**GENDER CLASSIFICATION USING TWITTER DATA SET.**

-VINITHA ELANGOVAN

HOME WORK-2

OBJECTIVE:

To predict the gender in the given data set (Twitter data set) using various feature engineering technique and machine learning model.

DATA SET: Gender Classification Dataset.

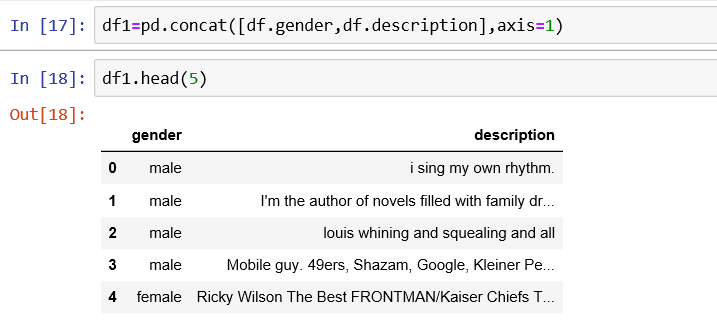
<https://www.kaggle.com/crowdflower/twitter-user-genderclassification>

1. Feature Construction:

Feature Engineering is the process of transforming raw data into feature

that better represent the underlying problem to predictive models, resulting in improved model accuracy.

Reducing the number of column in the data set and working with the column needed for training my model. The Gender Classification Dataset has 26 columns reducing it 2 column which I will be using to train the model (gender, description).Concating the data set in the following way.

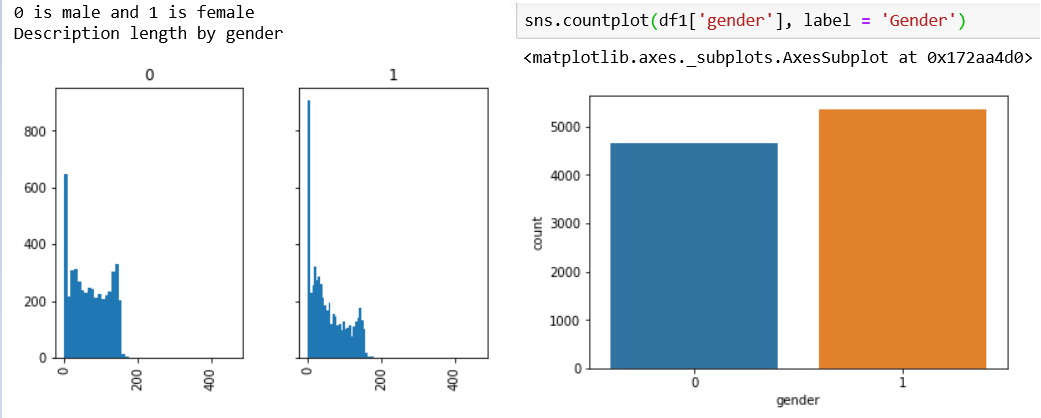


Features like Stopword, Tokenization, Lemmatization, CountVectorizer are used in analyze the text. I use CountVectorizer to count the frequency of the common repeated words in and assign the common words to the sparse\_matrix to train my model in it.

Finding the length of the description and mapping the length value to its corresponding description and assigning the values 0 and 1 to the male and female.

The min-max scalar form of normalization is used. It uses the mean and standard deviation to box all the data into a range lying between a certain min and max value. For most purposes, the range is set between 0 and 1 by this normalization technique the accuracy of the model can be increased.

Using only the data with gender confidence of 100% so that we can improve the accuracy of our model.



Using regular expression to clean the data by removing the special characters like html tags, special characters, assigning lower case value, punctuations which leads to poor performance of the model.

Once the raw data is preprocessed by the above steps it is ready to fit in to the model to classify.

1. Description Of The Classifier:

It is a binary classification problem (yes/no/male/female/spam/unspam ), we have to classify the gender in the given data set. The machine learning model I will be using is Guassian Navie Bayes Algorithm, Logistic Regression, Random Forest.

