Python/Deep Learning Project Report

Project Increment - 1

Classification of News into Categories Based on Headlines & Short Description

Team ID: 2

Member 1: Akhil Teja Kanugolu **Class ID: 11**

Member 2: Geetanjali Makineni **Class ID:13**

Work done so far:

Basically, we divided the whole project into 3 components as

- Dataset Preparation
- Feature Engineering
- Model Training

1) Dataset Preparation: The first step here is dataset preparation where we load the dataset and perform the basic preprocessing. The dataset is further split into training, validation and test sets.

2) Feature Engineering – In this step, the raw dataset is transformed further into flat features. This mainly includes process in which we create new features from the existing features. We use the Count Vector matrix notation where we check variance and drop some of the features based on the threshold it handles.

3) Model Training – The final most step we use is the Model Building where a machine learning model is here trained on the dataset. We implement the models like Naïve Bayes Classifier, Convolution Neural Network and Decision Tree Model and find which model gives best accuracy.

We have finished the first two components and currently working on the third component.

Current Progress:

Initially, we imported the required packages and loaded the dataset as below. And we also concatenated the headline as well as the short description into the attribute named ‘Combined_H&SD’

```
import pandas as pd
import numpy as np
import json
import copy
import string
import re
import nltk
import string
from nltk.stem import PorterStemmer
from nltk.corpus import stopwords
nltk.download('popular')
from wordcloud import WordCloud

from sklearn.model_selection import train_test_split
from sklearn.feature_extraction.text import CountVectorizer
from sklearn.feature_extraction.text import TfidfVectorizer
from sklearn.preprocessing import LabelEncoder
from sklearn.feature_selection import VarianceThreshold
from imblearn.over_sampling import SMOTE

from sklearn.tree import DecisionTreeClassifier

import matplotlib.pyplot as plt
np.random.seed(0)
```

+ Code + Text Connect Ed

```
import matplotlib.pyplot as plt
np.random.seed(0)
```

```
[nltk_data] Downloading collection 'popular'
[nltk_data]    |
[nltk_data]    | Downloading package cmudict to /root/nltk_data...
[nltk_data]    |   Package cmudict is already up-to-date!
[nltk_data]    | Downloading package gazetteers to /root/nltk_data...
[nltk_data]    |   Package gazetteers is already up-to-date!
[nltk_data]    | Downloading package genesis to /root/nltk_data...
[nltk_data]    |   Package genesis is already up-to-date!
[nltk_data]    | Downloading package gutenberg to /root/nltk_data...
[nltk_data]    |   Package gutenberg is already up-to-date!
[nltk_data]    | Downloading package inaugural to /root/nltk_data...
[nltk_data]    |   Package inaugural is already up-to-date!
[nltk_data]    | Downloading package movie_reviews to
[nltk_data]    |     /root/nltk_data...
[nltk_data]    |   Package movie_reviews is already up-to-date!
[nltk_data]    | Downloading package names to /root/nltk_data...
[nltk_data]    |   Package names is already up-to-date!
[nltk_data]    | Downloading package shakespeare to /root/nltk_data...
[nltk_data]    |   Package shakespeare is already up-to-date!
[nltk_data]    | Downloading package stopwords to /root/nltk_data...
[nltk_data]    |   Package stopwords is already up-to-date!
[nltk_data]    | Downloading package treebank to /root/nltk_data...
[nltk_data]    |   Package treebank is already up-to-date!
[nltk_data]    | Downloading package twitter_samples to
[nltk_data]    |     /root/nltk_data...
```

Here, we are reading the news dataset from the json file we have.

Reading News Dataset from News_Category_Dataset_v2.json

```
[ ] News_Dataset = pd.read_json('News_Category_Dataset_v2.json', lines=True)
```

```
[ ] News_Dataset.head(6)
```

	category	headline	authors	link	short_description	date
0	CRIME	There Were 2 Mass Shootings In Texas Last Week...	Melissa Jeltsen	https://www.huffingtonpost.com/entry/texas-ama...	She left her husband. He killed their children...	2018-05-26
1	ENTERTAINMENT	Will Smith Joins Diplo And Nicky Jam For The 2...	Andy McDonald	https://www.huffingtonpost.com/entry/will-smit...	Of course it has a song.	2018-05-26
2	ENTERTAINMENT	Hugh Grant Marries For The First Time At Age 57	Ron Dicker	https://www.huffingtonpost.com/entry/hugh-gran...	The actor and his longtime girlfriend Anna Ebe...	2018-05-26
3	ENTERTAINMENT	Jim Carrey Blasts 'Castrato' Adam Schiff And D...	Ron Dicker	https://www.huffingtonpost.com/entry/jim-carre...	The actor gives Dems an ass-kicking for not fi...	2018-05-26
4	ENTERTAINMENT	Julianna Margulies Uses Donald Trump Poop Bags...	Ron Dicker	https://www.huffingtonpost.com/entry/julianna-...	The "Dietland" actress said using the bags is ...	2018-05-26
5	ENTERTAINMENT	Morgan Freeman 'Devastated' That Sexual Harass...	Ron Dicker	https://www.huffingtonpost.com/entry/morgan-fr...	"It is not right to equate horrific	2018-05-26

Combining of column's headline's and short description into a single attribute.

