**Assignment Task**

**Rise 11 Technologies for ML/AI Intern Position**

**Problem**

The people of Rise11 are very active on social media and post their opinions on social media on a regular basis. You are provided a dataset featuring some of their social media text posts along with their sentiment (0,1,2). Your task involves crafting a NLP pipeline with the precision to accurately classify these social media texts.

**Summary and Logic of the project**

About the dataset

1. Overview of the Dataset: - About 40,000 opinions from social media outlets make up the dataset. Carefully chosen, it encompasses a wide variety of feelings that people have to say. These opinions are a reaction to the introduction of eleven unique technologies.

2. Data Representation: - A user-generated opinion text is associated with each data point in the dataset. These views are the subjective perspectives of those who are having a discussion about the mentioned technology.

3. Sentiment Labelling:- The sentiment classes (0, 1, and 2) are applied to the dataset. These labels suggest possible negative, neutral, and positive attitudes by grouping opinions into distinct sentiment categories.

4. Project Objective:- Using the dataset to build a solid sentiment analysis model is the main objective. The goal of this approach is to appropriately categorise viewpoints into the pre-established sentiment classifications.

5. The Value of Recognising Public Opinion: It is essential to comprehend how the general public views and responds to the developments in the 11 technologies.Sentiment research insights give stakeholders useful information about how the market is receiving the product and might point up areas that could have improvement.

Why can’t we use other models like Logistic Regression, SVM & Naive Baiyes Models ?

In comparison to more conventional models like logistic regression or SVM, using LSTM (Long Short-Term Memory) models for sentiment analysis in a multi-class classification task can have benefits. The following justifies why LSTM might be favoured:

1.Sequential Data Handling: - Sentiment analysis frequently entails examining word sequences in order to get contextual data. Being a particular kind of recurrent neural network (RNN), long-sequence transitive memories (LSTMs) are excellent at processing sequential input, which enables them to capture word dependencies and links in sentences.

2. Semantic Understanding: - LSTMs are able to comprehend word meaning in context and capture semantic links. Such intricate linkages in language may be difficult for conventional models, such as logistic regression, to understand, particularly when working with longer sequences.

3.Input Length Variable: - The length of sentences in social media data might fluctuate. Due to the inherent variety in the length of user-generated thoughts, LSTMs are able to manage sequences of varying lengths. Typically, fixed-size input vectors are expected by SVMs and logistic regression.

4.Feature Extraction: - While logistic regression and SVMs frequently rely on manually created features, LSTMs automatically extract pertinent features from the input data during training. Tasks involving natural language processing greatly benefit from this feature extraction capability.

5.Contextual Information: - By taking into account the context in which words appear, LSTMs are able to capture the temporal connections between words. This is important for sentiment analysis since words that come before might have an impact on the sentiment expressed in a sentence.

6.Non-linearity: - LSTMs are capable of modelling intricate, non-linear relationships, and sentiment analysis jobs are by nature non-linear. Since logistic regression is a linear model, it may not be able to fully represent the nuances of language-based sentiment.

7. Achievement with Big Datasets: - When trained on huge datasets, deep learning models—including LSTMs—often perform exceptionally well. By utilising the expressive capacity of deep neural networks, LSTMs have the potential to surpass conventional models in sentiment analysis projects involving large volumes of social media data.

Even though LSTMs provide benefits, it's important to take into account things like data size, model complexity, and processing resources. Simpler models, such as logistic regression or SVMs, may be enough in some situations, particularly when working with smaller datasets or when the results' interpretability is crucial. The model selection is based on the specific characteristics of the sentiment analysis task and the available resources.

Using TenserFlow rather than Pytorch

1. Sequential Data Handling: - Because LSTMs are skilled at processing sequential data, they can be used to analyse text data, such as opinions found on social media.

2. TensorFlow Sequential Model API: - The Sequential Model API in TensorFlow makes the design and training of LSTM networks easier.

3. Integration with TensorFlow Hub: - TensorFlow and TensorFlow Hub work together flawlessly, making it simple to incorporate models or pre-trained word embeddings into LSTM structures.

4. GPU Acceleration: TensorFlow offers GPU acceleration, which streamlines the training procedure and makes managing intricate LSTM networks easier.

5. Documentation and Community: - TensorFlow's robust documentation and vibrant community offer helpful resources for troubleshooting and optimisation during model deployment

6. TensorBoard for Visualisation: - TensorBoard in TensorFlow helps to monitor important metrics and visualise the training process, which is essential to comprehending the performance of LSTM models.

7. Compatibility with TensorFlow Extended (TFX): - TensorFlow Extended (TFX) facilitates the management and deployment of TensorFlow models, including sentiment analysis models based on LSTM, from start to finish.

