**PROG8245 – Machine Learning Programming**

**Project: Sentiment Analysis**

**Group: 6**

***Praiselin Lydia Gladston*** *and* ***Sudharsan Tirumal***

**Objective:**

The main objective of our project is to develop an efficient sentiment analysis model capable of accurately categorizing a given text statement into three sentiment categories: neutral, negative, and positive. This project aims to provide valuable insights into public opinions on various topics discussed on Reddit, particularly focusing on subreddits related to recent trends and what people talk about. By achieving this goal, we aim to offer businesses and organizations the tools to understand customer sentiment, thereby aiding in decision-making processes related to brand management, marketing, and customer satisfaction. The subreddits we have considered for our project are 'PoliticsPeopleTwitter', 'trendingsubreddits', 'Discussion' and ‘CasualConversation’.

**1. Overview:**

In the area of social media analytics, sentiment analysis plays a crucial role in understanding public opinion, market trends, and attitudes toward different topics. Our project seeks to leverage machine learning techniques and natural language processing to analyze and classify the emotions expressed in Reddit headlines and other social media posts. By categorizing these sentiments as positive, negative, or neutral, we aim to uncover underlying sentiments and provide valuable insights into public perception.

Our project employs a systematic approach, beginning with data collection from Reddit using the PRAW (Python Reddit API Wrapper) library. We then preprocess the gathered data, extract relevant features, develop machine learning models, and deploy them through an intuitive user interface. Our methodology includes exploration of various feature extraction techniques such as TF-IDF, Word2Vec, and GloVe, coupled with machine learning models like Naive Bayes, Support Vector Machines (SVM), and Logistic Regression. Based on the results, we have chosen the best model out of all and created a user-friendly graphical interface using Tkinter as its frontend, enabling users to input social media posts and receive real-time sentiment analysis results.

**2. Data Collection:**

The foundation of any data-driven initiative hinges upon the quality and relevance of the gathered data. Reddit serves as the primary data source for our sentiment analysis project, given its vast platform encompassing diverse user-generated content. Our primary objective is to extract pertinent headlines from subreddits revolving around recent trends and everyday conversations.

* **Data Source Selection:** We utilize the PRAW library to programmatically access Reddit's API and extract headlines of posts from relevant subreddits such as 'PoliticsPeopleTwitter', 'trendingsubreddits', 'Discussion' and ‘CasualConversation’ where people gather for a discussion and give out their opinions.
* **Subreddits Selection:** Our selection of subreddits is deliberate, focusing on topics relevant to current trend, everyday conversation, and casual review of any topic to ensure the dataset's relevance to our investigation.
* **Data Collection Process:** We gathered approximately 10000 distinct posts from the chosen subreddits, ensuring diversity and relevance while overcoming challenges such as API rate limitations and content filtering.
* **Lowercasing:** We convert all contents to lowercase to ensure consistency and prevent duplication.
* **Special Character Removal:** Remove special characters like at sign ‘@’, symbol like exclamatory mark ‘!’ etc.

**Challenges faced:** During the data collection phase, we faced several challenges, including limitations imposed by the Reddit API's rate restrictions. This compelled us to employ efficient search strategies and maintain courteous API usage practices. Furthermore, given Reddit's extensive content diversity, it was imperative for us to meticulously filter and prioritize relevant headlines to ensure the focus and quality of our dataset remained high.

**3. Preprocessing:**

In any text analysis endeavor, preprocessing is a vital step, particularly when dealing with unstructured data sourced from platforms like Reddit and other social networks. Cleaning up and standardizing raw text is crucial to prepare it for feature extraction and model training. This section outlines the preprocessing steps taken to ensure the selected Reddit headlines were primed for sentiment analysis.

* **Hashtag and Emoji Removal: Remal of ‘#’ symbol and emojis so that those don’t affect model training or feature extraction.**
* **Stopword Removal:** To remove words that don't contribute much to sentiment analysis, like common terms such as "the," "is," "in," and others. We accomplished this by leveraging NLTK's stop words list for the English language. This step not only helps streamline the dataset but also boosts computing efficiency.
* **Tokenization and Cleaning:** We tokenize each post title and remove non-alphabetic characters, punctuation and digits using NLTK's word tokenize function.
* **Stemming and Lemmatization:** We apply stemming and lemmatization techniques to reduce words to their root forms, ensuring uniformity and improving data quality.

