MAJOR PROJECT

Arpit Nigam
nigamarpit7000@gmail.com

This is with reference to the Major Project for ML Batch2.

Problem Statement: **For a given dataset (problem) which is the best classification algorithm (as per accuracy)**

Approach: I tried to break the entire process in steps so that it may ease the process of invigilation
Also I went ahead and used the dataset provided by the mentor (the once which was attached in the mail) for the part -2 of the problem statement I used a file named 'aspell.txt' which contains the misspelled words the given file can be found in kaggle and to ease the process of invigilation i have zipped it along with the report.

Step -1)

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

The dataset which was provided was quite raw and needed a lot of preprocessing and cleaning before it could be used to design the model.
Some of the steps included in cleaning the dataset can be seen in the code below.

```
In [11]: import re

def cleaning(s):
    s = str(s)
    s = re.sub(r'@[a-zA-Z0-9_]+','',s) #removing @Username
    s = s.lower()
    s = s.replace("'s",' is')
    s = s.replace("'re",' are')
    s = s.replace("'ve",' have')
    s = re.sub('\s\W',' ',s) #whitespace characters
    s = re.sub(r'http\S+', '', s ,flags=re.MULTILINE) #Removing URls
    s = re.sub('\W,\s',' ',s)
    s = re.sub(r'[^\w]', ' ', s)
    s = re.sub("\d+", "", s)
    s = re.sub('\s+',' ',s)
    s = re.sub('[!@#$_]', '', s)
    s = re.sub(r' [a-zA-Z0-9_] ',' ',s)#Removing single letters
    # Removing some Diacritic letter
    s = s.replace("ù","")
    s = s.replace("û", "")
    s = s.replace("âù", "")
    s = s.replace("ü", "")
    s = s.replace("å", "")
    s = s.replace("â", "")
    s = s.replace("ä", "")
    s = s.replace("ï", "")
    s = s.replace("î", "")
    s = s.replace("é", "")
    s = s.replace("è", "")
    s = s.replace("ô", "")
    s = s.replace("", "")
    s = s.replace(",","")
    s = s.replace("[\w*"," ")
    s = s.replace(" d ",' the ')

    return s
```

Exploring the data made me realise that only the columns ‘text’,’description’,’gender ‘ are going to be important for the model so i extracted only these columns.
Also since the size of the data set was too much i dropped the null values and extracted the rows only if the confidence was greater than 0.5 for a more accurate model.

gender:confidence

In [7]:

```
data = twitter_data.loc[(twitter_data['gender:confidence']>0.5) & (twitter_data['gender']!='unknown') & (twitter_dat
data.shape
```

Out[7]: (18009, 26)

In [8]:

```
data = data.loc[:,['description','text','gender']]
data.head(5)
```

Out[8]:

	description	text	gender
0	i sing my own rhythm.	Robbie E Responds To Critics After Win Against...	male
1	I'm the author of novels filled with family dr...	ÛÏIt felt like they were my friends and I was...	male
2	louis whining and squealing and all	i absolutely adore when louis starts the songs...	male
3	Mobile guy. 49ers, Shazam, Google, Kleiner Pe...	Hi @JordanSpieth - Looking at the url - do you...	male
4	Ricky Wilson The Best FRONTMAN/Kaiser Chiefs T...	Watching Neighbours on Sky+ catching up with t...	female

In [9]:

```
data.dropna()
```

Out[9]:

	description	text	gender
0	i sing my own rhythm.	Robbie E Responds To Critics After Win Against...	male
1	I'm the author of novels filled with family dr...	ÛÏIt felt like they were my friends and I was...	male
2	louis whining and squealing and all	i absolutely adore when louis starts the songs...	male
3	Mobile guy. 49ers, Shazam, Google, Kleiner Pe...	Hi @JordanSpieth - Looking at the url - do you...	male
4	Ricky Wilson The Best FRONTMAN/Kaiser Chiefs T...	Watching Neighbours on Sky+ catching up with t...	female
...	...	...	...
20045	(rp)	@lookupondeath ...Fine, and I'll drink tea too...	female