Splitting the dataset in to training and test data the training data is used to train the model and the test data is to test the model show the accuracy of the model.

*Logistic Regression:* It is a technique in statistical analysis that attempts to predict the data value based on prior observation. A logistic regression algorithm looks at the relationship between a dependent variable and one or more dependent variables. It looks in to the relationship between gender and gender description.

*Guassian Navie Bayes:* Naive Bayes classifiers have worked quite well in many real-world situations, famously document classification and spam filtering. They require a small amount of training data to estimate the necessary parameters. It trains the model in given training set and estimate the necessary parameter for the given data be male or female.

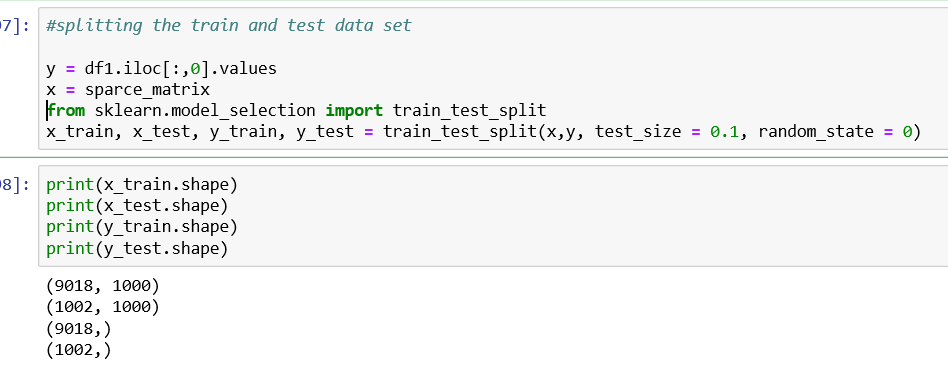
*Random Forest:* The random forest model is very good at handling tabular data with categorical features with fewer than hundreds of categories. Unlike linear models, random forests are able to capture non-linear interaction between the features and the target.

1. Evaluation Technique:

Assigning values for X and Y, splitting the train and test data, 90% of the data

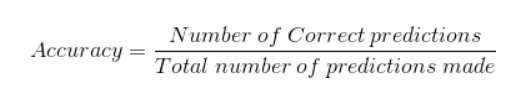
as train and 10% as test and assigning the random state=0. This is done by importing train\_test\_split from sklearn inbuild function.

Once the train data is split we fit x\_train and y\_train in to our model and train our model. Then test our model with the test data to calculate the efficiency / accuracy of the model.



Performed various metric evaluation technique like Accuracy, confusion matrix, and obtaining the classification report like precision, recall, f1-score.

*Accuracy calculation:*



*Confusion Matrix:*

**True Positives** : The cases in which we predicted YES and the actual output was also YES.

**True Negatives** : The cases in which we predicted NO and the actual output was NO.

**False Positives** : The cases in which we predicted YES and the actual output was NO.

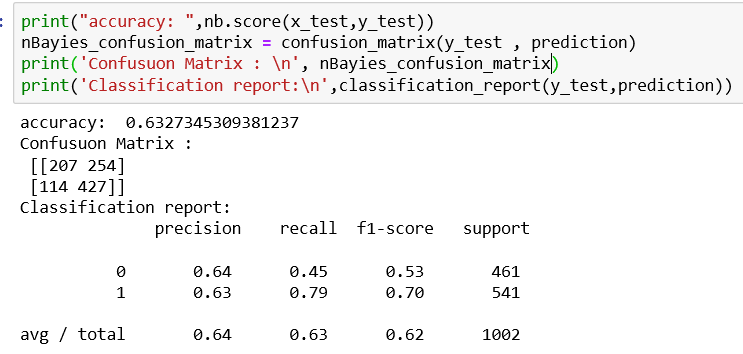
**False Negatives** : The cases in which we predicted NO and the actual output was YES.

*F1- score:*

F1-score is used to measure a test accuracy.