**Challenges Faced:** Challenges include managing slang and idiomatic expressions encountered in social media text and ensuring accurate preprocessing to maintain data integrity.

**4. Data Annotation:**

In our sentiment analysis project, a crucial step is data annotation. Here, we categorize each Reddit headline into one of three sentiment categories: neutral, negative, or positive. This process converts the raw, unstructured text data into a format that's primed for feature extraction and deeper analysis. We annotate using the **SentimentIntensityAnalyzer** from NLTK library.

**Challenges Faced:** Challenges were faced while trying to use pre-trained models from HuggingFace for annotation. There was API request limit per user, within an hour, and could not make frequent calls for the number of post titles to be annotated. Handling headlines with unclear or context-dependent language, where the sentiment was not made apparent, presented another difficulty. To guarantee correct classification, such circumstances needed to be carefully considered.

**5. Feature Extraction:**

In our project, we utilized various feature extraction techniques to transform raw text into numerical representations suitable for machine learning models. These techniques include Bag-of-Words (BoW), Global Vectors for Word Representation (GloVe), and features generated from the GPT-2 model.

* **BoW (Bag-of-words):**

**Relevance:** BoW is known for its simplicity and effectiveness in capturing the frequency of words in a document without considering their order or semantic meaning. sentiment analysis often relies on the frequency of specific words or phrases associated with different sentiments. BoW effectively captures these patterns by representing documents as bags of individual words, making it suitable for this task.  
**Insight:** The BoW model demonstrates robust performance across all sentiment categories, achieving high precision, recall, and F1-scores. It accurately predicts sentiment labels for most instances in the dataset, resulting in an overall accuracy of 82%. This performance underscores the effectiveness of the BoW technique for sentiment analysis tasks.

* **GloVe (Global Vectors for Word Representation):**

**Relevance**: GloVe provides word embeddings, which are numerical representations of words that capture their semantic meanings based on the surrounding context in large text corpora. Words with similar meanings have similar embeddings, allowing GloVe to capture semantic relationships between words.

**Insight**: Contrary to BoW, the GloVe model exhibits relatively poor performance, particularly in terms of precision and recall for the negative sentiment category. While the model performs reasonably well in classifying neutral sentiments, it struggles with accurately identifying negative and positive sentiments. The low recall scores for negative and positive sentiments indicate that the model fails to correctly identify instances of these sentiments, leading to a significant number of false negatives. Overall, the GloVe model's performance suggests that it may not be well-suited for sentiment analysis tasks without further refinement or augmentation with other techniques.

* **GPT-2 (Generative Pre-trained Transformer 2):**

**Relevance**: GPT-2 can be fine-tuned for sentiment analysis tasks, where it learns to associate textual patterns with sentiment labels. By leveraging its contextual understanding, GPT-2 can capture subtle linguistic cues and context-dependent sentiment expressions, thereby enhancing sentiment analysis performance.

**Insight**: GPT-2 is a powerful language model capable of generating coherent text based on given prompts. However, its performance in sentiment analysis may be limited by its pre-training on a diverse range of text data, which may not specifically optimize for sentiment-related features. While GPT-2 can capture semantic meanings and contextual information, its performance in sentiment analysis may vary depending on the complexity and subtlety of sentiment expressions in the text data.

**Challenges Faced:** The LBFGS optimizer for logistic regression reached its maximum number of iterations while performing feature extraction using GPT-2, suggesting potential convergence issues that could impact performance.

**6. Model Training:**

We develop machine learning models including Naive Bayes, Support Vector Machines (SVM), and a TensorFlow-based neural network to classify sentiments expressed in Reddit posts headlines. We evaluated model performance using metrics such as accuracy, precision, recall, and F1-score. We have split the existing data into 80% training and 20% testing dataset. Through utilizing different models, we were able to comprehend various methods of performing sentiment analysis. Reason for selecting these models:

* **Naïve Bayes:** Naive Bayes models provide probabilistic predictions based on the features of the input data. In sentiment analysis, they can effectively capture the relationships between words and sentiments, making them suitable for tasks where feature independence assumptions hold reasonably well.
* **SVM:** SVMs excel in finding the optimal hyperplane that separates different classes in feature space. In sentiment analysis, they can effectively capture complex relationships between words and sentiments, leading to accurate classification.
* **Neural Network:** Neural networks are composed of interconnected layers of nodes that can learn hierarchical representations of data. In sentiment analysis, they can learn intricate patterns and dependencies in text data, allowing them to capture the semantic meanings of words and phrases and their relationships to sentiments.