[5]:

```
df.shape
```

[5]: (20050, 26)

[6]:

```
df.info
```

[6]:

```
<bound method DataFrame.info of        _unit_id  _golden _unit_state  _t
0      815719226    False   finalized               3    10/26/15 23:
1      815719227    False   finalized               3    10/26/15 23:
2      815719228    False   finalized               3    10/26/15 23:
3      815719229    False   finalized               3    10/26/15 23:
4      815719230    False   finalized               3     10/27/15 1:
...          ...      ...         ...             ...              .
20045  815757572     True      golden             259               N
20046  815757681     True      golden             248               N
20047  815757830     True      golden             264               N
20048  815757921     True      golden             250               N
```

[20050 rows x 26 columns]>

[7]:
```
df.describe()
```

[7]:

	_unit_id	_trusted_judgments	gender:confidence	profile_yn:confidence	fav_number	retweet_count	tweet_count	tweet_id
count	2.005000e+04	20050.000000	20024.000000	20050.000000	20050.000000	20050.000000	2.005000e+04	2.005000e+04
mean	8.157294e+08	3.615711	0.882756	0.993221	4382.201646	0.079401	3.892469e+04	6.587350e+17
std	6.000801e+03	12.331890	0.191403	0.047168	12518.575919	2.649751	1.168371e+05	5.000124e+12
min	8.157192e+08	3.000000	0.000000	0.627200	0.000000	0.000000	1.000000e+00	6.587300e+17
25%	8.157243e+08	3.000000	0.677800	1.000000	11.000000	0.000000	2.398000e+03	6.587300e+17
50%	8.157294e+08	3.000000	1.000000	1.000000	456.000000	0.000000	1.144150e+04	6.587300e+17
75%	8.157345e+08	3.000000	1.000000	1.000000	3315.500000	0.000000	4.002750e+04	6.587400e+17
max	8.157580e+08	274.000000	1.000000	1.000000	341621.000000	330.000000	2.680199e+06	6.587400e+17

[8]:
```
df.isnull().sum().sort_values(ascending = False)
```

[8]:
```
gender_gold              20000
profile_yn_gold          20000
tweet_coord              19891
user_timezone             7798
tweet_location            7484
description               3744
gender                      97
_last_judgment_at           50
gender:confidence           26
created                      0
```

]:
```
df.select_dtypes(include =['number'] ).columns
```

]:
```
Index(['_unit_id', '_trusted_judgments', 'gender:confidence',
       'profile_yn:confidence', 'fav_number', 'retweet_count', 'tweet_count',
       'tweet_id'],
      dtype='object')
```

]:
```
df.select_dtypes(include =['object'] ).columns
```

]:
```
Index(['_unit_state', '_last_judgment_at', 'gender', 'profile_yn', 'created',
       'description', 'gender_gold', 'link_color', 'name', 'profile_yn_gold',
       'profileimage', 'sidebar_color', 'text', 'tweet_coord', 'tweet_created',
       'tweet_location', 'user_timezone'],
      dtype='object')
```