With the help of above technique we will come to conclusion of best fit model for the given data set.

Sample o/p: Guassian Navie Bayies.

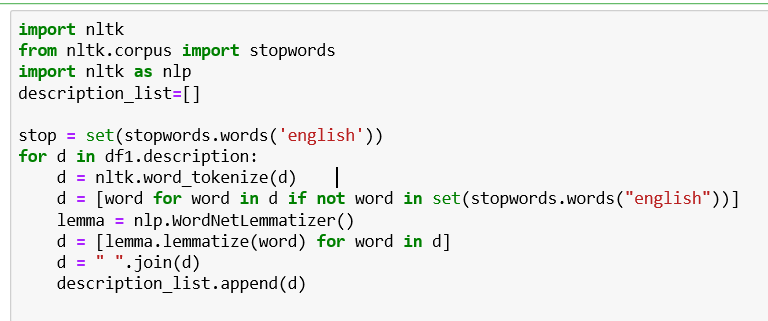


1. Implementations:

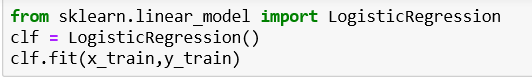
4.a. *Data Preprocessing*: Regular Expression is a inbuild python library

used in data cleaning that is removing the special characters, html tags, extra space, punctuations, converting the text into lower cases.  in built function is used to perform tokenizing and lemmatizing on the given data set. is used to import the stop words and performed on the given dataset.

4.b. As we seen in 4.a the feature is extracted using  with the help of NLTK build in function feature extraction is done in the given data set. The below are steps followed for the feature etraction.



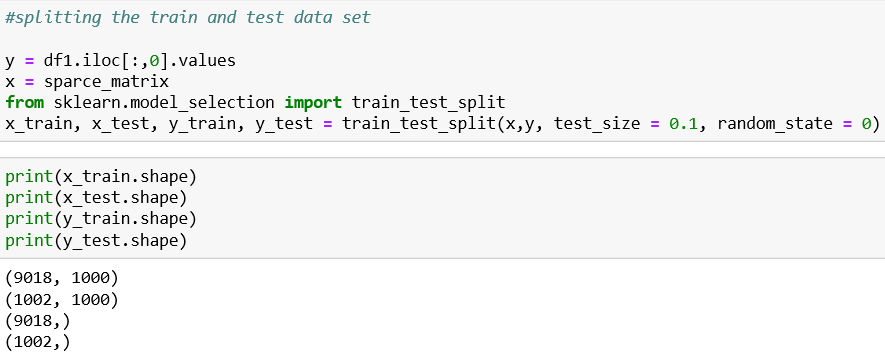
4.c *Logistic Regression:* importing the logistic regression model from the sklearn.linear\_model (Scikit) and assigning the clf variable to the model. Fitting out data in to the model.



Similarly Random forest and Guassian Navie Bayies classifier algorithms are imported  and fit in to the model to train the model.

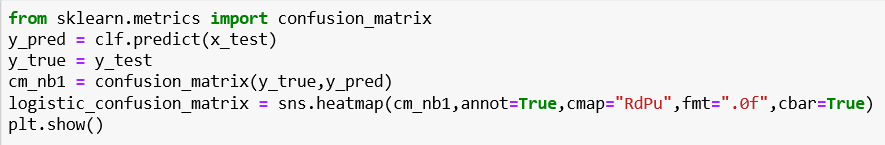
The models performances have been tuned with the help of Min\_Max scalar form of normalization. Once the models are trained we test our model with the test data and perform various metric evaluation technique to find the best fit model.

4.d importing train\_test\_split from sklearn.model\_selection, assigning the values for x and y variables and splitting the variables in to train and test data in the ratio of 90:10 so that we can train our model with 90% of data set. ML model learns from the training data more the training data more the accuracy of the model.



