CONTENTS

Acknowledgement

IT GIVES US IMENSE PLEASURE IN BRINGING OUT THIS SYNOPSIS OF THE PROJECT ENTITLED

“ML Project on Information.csv”

FIRSTLY, WE WOULD LIKE TO THANK OUR TEACHER AND MENTOR MR. AQIB AHMED WHO GAVE US HIS VALUABLE SUGGESTIONS AND IDEAS WHEN WE WERE IN NEED OF THEM. HE ENCOURAGED US TO WORK ON THIS PROJECT.

WE ARE ALSO GRATEFUL TO VERZEO FOR GIVING US THE OPPORTUNITY TO WORK WITH THEM AND PROVIDING US THE NECESSARY RESOURCES FOR THE PROJECT

WE WOULD ALSO THANK TO ALL OF THEM WHO HELPED US TO COMPLETE THIS PROJECT.

WE ARE IMMENSELY GRATEFULL TO ALL INVOLVED IN THIS PROJECT AS WITHOUT THEIR INSPITRATION AND VALUABLE SUGGESTION IT WOULD NOT HAVE BEEN POSSIBLE TO DEVELOP THE PROJECT WITHIN THE PRESCRIBED TIME.

WITH SINCERE THANKS,

“ML062B12”

Problem Statement for Project Analysis:

For a given dataset,

Information.csv

Information.csv

Open with Google Sheets

A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
_unit_id	_golden	_unit_state	_trusted_judgments	_last_judgment_at	gender	gender:confidence	profile_yn	profile_yn:confidenc	created	description	fav_number	gender_gold	link_color	name
815719226	FALSE	finalized	3	10/26/15 23:24	male	1	yes	1	12/5/13 1:48	i sing my own rhythm	0		08C2C2	sheezy0
815719227	FALSE	finalized	3	10/26/15 23:30	male	1	yes	1	10/1/12 13:51	I'm the author of nov	68		0084B4	DavdBurnett
815719228	FALSE	finalized	3	10/26/15 23:33	male	0.6625	yes	1	11/28/14 11:30	louis whining and sq	7696		AB88C2	lwtprettylaugh
815719229	FALSE	finalized	3	10/26/15 23:10	male	1	yes	1	6/11/09 22:39	Mobile guy. 49ers, S	202		0084B4	douggarland
815719230	FALSE	finalized	3	10/27/15 1:15	female	1	yes	1	4/16/14 13:23	Ricky Wilson The Be	37318		3B94D9	WilfordGemma
815719231	FALSE	finalized	3	10/27/15 1:47	female	1	yes	1	3/11/10 18:14	you don't know me.	3901		F5ABB5	monroevicious
815719232	FALSE	finalized	3	10/27/15 1:57	brand	1	yes	1	4/24/08 13:03	A global marketplace	4122		298AAE	Shutterstock
815719233	FALSE	finalized	3	10/26/15 23:48	male	1	yes	1	12/3/12 21:54	The secret of getting	80		0000FF	RobinMeske
815719234	FALSE	finalized	3	10/27/15 1:52	female	1	yes	1	9/8/15 4:50	Pll Fan // Crazy abou	1825		9266CC	pigzilla_
815719235	FALSE	finalized	3	10/27/15 1:49	female	1	yes	1	5/13/11 3:32	Renaissance art histo	3115		9266CC	GabrieleNeher
815719236	FALSE	finalized	3	10/26/15 23:17	brand	0.7002	yes	1	11/16/11 17:14	Clean food that taste	516		0084B4	Blissful_Eats
815719237	FALSE	finalized	3	10/26/15 22:33	brand	1	yes	1	2/22/15 20:06	highly extraordinary	0		0084B4	ellseotecleo
815719238	FALSE	finalized	3	10/26/15 22:20	female	0.6509	yes	1	8/10/12 5:05	Senior '16 . XI-XII-MI	3371		0084B4	kaylanabrewer_
815719239	FALSE	finalized	3	10/26/15 23:29	brand	1	yes	1	5/1/12 22:14	Come join the fastes	0		2FC2EF	americacredit
815719240	FALSE	finalized	3	10/27/15 1:29	female	0.6501	yes	1	4/6/13 15:31	im just here for t◆◆p	13928		0084B4	cheyfian
815719241	FALSE	finalized	3	10/27/15 1:50	female	1	yes	1	10/3/15 21:32		0		0084B4	Ayu7sth4
815719242	FALSE	finalized	3	10/26/15 23:43	female	1	yes	1	8/27/11 9:42	JMKM◆_§◆◆	1762		58185	Toucaan_Sam
815719243	FALSE	finalized	3	10/26/15 22:50	male	1	yes	1	10/18/09 11:41	Over enthusiastic F1	5		0084B4	SSmyth2010
815719244	FALSE	finalized	3	10/27/15 1:42	male	1	yes	1	7/20/15 12:01		1		0084B4	DarkToonsGam
815719245	FALSE	finalized	3	10/26/15 22:19	unknown	0.3527	yes	1	1/30/15 9:52		160		0084B4	GourlayLynn
815719246	FALSE	finalized	3	10/27/15 1:21	female	1	yes	1	2/28/13 3:04	Artisan specializing i	18751		3B94D9	jlpeter37
815719247	FALSE	finalized	3	10/27/15 0:09	female	1	yes	1	10/14/11 17:53	He bled and died to	5454		D41EBF	SarahMaddry
815719248	FALSE	finalized	3	10/26/15 22:47	female	1	yes	1	2/15/15 6:41	union j xxxx	133		0084B4	bethleia
815719249	FALSE	finalized	3	10/27/15 1:34	male	1	yes	1	1/11/13 1:18	You had me from the	2871		3B94D9	StrictlyAri
815719250	FALSE	finalized	3	10/26/15 23:01	male	1	yes	1	4/21/11 12:21	BSc economics grad	228		0084B4	JHurkett
815719251	FALSE	finalized	3	10/26/15 22:54	female	1	yes	1	8/14/13 12:54	Wife to my Coach. M	1051		0084B4	karenhbullard
815719252	FALSE	finalized	3	10/26/15 22:24	brand	0.6576	yes	1	8/8/15 16:15	If you have any ques	0		0084B4	1oneonlyone1
815719253	FALSE	finalized	3	10/27/15 1:22	brand	0.6667	yes	1	4/26/15 8:15	14 ,Canadian , Spac	5601		DD2E44	the_offical_ios