.]:
```
df.describe(include=object)
```

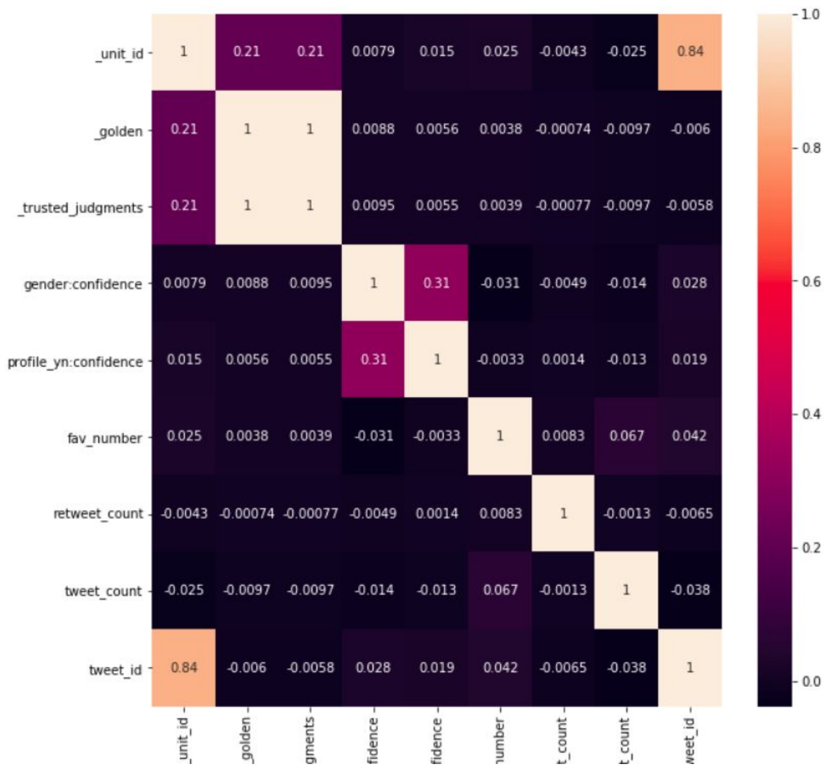
.]:

	_unit_state	_last_judgment_at	gender	profile_yn	created	description	gender_gold	link_color	name	profile_yn_gold	
count	20050	20000	19953	20050	20050	16306	50	20050	20050	50	
unique	2	283	4	2	18699	15140	6	3001	18795	1	
top	finalized	10/26/15 23:05	female	yes	8/24/15 14:19	You can be spiritually empowered, financially ...	male	0084B4	TudoSobreQuase	yes	https://abs.twimg.co
freq	20000	217	6700	19953	30	33	19	9890	30	50	

]:
```
df.dtypes
```

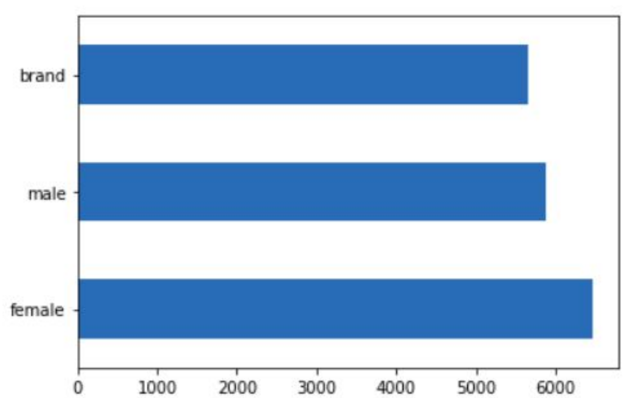
]:
```
_unit_id                    int64
_golden                      bool
_unit_state                object
_trusted_judgments          int64
```

<matplotlib.axes._subplots.AxesSubplot at 0x7f12a1b9c890>



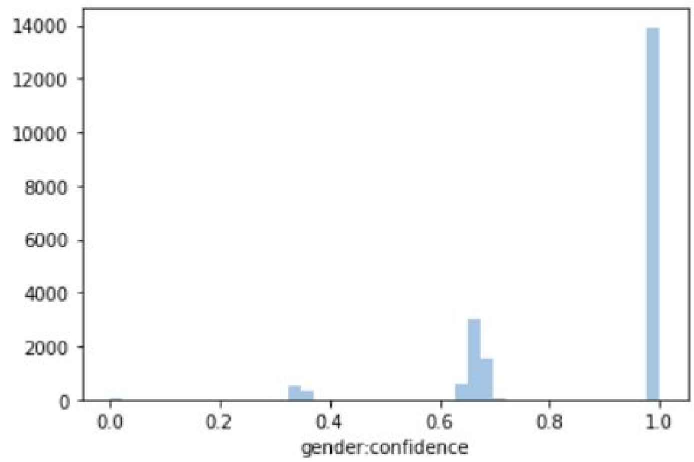

```
]: clean_data['Gender'].value_counts().plot(kind='barh')
```