Also, the headline is cleaned as below:

incidents ..

Combining column's Headline & Short Description into Combined_H&SD

```
[ ] News_Dataset['Combined_H&SD']=News_Dataset['headline']+News_Dataset['short_description']
```

```
[ ] stemmer = PorterStemmer()
```

```
[ ] def process_text(value):
        no_punc=[char for char in value if char not in string.punctuation]
        new1=''.join(no_punc)
        new2=[stemmer.stem(word) for word in new1]
        new3=''.join(new2)
        return[word for word in new3.split()if word.lower()not in stopwords.words('english') ]
```

```
[ ] News_Dataset['Combined_H&SD'].head()
```

```
0    There Were 2 Mass Shootings In Texas Last Week...
1    Will Smith Joins Diplo And Nicky Jam For The 2...
2    Hugh Grant Marries For The First Time At Age 5...
3    Jim Carrey Blasts 'Castrato' Adam Schiff And D...
4    Julianna Margulies Uses Donald Trump Poop Bags...
Name: Combined_H&SD, dtype: object
```

Splitting of the Dataset into train, test and development.

Training data is used for training out the model and Development data for tuning and checking the hyper parameters and test data to check how the model is performing.

After Processing of column- Combined_H&SD using Stemmer

```
[ ] News_Dataset['Combined_H&SD'].head(5).apply(process_text)
```

```
0    [2, Mass, Shootings, Texas, Last, Week, 1, TVS...
1    [Smith, Joins, Diplo, Nicky, Jam, 2018, World,...
2    [Hugh, Grant, Marries, First, Time, Age, 57The...
3    [Jim, Carrey, Blasts, Castrato, Adam, Schiff, ...
4    [Julianna, Margulies, Uses, Donald, Trump, Poo...
Name: Combined_H&SD, dtype: object
```

Splitting of News Data set Into Train, Test & Development

```
[ ] train_title, test_title, train_category, test_category = train_test_split(News_Dataset['Combined_H&SD'],News_Dataset['cate
    train_title, devp_title, train_category, devp_category = train_test_split(train_title,train_category)
```

Number of Records for train, Test & Development

```
[ ] print("Training Records    : ",len(train_title))
    print("Developement Records: ",len(devp_title))
    print("Testing Records     : ",len(test_title))
```

```
Training Records    :  112979
Developement Records:  37660
Testing Records     :  50214
```

Here, we visualize the data combined using WordCloud.

Visualized the Data of combined_H&SD using WordCloud

```
[ ] train_text = " ".join(train_title)
    wordcloud = WordCloud().generate(train_text)
    plt.figure()
    plt.subplots(figsize=(50,50))
    wordcloud = WordCloud(
        background_color="Black",
        max_words=len(train_text),
        max_font_size=30,
        relative_scaling=.5).generate(train_text)
    plt.imshow(wordcloud)
    plt.axis("off")
    plt.show()
```

```
<Figure size 432x288 with 0 Axes>
```

```
[ ]
```

```
<Figure size 432x288 with 0 Axes>
```

The next step is Data Preprocessing. Here, we are vectorizing the data using a Bag of Words (BOW)

Data Preprocessing

Vectorizing the data using Bag of words (BOW)

```
[ ] tokenizer = nltk.tokenize.RegexpTokenizer(r"\w+")
    stop_words = nltk.corpus.stopwords.words("english")
    c_vectorizer = CountVectorizer(tokenizer=tokenizer.tokenize, stop_words=stop_words)
```

```
[ ] c_vectorizer.fit(iter(train_title))
    X_train = c_vectorizer.transform(iter(train_title))
    X_devp = c_vectorizer.transform(iter(devp_title))
    X_test = c_vectorizer.transform(iter(test_title))
```

```
/usr/local/lib/python3.6/dist-packages/sklearn/feature_extraction/text.py:507: UserWarning: The parameter 'token_pattern'
  warnings.warn("The parameter 'token_pattern' will not be used"
```

Encoding the column categories are done using the label encoder for all the categories. After that the features are reduced.