Find out the best algorithm as per accuracy.

Analysis:

After having a brief look, we found that the dataset (information.csv) gives very detailed and vast information about the tweets posted by both males and females. We also found that there was still some scope for cleansing of the data so as to classify the data more efficiently and accurately using various classification algorithms.
Thus, We went ahead and performed EDA in order to clean the dataset, in the first step of this project work.

Libraries Imported and Used for the project:

1. Numpy
2. Pandas
3. Sklearn
    - Preprocessing
    - Model_Selection
    - Metrics
    - DictVectorizer
    - TfidVectorizer
    - LogisticRegression
    - KNeigborsClassifier
    - SVC (sklearn.svm)
    - RandomForestClassifier
4. Nltk
    - Corpus
5. Re
6. Matplotlib.Pyplot
7. TextBlob

MAJOR PROJECT

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

In [1]:
```
import numpy as np
import pandas as pd
from sklearn import preprocessing
import nltk
nltk.download("stopwords")
from nltk.corpus import stopwords
import re
```

```
[nltk_data] Downloading package stopwords to
[nltk_data]     C:\Users\KIIT\AppData\Roaming\nltk_data...
[nltk_data]   Package stopwords is already up-to-date!
```

In [2]:
```
import matplotlib.pyplot as plt
from sklearn.preprocessing import LabelEncoder
from sklearn.model_selection import train_test_split
from sklearn.metrics import accuracy_score
```

In [3]:
```
#Reading the dataset
df = pd.read_csv('Information.csv',encoding = "latin-1")
df
```

Exploratory Data Analysis (Cleaning of the Data):

The dataset provided contains: 20050 rows × 26 columns, it needed cleaning. So the next thing we did was to perform EDA on the dataset. We needed dataset with only 'male' and 'female' as 'gender' values. So we used pd.concat() to remove the unnecessary values in 'gender'. Now we were left with: 12894 rows × 26 columns. Accordingly, we used appropriate functions to clean the dataset. In addition, 'text' is cleaned and stored as 'Tweets' whereas 'description' is cleaned and stored as 'Desc'. We also dropped those rows where 'location' was not available, to be left with: 8747 rows × 28 columns and hence, performed EDA successfully.