]: <matplotlib.axes._subplots.AxesSubplot at 0x7f639e610c50>



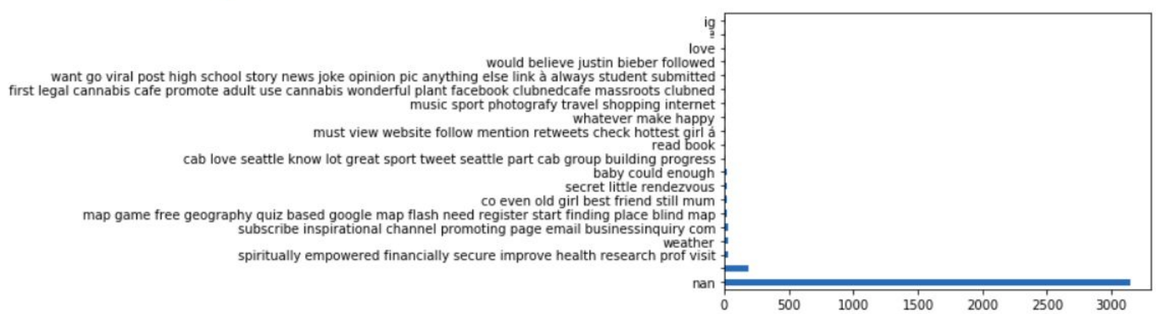
```python
sns.distplot(twitter_data['gender:confidence'],kde=False)
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7f63a78a1c90>
```




```python
clean_data['Description'].value_counts()[:20].plot(kind='barh')
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7f639e589250>
```



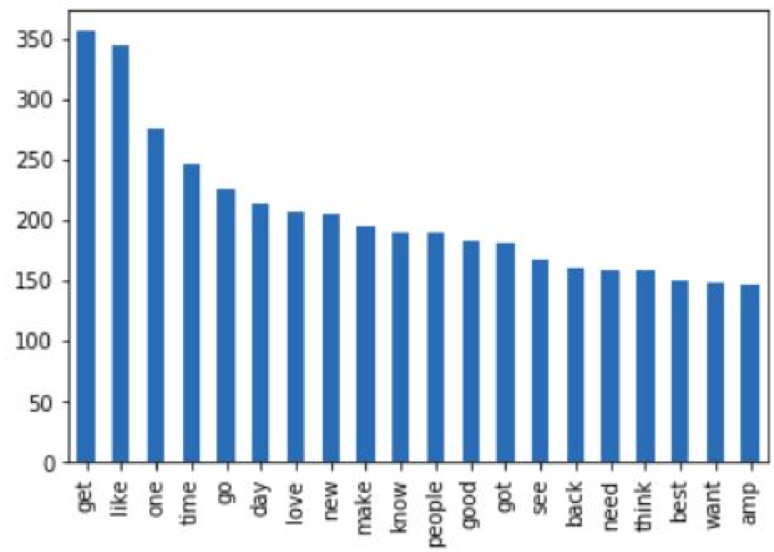
Step 2:Questions asked on dataset and answers for the same with brief explanation

Question 1) What are the most common emotions/words used by Males and Females?

Ans) the answer to this question can better be represented with the help of a graph as follows:

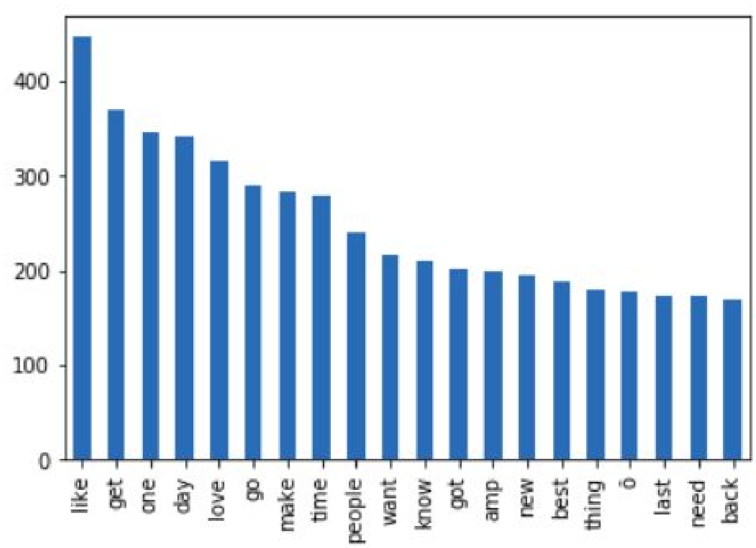
]: Male_Words.plot(kind='bar',stacked=True)

]: <matplotlib.axes._subplots.AxesSubplot at 0x7fe4a491c3d0>



: Female_Words.plot(kind='bar',stacked=True)

: <matplotlib.axes._subplots.AxesSubplot at 0x7fe4a48a8ed0>



Q2) Which gender makes more typos in their tweets?

I made use of the aspell.txt about which i mentioned in the earlier context

Thus the results can be analysed as Brand makes the most number of mistakes.(97) then male(81) female (74).

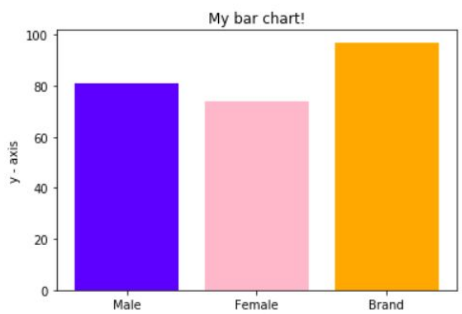
Note: The mistake count is not highly universal whereas it is relative as it only contains the words which are present in the file aspell.txt ,So if a word is not present it is considered to be accurate and is not considered.

So if we use a dictionary with more number of words.We may find that our answers change

```
[49]: import matplotlib.pyplot as plt
tick_label = ['Male', 'Female','Brand']
left = [1, 2,3]
height = [misspelled_correction(Male_Word),misspelled_correction(Female_Word),misspelled_correction(Brand_Word)]
# plotting a bar chart
plt.bar(left, height, tick_label = tick_label,
        width = 0.8, color = ['blue', 'pink','orange'])

# naming the x-axis
plt.xlabel('x - axis')
# naming the y-axis
plt.ylabel('y - axis')
# plot title
plt.title('My bar chart!')

# function to show the plot
plt.show()
# thus we conclude males make more mistakes
```



Step 3) Feature Selection and feature Engineering if required depending on the dataset

The features were selected and processed as when required in the various stages of the project For eg.

Exploring the data made me realise that only the columns 'text','description','gender ' are going to be important for the model so i extracted only these columns.
Also since the size of the data set was too much i dropped the null values and extracted the rows only if the confidence was greater than 0.5 for a more accurate model.

Step 4) Ensemble Machine learning Modelling (3 Classification Algorithms)

Due to the sufficient time provided by the mentor i made two separate models in the first I used the 'Text' as my independent variable in the second I used 'Description' as my dependent variable for bothe the dependent variable was 'Gender'

The machine learning algorithms I used for the classification were -:

1)Random forest

When i used Description

```
: from sklearn.ensemble import RandomForestClassifier

  rfc = RandomForestClassifier(n_estimators = 100)

  rfc.fit(train_x,train_y)

: RandomForestClassifier(bootstrap=True, ccp_alpha=0.0, class_weight=None,
                         criterion='gini', max_depth=None, max_features='auto',
                         max_leaf_nodes=None, max_samples=None,
                         min_impurity_decrease=0.0, min_impurity_split=None,
                         min_samples_leaf=1, min_samples_split=2,
                         min_weight_fraction_leaf=0.0, n_estimators=100,
                         n_jobs=None, oob_score=False, random_state=None,
                         verbose=0, warm_start=False)

: rfc.score(test_x,test_y)

: 0.5805243445692884
```