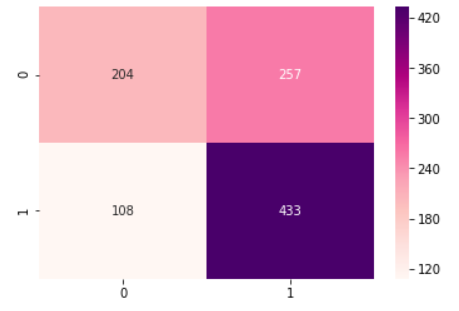
4.e The following are the metric evaluation technique which I have been used

*Confusion Matrix For Visual Representation(Logistic Regression):*



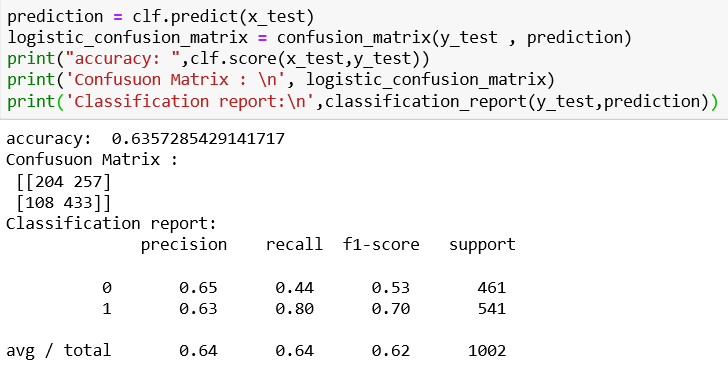
By importing the Confusion matrix from the sklearn.metrics we are assigning the X\_test predicted value for a new variable y\_pred and y\_test as y\_true now we are forming an confusion matrix with this new assigned variables. With the help of seaborn  heatmap plotting the confusion matrix.

Sample o/p:



1. Male 1- Female.

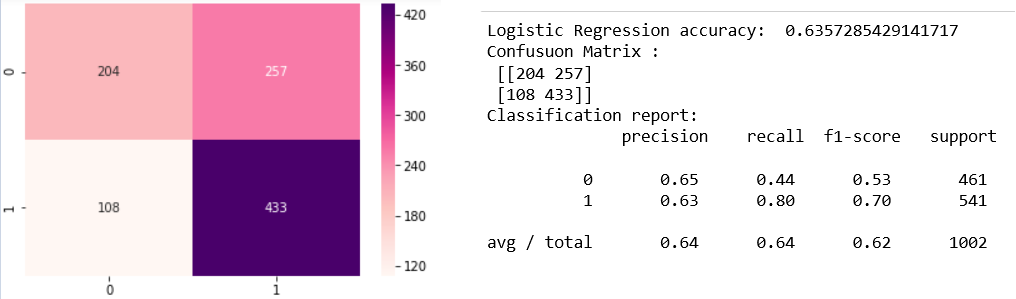
*Accuracy, Confusion Matrix Representation, Classification Report(Logistic Regression):* To find the accuracy we have to check the score of our test data. For confusion matrix x\_test predicted by the algorithm and the y\_test value is passed to obtain the confusion matrix. Similarly to the Classification report.



1. Analysis Of Result:

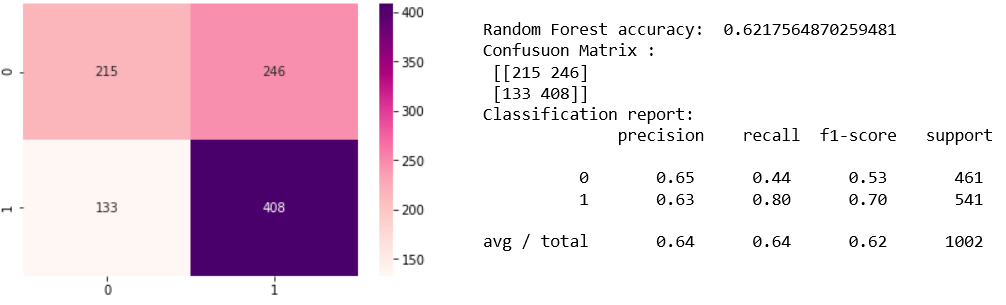
Logistic Regression provides an highest accuracy of 63.57%. By using python seaborn module I plotted heatmap for confusion\_matrix. Which is evident how many is true prediction and false prediction.

