**REPORT ON NEWS TWEET SANITIZED**

**Project Overview**

In this project, we will explore the basics of natural language processing (NLP), the project's goal is to create a system that cleans up news tweets. On social media platforms, the objective is to screen out material that could be relevant or irrelevant tweets, reduce the spread of erroneous information, and encourage the distribution of factual news.

Throughout this project, the following abilities will be exercised in NLP

* How to rapidly analyze and clean up big text datasets with pandas.
* How to preprocess text data using fundamental NLP techniques and produce features.
* How to train, assemble, fit, and assess deep learning models in Keras.
* Ways to classify unlabeled text data.

**STEP 1: PROBLEM IDENTIFICATION**

* 1. **Problem statement**

To show the tweeted text that could be relatable or irrelevant newsposted.

* 1. **Context**

To filter and sanitize news tweets, certain conditions and factors must be taken into account. This is referred to as the context for news tweet sanitization. In contrast to the filtered news offered by media outlets, Twitter is an outstanding repository of uncorrected discussions, views, and news events that are directly tweeted by the persons themselves.

* 1. **Success Indicators**

Identify the themes of the tweets with at least 75% accuracy.

* 1. **Solution Approach**

Using the appropriate keywords, gather information from Twitter.

The data will be cleaned and preprocessed using Natural Language Processing (NLP) methods.

Distributed computing will be utilized to do this and logistic regression and naive Bayes LSTM models will be trained and tested.

* 1. **Constraints**

For now, we observed the following points are constraints

* A limited dataset of news tweets can be considered a constraint.
* When dealing with the casual language, sarcasm, acronyms, or accents frequently employed in tweets, NLP approaches may encounter difficulties.
* It might be difficult to ensure appropriate comprehension and interpretation of such subtleties.
  1. **Stakeholders**

Professor and my team members

* 1. **Deliverables**
* Final Report
* Final Presentation
* code in Jupyter Notebooks for every phase of the process
* Model Building

**STEP 2: DATA COLLECTION**



**STEP 3: DATA WRANGLING:**

The dataset needed to be cleaned up and converted into an effectively visualizable format before the data could be displayed. Finding a productive method of reading and cleaning a dataset containing 10324 rows is necessary. To do that, data from a CSV file might be read into a pandas data frame. This is how the data was read and cleaned in our project.

The dataset was shaped into a format necessary for visualization using the following fundamental NLP techniques:

1. Reading CSV file into a pandas data frame.

2. Remove any duplicates.

3. Eliminate any missing values.

4. Tokenize (reduce the tweets to their words)

5. Change the letters to lowercase.

6. Do not use punctuation

7. Get rid of stop words

8. Get rid of URLs, Twitter, and other abbreviations

The following strategy was used to carry out the aforementioned steps:

1. Created a second custom function that processes the data using all of the cleaning procedures discussed previously. A white screen with text

Description automatically generated

**STEP 4: MODELLING**

For modeling news tweet sanitization the following tasks should be done

1. The dataset should be tokenized, padded, and sequenced.

2. Separate the dataset into test and training sets.

3. prepared sequential model.

4. Evaluating models

**4.1 Tokenize, pad, and Sequence**

Initially being put into a deep learning architecture, the dataset was in this stage tokenized, padded, and sequenced as needed.

A screenshot of a computer program

Description automatically generated

**4.2 Split the data into training and test datasets**

After preprocessing the data and getting it ready for training, I divided the dataset into training and test sets with a ratio of 70% to 30%, respectively. Specifically, 30% of the dataset was set aside for testing the model's accuracy after it had been trained on 70% of the data.

**4.3 Model**

After dividing the dataset into training and testing we created a sequential model with Relu and sigmoid activation functions and used kerasclassifers. Later for tuning the model we used hyperparameters. Our designs employed dropout layers to prevent overfitting. We used word embeddings as an input to the hidden layers and have an embedding layer as their first layer.

A screenshot of a computer program

Description automatically generated

In a sequential model, the activation function is by default set to "Relu and sigmoid". When building the model, "accuracy" was utilized as the metric, "binary\_crossentropy" as the loss function, and the "Adam" optimizer as the optimizer. When the model was fitted to a training set, 70% of it was used to train the model, and the remaining 30% was utilized for validation. The number of epochs was set to 10. This is not to be confused with the test set that was kept as 30% of the original data.

**4.4 Model Evaluation:**

Finally, we got an accuracy of 0.85% on the test dataset and we plot the ROC curve and confusion matrix.

Let's plot the algorithm's outputs now and examine the accuracy of their predictions over a period of 5 epochs.

A screen shot of a graph

Description automatically generated

**CONCLUSION:**