When i used Text::

```
: from sklearn.ensemble import RandomForestClassifier

  rfc = RandomForestClassifier(n_estimators = 100)

  rfc.fit(train_x,train_y)

: RandomForestClassifier(bootstrap=True, ccp_alpha=0.0, class_weight=None,
                         criterion='gini', max_depth=None, max_features='auto'
                         max_leaf_nodes=None, max_samples=None,
                         min_impurity_decrease=0.0, min_impurity_split=None,
                         min_samples_leaf=1, min_samples_split=2,
                         min_weight_fraction_leaf=0.0, n_estimators=100,
                         n_jobs=None, oob_score=False, random_state=None,
                         verbose=0, warm_start=False)

: rfc.score(test_x,test_y)

: 0.4666666666666667
```

Since Description was highly accurate for further discussion I used Description as the user tweet

2)SVM

```
: from sklearn.svm import SVC
  svclassifier = SVC(kernel='linear')
  svclassifier.fit(train_x, train_y)

: SVC(C=1.0, break_ties=False, cache_size=200, class_weight=None, coef0=0.0,
      decision_function_shape='ovr', degree=3, gamma='scale', kernel='linear',
      max_iter=-1, probability=False, random_state=None, shrinking=True,
      tol=0.001, verbose=False)

: y_pred = svclassifier.predict(test_x)

: from sklearn.metrics import classification_report, confusion_matrix
  print(confusion_matrix(test_y,y_pred))
  print(classification_report(test_y,y_pred))

  [[858 162  71]
   [570 391  88]
   [229 118 450]]
                precision    recall  f1-score   support

             0       0.52      0.79      0.62      1091
             1       0.58      0.37      0.45      1049
             2       0.74      0.56      0.64       797

      accuracy                           0.58      2937
     macro avg       0.61      0.57      0.57      2937
  weighted avg       0.60      0.58      0.57      2937
```

Conclusion Brand is easy to seperate but genders are relatively more complex

3)Logistic Regression

```
9]: logreg.fit(train_x,train_y)

    /home/arpit/anaconda3/lib/python3.7/site-packages/sklearn/linear_model/_logistic.py:940: ConvergenceW
    failed to converge (status=1):
    STOP: TOTAL NO. of ITERATIONS REACHED LIMIT.

    Increase the number of iterations (max_iter) or scale the data as shown in:
        https://scikit-learn.org/stable/modules/preprocessing.html
    Please also refer to the documentation for alternative solver options:
        https://scikit-learn.org/stable/modules/linear_model.html#logistic-regression
      extra_warning_msg=_LOGISTIC_SOLVER_CONVERGENCE_MSG)
9]: LogisticRegression(C=1.0, class_weight=None, dual=False, fit_intercept=True,
                       intercept_scaling=1, l1_ratio=None, max_iter=100,
                       multi_class='auto', n_jobs=None, penalty='l2',
                       random_state=None, solver='lbfgs', tol=0.0001, verbose=0,
                       warm_start=False)

9]: logreg.score(test_x,test_y)
9]: 0.5914198161389173
```

Step 5) Accuracy calculation

For both the considerations i.e Text and Description SVM gave the highest accuracy around 61% along with Description

The other notable mentions were Logistic regression (59%)
& Random Forest(58%)

Step 6) Summarised write up at the end: This report summarises all the major things to be highlighted in the major project.
The key observations were shown through the screenshots at the various stages of writing the report.

A word of thank:

This major project made my concepts really clear all the concepts taught in the class helped me in completing the project I would really like to Thank my mentor Mr. Aqib Ahmed for teaching us all these concepts in such an interesting and efficient way and also for not making these 2 hours online class not boring at all.And also for keeping my morale up while completing the course and also a special thanks to the Support team of Verzeo for answering all my queries asap.The course was well designed with really good mentors and support staff spanning fo rjust the right duration of time.