Here we cleaned the data in the following steps:

- Kept Data where gender was either male or female.

EDA

```
In [4]:
#Keeping those data where gender is male or female
df_male = df[df["gender"] == "male"]
df_female = df[df["gender"] == "female"]
df = pd.concat([df_male,df_female])
```

- Defined a function clean() which uses regular expressions to clean the data.

```
In [5]:
#Function to clean the dataset
def clean(s):
    s = str(s)
    s = s.lower()
    s = re.sub('\s\W',' ',s)
    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 = s.replace("co","")
    s = s.replace("https","")
    s = s.replace(",","")
    s = s.replace("[\w*"," ")
    return s
```

- Using the clean function cleaned and changed words like "text" and "description" into "Tweets" and "Desc" respectively.

```
In [6]:
df['Tweets'] = [clean(s) for s in df['text']]
df['Desc'] = [clean(s) for s in df['description']]

stop = set(stopwords.words('english'))
df['Tweets'] = df['Tweets'].str.lower().str.split()
df['Tweets'] = df['Tweets'].apply(lambda x : [item for item in x if item not in stop])

for i in range(df.shape[1]):
    df[df.columns[i]] = [clean(s) for s in df[df.columns[i]]]

#'text' is cleaned and stored as 'Tweets'
#'description' is cleaned and stored as 'Desc'
```

- Dropped data where Tweet Location was not available.

```
In [7]:
#Dropped those where location was not available
df = df[df["tweet_location"]!="nan"]
df
```

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

Using Pandas' Series Function and plotting the graphs of tweet word counts of both Males and Females (using Matplotlib).

We found out that:

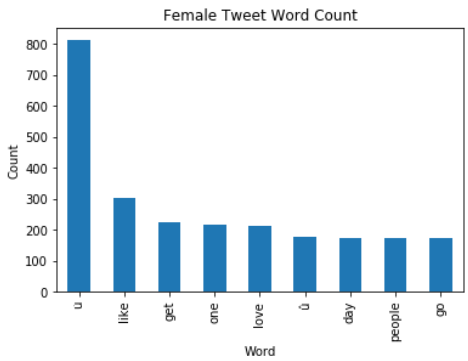
The most common emotions/words used by Males : ú , get
The most common emotions/words used by Females : ú , like

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

In [8]:
```
male = df[df['gender'] == 'male']
female = df[df['gender'] == 'female']
male_words = pd.Series(' '.join(male['Tweets'].astype(str)).lower().split(" ")).value_counts()[:10]
female_words = pd.Series(' '.join(female['Tweets'].astype(str)).lower().split(" ")).value_counts()[:10]
male_words = male_words.iloc[1:]
female_words = female_words.iloc[1:]
```

In [9]:
```
plt.title("Female Tweet Word Count")
plt.xlabel("Word")
plt.ylabel("Count")
female_words.plot.bar()
```

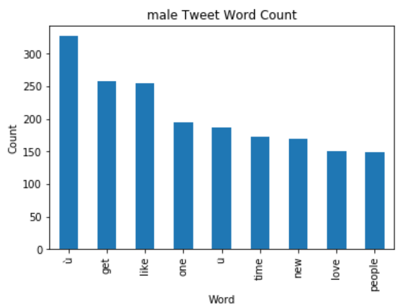
Out[9]: <matplotlib.axes._subplots.AxesSubplot at 0x22a186de548>



(Female Tweet Word Count)

In [10]:
```
plt.title("male Tweet Word Count")
plt.xlabel("Word")
plt.ylabel("Count")
male_words.plot.bar()
```

Out[10]: <matplotlib.axes._subplots.AxesSubplot at 0x22a175fbf88>



Answer:
The most common emotions/words used by Males : ú , get
The most common emotions/words used by Females : ú , like

(Male Tweet Word Count)

Q2) Which gender makes more typos in their tweets?

