**Introduction**

This project uses a dataset of labeled tweets to train a logistic regression model in PyTorch for sentiment analysis. The model takes 12 different features, including sentiment scores and word count, and outputs a probability of the tweet being positive. The model is trained on a training set and evaluated using accuracy and F1-score on a test set.

**Discussion**

Text

Description automatically generated

Importing the required libraries to run the code and analyze it.

A picture containing icon

Description automatically generated

Setting the encoding type to the variable. Defining this as this is used by all the tsv files while loading them on to the system.

Graphical user interface, text

Description automatically generated

This function or method that is defined is used to clean the data using the regular expressions. This function clears all the data like extra spaces, emojis, multiple spaces, square brackets, special symbols and punctuation marks. This also converts the data to lowercase.

Text

Description automatically generated

This method is used to read the data from the file using the r mode and encoding format as utf-8.

Text, letter

Description automatically generated

Reading the text files and label files.

Text

Description automatically generated with low confidence

Train tweets file sample data.

Graphical user interface, text, application

Description automatically generated with medium confidence

Train labels sample data

Text

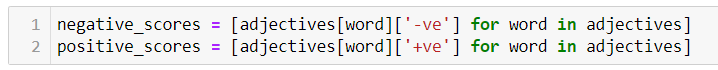
Description automatically generated

Here in this we have defined the function to read the lexicon files and converting it to a dictionary and returning it to the calling function.

Text

Description automatically generated

These are all the lexicon files that are used for the project.



Assigning all the positive and negative values to the lists from the adjectives lexicon file which is named as 2000.tsv.



Getting the names of all the words in the 2000.tsv file.

Text

Description automatically generated

Plotting a bar graph to show the positive and -ve scores for the given word.

Chart

Description automatically generated

The wording at the bottom is not clear due to the screen size that is present.

Graphical user interface, text

Description automatically generated

Adding all the lexicon values to the list.

Graphical user interface, text, application

Description automatically generated

This stores the list of file names and contents length of all the files.

Text

Description automatically generated

This prints the below graph

Chart, bar chart

Description automatically generated

Graphical user interface, text, application

Description automatically generated

This is all the lexicons combined from all the files.

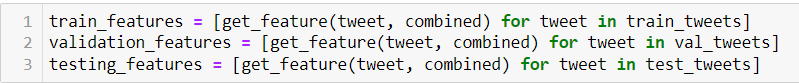
Text

Description automatically generated

Text

Description automatically generated

The above code will store the features on feature\_set variable. The first nine are taken from the lexicon files and the other three features manually added by doing some actions on dataset.



Loading the data on the variables.

A screenshot of a computer

Description automatically generated

Preparing the framework using pytorch to create the data to provide data as input.

Text

Description automatically generated

The above class is defined to provide a base for performing the logistic regression for the tweets.

Text, letter

Description automatically generated

This provides a reference for the class that is defined above. The object that is generated is calling the logistic regression functions and performing the training and test to find the accuracy and f1\_score.

**Output Analysis:**

Text

Description automatically generated

The Accuracy of the model is 48.3% where as the f1 score is 31.5%

**References**

<https://numpy.org/>

<https://pandas.pydata.org/>

<https://matplotlib.org/>

<https://scikit-learn.org/stable/>

<https://pytorch.org/get-started/locally/>

<https://scikit-learn.org/stable/modules/generated/sklearn.feature_extraction.text.CountVectorizer.html>

<https://stackoverflow.com/questions/34093264/python-logistic-regression-beginner>

<https://pytorch.org/docs/stable/optim.html>

<https://stackoverflow.com/questions/22540449/how-can-i-rotate-a-matplotlib-plot-through-90-degrees>

<https://stackoverflow.com/questions/53975717/pytorch-connection-between-loss-backward-and-optimizer-step>

<https://pytorch.org/docs/stable/generated/torch.nn.BCELoss.html>

<https://scikit-learn.org/stable/modules/generated/sklearn.metrics.f1_score.html>

https://stackoverflow.com/questions/42471082/how-to-find-out-the-accuracy