**DESCRIPTION**

***Domain – Entertainment***

**Business challenge/requirement**

The data set contains 5331 positive and 5331 negative reviews in a file. The movie database provider wishes to quantify each review as positive or negative based on the content of reviews.

You as an NLP expert are asked to design a system and build a classifier which will flag all incoming reviews as positive or negative.

1.   Load the dataset and create a dataframe.

2.   Define a function which can perform the following functions:

* Remove non-alphabets
* Remove URLs
* Remove digits
* Remove stopwords
* Stem the texts using PorterStemmer
* Remove and replace “’”, “--”, “-”, “[”, “]” by “ ”.

 3.   Create a list of 30 most frequently occurring words from cleaned reviews and write it to 'nlargest.txt'.

4.   Create a train (67%) and test (33%) split with random state 42

5.   Create a TF-IDF vector with the following parameter:

* ngram\_range = (1,2)
* max\_df=0.3
* min\_df=7

6.   Build a Random Forest Classifier [*Preferably, perform step 5 and 6 together using****Pipeline****from****sklearn***]

7.   After building the classification model and predicting on the whole dataset, save confusion matrix to a text file using :

confusion\_matrix(observed,predicted).tofile('cfmatrix.txt',sep=',')

**Considerations**

Reviews have been saved in separate files with extension ‘pos’ and ‘neg’.

**Data volume**

Approx 10662 records

* Positive - 5331
* Negative - 5331

**Dataset Path:**

/data/training/ReviewsFileName.xlsx

**Business benefits**

Since the number of reviews is surging with the rapid pace at movies database website. So, it is becoming almost impossible for admins to read each review and gauge the sentiment of patrons about the movies. This exercise will save over 90 manhours per month.

**DATASETS**

* [Training dataset](https://media-doselect.s3.amazonaws.com/generic/qrv4pQ59L4wGq9PArA4Z91MAB/training-ReviewsFileName.zip)help\_outline

**EXECUTION TIME LIMIT**

Default.

Sample code

from sklearn import preprocessing,metrics

from sklearn.feature\_extraction.text import TfidfVectorizer, TfidfTransformer,CountVectorizer

from sklearn.naive\_bayes import MultinomialNB

from sklearn.linear\_model import SGDClassifier

from sklearn.ensemble import RandomForestClassifier

from sklearn.pipeline import Pipeline

from sklearn.model\_selection import GridSearchCV

from nltk.stem import PorterStemmer

from nltk.stem.wordnet import WordNetLemmatizer

from nltk.corpus import stopwords

from sklearn.metrics import confusion\_matrix

import pandas as pd, numpy as np

import re,nltk, textblob, string

from string import punctuation

import warnings

warnings.filterwarnings('ignore')

# #### 1. Load the dataset and create a dataframe.

# #### 2. Define a function which can perform following functions:

# `a. Remove non-alphabets

# b. Remove urls

# c. Remove digits

# d. Remove stopwords

# e. Stem the texts using PorterStemmer

# f. Remove and replace “’”, “--”, “-”, “[”, “]” by “”.

#### Creating a corpus

#### 3. Create a list of 30 most frequent occurring words from cleaned reviews and write it to 'nlargest.txt'

from nltk.probability import FreqDist

from nltk.tokenize import word\_tokenize, sent\_tokenize

from heapq import nlargest

#### 4. Create a train (67%) and test (33%) split with random state 42

from sklearn.model\_selection import train\_test\_split

# #### 5. Create TF-IDF vector

# #### 6. Build a Random Forest Classifier

# #### 7. After building classification model and predicting on whole dataset, save confusion matrix to text file