*Logistic Regression Algorithm:*

**

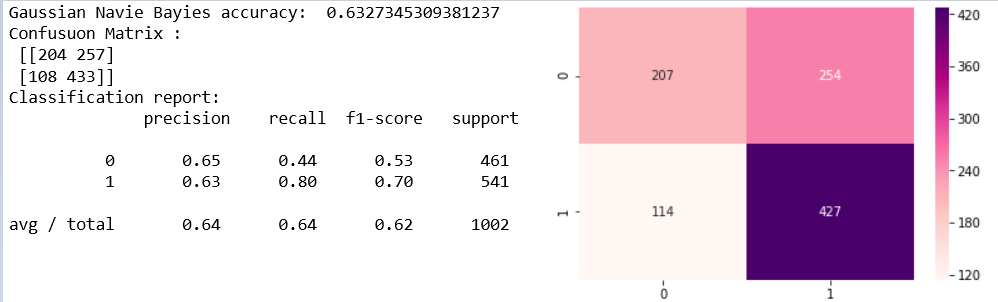
Has an accuracy of 63.57%

*Random Forest Algorithm:*

**

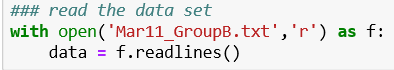
Has an accuracy of 62.17%

*Gaussian Navie Bayies Algorithm:*

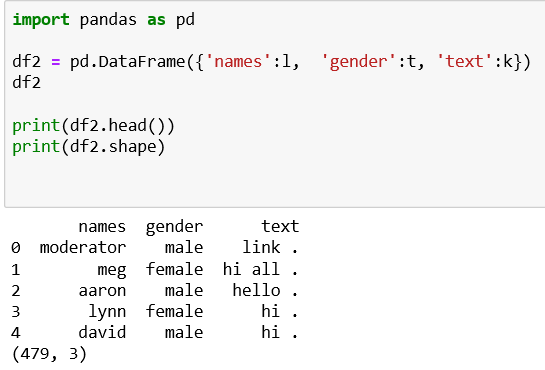
**

Has an accuracy of 63.27%

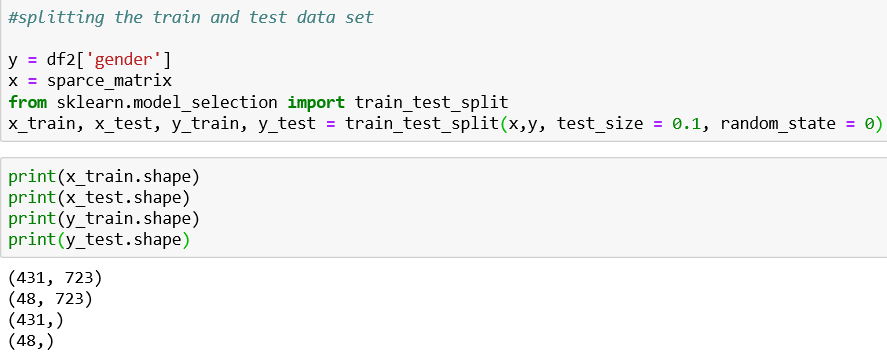
1. *Applying classifier to conversational data:*



Reading the text file and assigning the text file to the variable data. Splitting the data names and text from the given Mar11\_GroupD.txt data set and assigning them to an empty list l[] and k[]. Creating a list of male and female list. Feature extraction like Tokenizing, stopword, lemmatizing is done in the Mar11\_GroupD.txt data set



Creating an data frame including names, text, and gender. Assigning 0-male and 1-female. Data cleaning and preprocessing has done for the new data frame.



Then fitting the data in to the Logistic Regression model since it’s the best fit model and when the model is tested it gives an accuracy of 73%.