Again, Using Pandas' Series function and Python's TextBlob library over the dataset, we calculated which gender makes more Typos(Mistakes) while writing their tweets.

We found out that it was MALES who made more number of typos in their tweets as compared to FEMALES.

Q2. Which gender makes more typos in their tweets?

In [11]:
```
male_words = pd.Series(' '.join(male['Tweets'].astype(str)).lower().split(" "))
female_words = pd.Series(' '.join(female['Tweets'].astype(str)).lower().split(" "))
```

In [12]:
```
from textblob import TextBlob
```

In [13]:
```
d=male_words.values
a=0
for word in d:
    b = TextBlob(word)
    c=str(b.correct())
    if word==c:
        pass
    else:
        a=a+1

print("Total no. of male words = ",male_words.count())
print("Total no. typos by male = ",a)
```

```
Total no. of male words =  50967
Total no. typos by male =  7590
```

In [14]:
```
d=female_words.values
e=0
for word in d:
    b = TextBlob(word)
    c=str(b.correct())
    if word==c:
        pass
    else:
        e=e+1

print("Total no. of female words = ",female_words.count())
print("Total no. typos by female = ",e)
```

```
Total no. of female words =  47482
Total no. typos by female =  6599
```

In [15]:
```
if (a/(male_words.count())) > (e/(female_words.count())):
    print("Males make more typos.")
else:
    print("Females make more typos.")
```

```
Males make more typos.
```

Answer:
Males make more typos.

Final Step: Classification of entire Dataset (Information.csv)

In order to get started with the classification of the dataset, we decided to take “Gender” as a Dependent Variable and “Description” as an Independent Variable, while also performing the Vectorization and Label Encoding on the data using Tfidvectorizer() giving us the signal to finally apply classification algorithms on the data set.

Here, we are taking "gender" as our dependent variable and "description" as our independent variable.

In [16]:
```
import collections
from sklearn.feature_extraction import DictVectorizer
from sklearn.feature_extraction.text import TfidfVectorizer
```

In [17]:
```
#Performing Vectorization and Label Encoding
Vectorize = TfidfVectorizer(stop_words='english')
x = Vectorize.fit_transform(df['Desc'])
y = df.gender
le = preprocessing.LabelEncoder()
y = le.fit_transform(y.values)
```

We Split the data set into for training and testing purposes as usual.

In [18]:
```
#Splitting the dataset for training and testing
X_train, X_test, Y_train, Y_test = train_test_split(x, y)
```

Algorithm 1- Logistic Regression:

Accuracy Achieved: 64.15%

1. Logistic Regression

In [19]:
```
from sklearn.linear_model import LogisticRegression

LogReg = LogisticRegression(random_state=25)
#Training the model
LogReg.fit(X_train, Y_train)
```

```
C:\Anaconda\lib\site-packages\sklearn\linear_model\logistic.py:432: FutureWarning: Default solver will be changed to 'lbfgs' in 0.22. Specify a solver to silence this warning.
  FutureWarning)
```

Out[19]:
```
LogisticRegression(C=1.0, class_weight=None, dual=False, fit_intercept=True,
                   intercept_scaling=1, l1_ratio=None, max_iter=100,
                   multi_class='warn', n_jobs=None, penalty='l2',
                   random_state=25, solver='warn', tol=0.0001, verbose=0,
                   warm_start=False)
```

In [20]:
```
#Testing and predicting
y_pred = LogReg.predict(X_test)
```

In [21]:
```
#Checking the accuracy
from sklearn import metrics
print("Test set Accuracy: ", metrics.accuracy_score(Y_test, y_pred))
```

```
Test set Accuracy:  0.6415180612711477
```

Algorithm 2 – K Nearest Neighbor (KNN)

Accuracy Achieved : 56.33%

2. KNN

In [22]:
```
from sklearn.neighbors import KNeighborsClassifier

knn = KNeighborsClassifier(n_neighbors=107)
#Training the model
knn.fit(X_train, Y_train)
```

Out[22]:
```
KNeighborsClassifier(algorithm='auto', leaf_size=30, metric='minkowski',
                     metric_params=None, n_jobs=None, n_neighbors=107, p=2,
                     weights='uniform')
```