If we clearly see the features before reduction are 126219 and after reduction are 3380 The threshold which we took is 0.001

Categorical Encoding of category Column using Label Encoder

```
[ ]  encoder = LabelEncoder()
     encoder.fit(train_category)
     Y_train = encoder.transform(train_category)
     Y_devp = encoder.transform(devp_category)
     Y_test = encoder.transform(test_category)
```

Feature Reduction

```
[ ]  print("Number of features before reduction : ", X_train.shape[1])
     selection = VarianceThreshold(threshold=0.001)
     X_train_whole = copy.deepcopy(X_train)
     Y_train_whole = copy.deepcopy(Y_train)
     selection.fit(X_train)
     X_train = selection.transform(X_train)
     X_devp = selection.transform(X_devp)
     X_test = selection.transform(X_test)
     print("Number of features after reduction : ", X_train.shape[1])
```

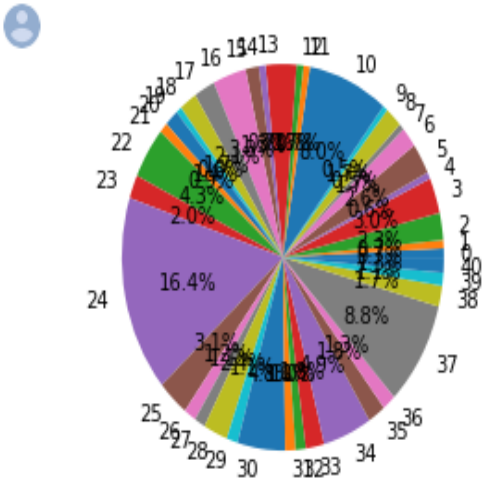
```
Number of features before reduction :  126219
Number of features after reduction :  3380
```

In Data Sampling,

We have counted the number of total labels and plotted them using a pie chart distribution model.

Sampling the data

```
[ ]  labels = list(set(Ytr))
     counts = []
     for label in labels:
         counts.append(np.count_nonzero(Y_train == label))
     plt.pie(counts, labels=labels, autopct='%1.1f%%')
     plt.show()
```

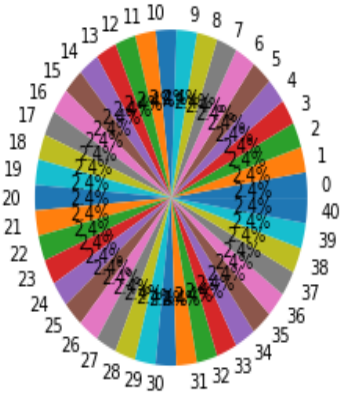


We can clearly have a look that the class labels are here not distributed uniformly.

So, we had to use SMOT and then over sampled the classes which are lowest in the number. This is done because we can samples can be equally distributed.

```
[ ]  sm = SMOTE(random_state=42)
     X_train, Y_train = sm.fit_sample(X_train, Y_train)
     labels = list(set(Y_train))
     counts = []
     for label in labels:
         counts.append(np.count_nonzero(Y_train == label))
     plt.pie(counts, labels=labels, autopct='%1.1f%%')
     plt.show()
```

```
/usr/local/lib/python3.6/dist-packages/sklearn/utils/deprecation.py:87: FutureWarning
  warnings.warn(msg, category=FutureWarning)
```



Remaining parts of the projects:

We have nearly finished like the 2 components of the divided project. We are done with the dataset preparation and then feature engineering.

We are left out with the Model training where we build a machine learning model on the dataset that is trained.

We will implement the models like Naïve Bayes Classifier, Convolution Neural Network and Decision Tree Model and find which model gives best accuracy.

Team-work division:

- Geetanjali Makineni –
  - Dataset Preparation:
    - Combining Column Headline & Short Description
    - Stemming

- Feature Engineering
    - Text Processing
    - Sampling
- Model Selection
    - Decision Tree Model
    - Random Forest Model
- Loss & Accuracy

- Akhil Teja Kanugolu –
    - Dataset Preparation:
        - Splitting Train, Test, Development
        - Visualization of Combined H&SD
    - Feature Engineering
        - Vectorization
        - Feature Reduction based on Threshold
    - Model Selection
        - Multinomial Naïve Bayes Model
        - Support Vector Classification
        - CNN
    - Hosting Static Webpage

Challenges Facing:

Since, we used a dataset with around 200K records, it can take more time to run the models.

More time consumed when pre-processing the data and cleaning it up.

Visualizing the data using Word Cloud is a little complex.

Future Work:

- Here, we can also use some other machine learning as well as deep learning algorithms on our data set and we can see which model can give the better accuracy.

- We can here also induce some more other methods in feature engineering as well as parameters and can check how it might affect the accuracy of the model.