434 337** and **275** respectively.

To a solve Q2 a function is created to check the total number of typos (misspelled words / Irrelavent words). The SpellChecker library was used to check the typos in a tweet. The count of total typos in a tweet were stored in dataframe format. For male the count was stored maletypos\_count and for females counts were stored in femaletypos\_count. It was obseverd that the average number of typos in tweets were **1.89** and for females the average typos in tweets were **1.67**

DATA VISUALISATION

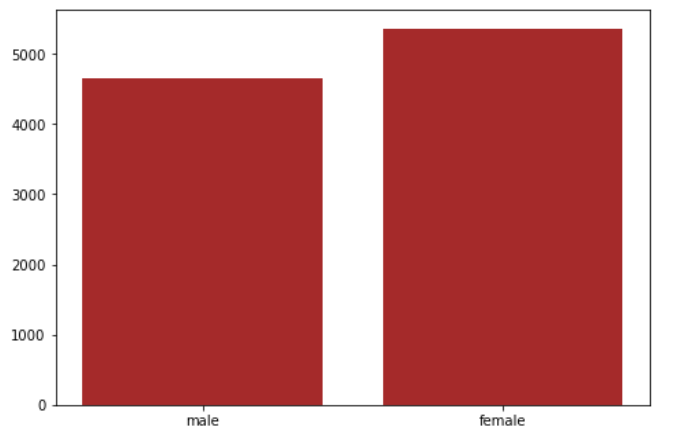
Later data visualisations was done on the following aspects

1. bar graph of counts of top 10 most common words by all genders



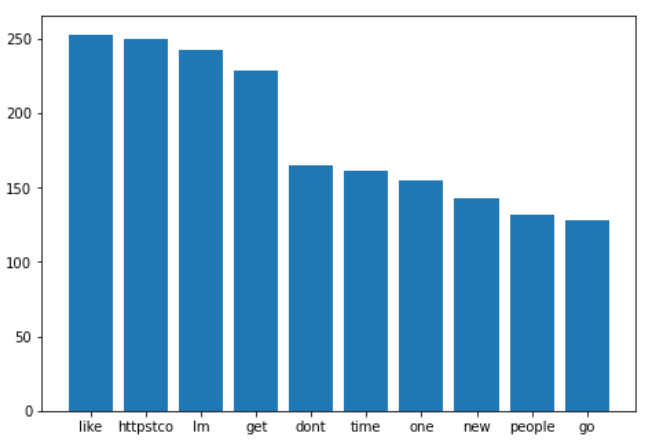
Where the x-axis is top 10 words and y-axis is counts

1. bar graph of counts of total tweets by male and female



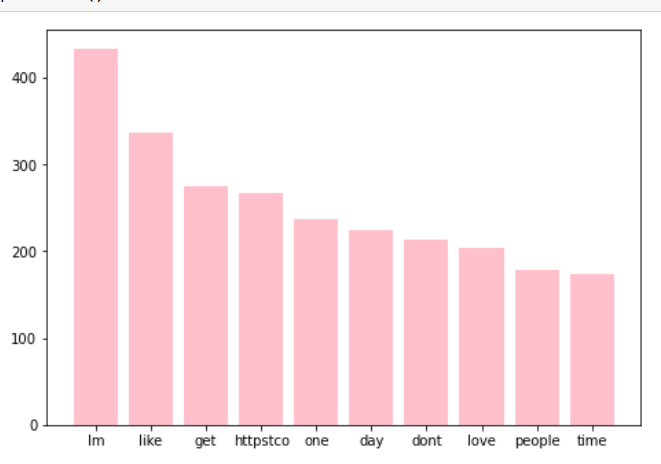
Where the x-axis is gender and y-axis is counts

1. bar graph of counts of top 10 most common words by males including typos and sensless words



Where the x-axis is top 10 words used by male and y-axis is counts

1. bar graph of counts of top 10 most common words by females including typos and sensless words



Where the x-axis is top 10 words used by male and y-axis is counts

**Feature Selection and text pre processing**

Since the independent variables were ‘description’ and ‘text’ and dependant variable was ‘gender’. A new dataframe is created named as df\_final which has only 3 columns ‘description’, ‘text’ and ‘gender’. Add all the rows with null values were removed.

After this texts the two columns ‘description’ and ‘text’ were joined together and stored in a new column 'desc+text'.

After this all the punctuations and stop words were removed from the texts and were stored in a list format by tokenizing the words by splitting the string with respect to spaces. After this the all the words were converted into lower case and also all the typos and irrelevant words were removed.

After this the WordLemmatizer was imported from the nltk library and all the words that were present after the above preprocessing were lemmatized.

A dictionary was also created that stores count of all the words.

After the CountVectorizer function was applied on the text to convert it into a sparse matrix. Also the tfidTransformer function was applied to the sparse matrix.

After the text was converted into a sparse matrix the function LabelEncoder was applied on the target variable. Male was labelled to 1 and female was labelled to 0

**Ensemble Machine learning Model creation**

After the complete pre-processing of the model the data was split into 20% testing data 80% training data. Now 3 sets of empty data dataframes were created from training and testing. They were named as model1x, model2x, model3x, model1y, model2y and model3y. After this a for loop was run 3 times. Each time the training data was split into 3 sets of 33% each and append to the 3 sets of dataframes created early (where model1x, model2x, model3x were input variables and model1y, model2y, model3y are target variables)

After this 4 algorithms were imported namely KNN, Logistic Regression, Decision Tree, Multinomial Naïve Bayes. Now this model were trained with (model1x, model1y), (model2x, model2y), (model3x, model3y) and (train\_x, train\_y ) respectively. Where train\_x and train\_y were original training set.

After these all the algorithms were fed with the same testing input and the output as given on the basis of majority vote of the output predicted by all the algorithms for a particular set of input (eg:- if each algo predict output as 1 , 0 , 0 , 0 respectively. Then the final prediction would be 0)

And all these predictions were stored in variable pred\_final

**Accuracy calculation**

After the prediction of the data accuracy was calculated for each individual algorithm using the accuracy\_score function from sklearn. We were able to achieve individuals accuracies of around 65%, 56%, 65%, 59% for models namely Logistic Regression, Decision Tree, Multinomial Naïve Bayes, KNN respectively.

And the final accuracy of the ensemble model was around 65.84%

**Calculating accuracies of inbuilt sklearn ensemble models**

After creating our own model we tried to test the dataset with inbuilt sklearn ensemble models. The model used were Random Forest Classifier, AdaBoost Classifier, Bagging Classifier (with base model of logistic regression) after training and testing this model we got an accuracy of 53%,53% and 65% respectively.