In [23]:
```
#Testing and predicting
y_pred = knn.predict(X_test)
```

In [24]:
```
#Checking the accuracy
print("Test set Accuracy: ", metrics.accuracy_score(Y_test, y_pred))
```

```
Test set Accuracy:  0.563328760859625
```

Accuracy = 56.33%

Algorithm 3 – Support Vector Machine (SVM)

Accuracy Achieved : 51.53%

3. SVM

In [25]:
```
from sklearn.svm import SVC

svc = SVC(kernel='rbf',random_state=25)
#Training the model
svc.fit(X_train, Y_train)
```

```
C:\Anaconda\lib\site-packages\sklearn\svm\base.py:193: FutureWarning: The default value of gamma will change from 'auto' to
'scale' in version 0.22 to account better for unscaled features. Set gamma explicitly to 'auto' or 'scale' to avoid this war
ning.
  "avoid this warning.", FutureWarning)
```

Out[25]:
```
SVC(C=1.0, cache_size=200, class_weight=None, coef0=0.0,
    decision_function_shape='ovr', degree=3, gamma='auto_deprecated',
    kernel='rbf', max_iter=-1, probability=False, random_state=25,
    shrinking=True, tol=0.001, verbose=False)
```

In [27]:
```
#Testing and predicting
y_pred = svc.predict(X_test)
```

In [28]:
```
from sklearn.metrics import classification_report, confusion_matrix
print (classification_report(Y_test, y_pred))
```

```
              precision    recall  f1-score   support

           0       0.00      0.00      0.00      1060
           1       0.52      1.00      0.68      1127

    accuracy                           0.52      2187
   macro avg       0.26      0.50      0.34      2187
weighted avg       0.27      0.52      0.35      2187
```

```
C:\Anaconda\lib\site-packages\sklearn\metrics\classification.py:1437: UndefinedMetricWarning: Precision and F-score are ill-
defined and being set to 0.0 in labels with no predicted samples.
  'precision', 'predicted', average, warn_for)
```

In [29]:
```
#Checking the accuracy
print("Test set Accuracy: ", metrics.accuracy_score(Y_test, y_pred))
```

```
Test set Accuracy:  0.5153177869227252
```

Accuracy = 51.53%

Algorithm 4 – Random Forest

Accuracy Achieved : 60.95%

4. Random Forest

```
In [30]:  from sklearn.ensemble import RandomForestClassifier

          rfc = RandomForestClassifier(random_state=25)
          #Training the model
          rfc.fit(X_train, Y_train)

          C:\Anaconda\lib\site-packages\sklearn\ensemble\forest.py:245: FutureWarning: The default value of n_estimators will change f
          rom 10 in version 0.20 to 100 in 0.22.
            "10 in version 0.20 to 100 in 0.22.", FutureWarning)

Out[30]:  RandomForestClassifier(bootstrap=True, class_weight=None, criterion='gini',
                                 max_depth=None, max_features='auto', max_leaf_nodes=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=10,
                                 n_jobs=None, oob_score=False, random_state=25, verbose=0,
                                 warm_start=False)

In [31]:  #Testing and predicting
          y_pred = rfc.predict(X_test)

In [32]:  from sklearn.metrics import classification_report, confusion_matrix
          print (classification_report(Y_test, y_pred))

                        precision    recall  f1-score   support

                     0       0.57      0.76      0.65      1060
                     1       0.67      0.47      0.55      1127

              accuracy                           0.61      2187
             macro avg       0.62      0.61      0.60      2187
          weighted avg       0.63      0.61      0.60      2187


In [33]:  #Checking the accuracy
          print("Test set Accuracy: ", metrics.accuracy_score(Y_test, y_pred))

          Test set Accuracy:  0.6095107453132145
```

Accuracy = 60.95%

From the above results, it is found that Logistic Regression has the greatest accuracy(64.15%) in prdicting the gender of the a user

Final Result:

It was found that the highest accuracy was achieved by the Logistic Regression Classification Algorithm i.e. 64.15%, thus making it the most suitable algorithm for the given dataset INFORMATION.CSV.

Thank You.