**7. Model results analysis:**

**7.1 Naive Bayes Classifier:**

* Accuracy: 0.79
* Precision (weighted avg): 0.78
* Recall (weighted avg): 0.79
* F1-score (weighted avg): 0.78

**Analysis:** The Naive Bayes classifier demonstrates solid overall performance with an accuracy of 0.79. It shows balanced precision, recall, and F1-score across all sentiment categories, though slightly lower for negative sentiment. Naive Bayes is known for its simplicity and efficiency, making it suitable for text classification tasks.

**7.2 Support Vector Machine (SVM):**

* Accuracy: 0.81
* Precision (weighted avg): 0.81
* Recall (weighted avg): 0.81
* F1-score (weighted avg): 0.81

**Analysis:** The SVM classifier performs slightly better than Naive Bayes with an accuracy of 0.81. It shows strong precision, recall, and F1-score across all sentiment categories, indicating its effectiveness in classification tasks. SVMs are known for their flexibility in handling both linear and non-linear classification tasks, making them suitable for sentiment analysis.

**7.3 Neural Network (TensorFlow):**

* Accuracy: 0.11
* Loss: 11.6876

**Analysis:** The neural network implemented using TensorFlow shows significantly lower performance compared to the other two models with an accuracy of only 0.11. It seems that the neural network may not be effectively capturing the patterns in the data or may be overfitting to the training data. Further analysis and adjustments to the model architecture, hyperparameters, or training process may be necessary to improve its performance.

**8. Conclusion:**

Our project on sentiment analysis of Reddit headlines has successfully employed machine learning techniques to gain valuable insights into public opinions and attitudes expressed across various subreddits. Through meticulous data collection, preprocessing, and feature extraction, we developed and evaluated multiple machine learning models for sentiment classification. Our findings highlight the effectiveness of traditional models such as Naive Bayes and Support Vector Machines (SVM) in accurately categorizing sentiments, with SVM exhibiting slightly superior performance. However, the implementation of a neural network using TensorFlow showed lower performance, suggesting the need for further refinement. Overall, our project underscores the importance of selecting appropriate models and techniques in sentiment analysis tasks and contributes to informed decision-making processes in areas such as brand management and customer satisfaction. Continued research in sentiment analysis methodologies will enhance our ability to extract meaningful insights from social media data.

**10. References:**

1. A complete guide to Web Scraping Reddit with Python (March 2023) <https://medium.com/geekculture/a-complete-guide-to-web-scraping-reddit-with-python-16e292317a52>  
2. Sentiment Analysis on Reddit News Headlines with Python’s Natural Language Toolkit (NLTK) <https://www.learndatasci.com/tutorials/sentiment-analysis-reddit-headlines-pythons-nltk>

3. NLTK Project NLTK Sentiment Analysis (using SentimentIntensityAnalyzer) <https://www.nltk.org/howto/sentiment.html>

4. A Gentle Introduction to the Bag-of-Words Model <https://machinelearningmastery.com/gentle-introduction-bag-words-model>  
5. Word Embedding using GloVe <https://www.hackersrealm.net/post/word-embedding-using-glove-python>  
6. How to use GPT-2 Transformer for Feature Extraction <https://community.wolfram.com/groups/-/m/t/2925480>

7. Sentiment Analysis with Naïve Bayes Algorithm <https://medium.com/@zubairashfaque/sentiment-analysis-with-naive-bayes-algorithm-a31021764fb4>  
8. Sentiment Analysis using SVM <https://medium.com/scrapehero/sentiment-analysis-using-svm-338d418e3ff1>  
9. An easy tutorial about Sentiment Analysis with Deep Learning and Keras <https://towardsdatascience.com/an-easy-tutorial-about-sentiment-analysis-with-deep-learning-and-keras-2bf52b9cba91>