8. Transfer Learning with Pre-trained LSTMs: - TensorFlow facilitates the adaption and fine-tuning of pre-trained LSTM models for particular sentiment analysis applications.

In conclusion, TensorFlow and LSTM models work well together to handle sequential data efficiently, make implementation simple, and offer strong community and tool support for sentiment analysis of social media viewpoints.

What is LSTM(Long Short-Term Memory) Model ?

1. Progression Analysis: Because LSTMs are built to process sequential data, they work especially well for natural language processing (NLP) jobs where interpreting a text depends on the sequence in which its words are used.

2. Memory Cells and Gates: - LSTMs have memory cells with input, output, and forget gates installed in them. By controlling the input flow, these gates enable the model to gradually forget or remember certain pieces of information.

3. Contextual Understanding: - LSTMs are able to comprehend word meanings in the context of the complete sequence because they are able to capture long-term dependencies and contextual information. For NLP tasks like sentiment analysis, this is essential.

4. Bidirectional Processing: - By processing sequences in both forward and backward orientations, bidirectional LSTMs improve context understanding and their capacity to capture dependencies from both ends of the sequence.

5. Embedding Layer: - Tokens or sentences are transformed into high-dimensional vectors by feeding input sequences through an embedding layer. This layer aids in the model's acquisition of meaningful word representations.

6. Use in Sentiment Analysis: - Long Short-Term Memory (LSTM) systems are extensively employed in sentiment analysis assignments, where they evaluate word sequences to identify subtleties in sentiment. By taking into account word dependencies and context, the model is able to accurately represent the sentiment conveyed in a sentence.

7. Training with BPTT: - Backpropagation through time (BPTT) is used to train LSTMs so that the model can learn from the complete sequence. For long-range dependencies in sequential data to be captured, this training method is essential.

8. Optimisation Techniques: - To deal with issues like exploding gradients, optimisation techniques like gradient clipping are frequently used during training. These methods guarantee steady and effective learning while the trainee is being trained.

These bullet points offer a succinct summary of the salient features and practical uses of LSTM models in natural language processing (NLP), highlighting their potency in identifying sequential dependencies and comprehending contextual data.

Why Is LSTM used ?

1. Sequential Data Handling: LSTMs are designed for processing sequential data, making them well-suited for tasks like sentiment analysis on text, where the order of words in a sentence is crucial.

2. Capturing Long-Term Dependencies: LSTMs excel at capturing long-term dependencies in sequences, allowing them to understand the context of words and phrases, which is vital for sentiment analysis.

3. Memory Cells: The memory cells in LSTMs enable the model to selectively remember and forget information, facilitating the retention of relevant sentiment-related context.

4. Contextual Understanding: LSTMs have the ability to understand the meaning of words in the context of the entire sequence, making them effective in discerning sentiment nuances expressed across a sentence.

5. Effective in NLP Tasks: LSTMs are widely used in various NLP tasks, including sentiment analysis, due to their capability to capture complex relationships within sequential data.

6. Bidirectional Processing: Bidirectional LSTMs process sequences in both directions, enhancing their ability to capture dependencies from both ends of the sequence, which is beneficial in understanding sentiment context.

7. Embedding Layer Integration: LSTMs seamlessly integrate with embedding layers, allowing the model to learn meaningful representations of words or tokens, contributing to improved sentiment analysis performance.

8. Application in Transfer Learning: LSTMs can be effectively used in transfer learning scenarios, leveraging pre-trained models on large datasets to enhance performance on sentiment analysis tasks with limited labeled data.

In summary, LSTMs are a powerful choice for sentiment analysis in NLP due to their sequential data handling, ability to capture long-term dependencies, and effective contextual understanding, making them well-suited for tasks involving understanding sentiment in textual data.

Shortcomings of LSTM

1. Computational Intensity: LSTMs can be computationally intensive, requiring significant resources for training and inference, especially with large datasets.

2. Difficulty in Capturing Global Dependencies: LSTMs may struggle to capture global context in extremely long sequences, potentially leading to information loss.

3. Limited Handling of Out-of-Vocabulary Words:LSTMs might face challenges with out-of-vocabulary words or rare terms not encountered during training.

4. Risk of Overfitting: LSTMs, particularly with smaller datasets, are prone to overfitting, necessitating careful hyperparameter tuning and regularization.

5. Difficulty with Sarcasm and Contextual Ambiguity: LSTMs may have difficulty understanding sarcasm and handling contextual ambiguity, potentially affecting sentiment predictions in nuanced situations.

Preprocessing

1. Loading SpaCy Model: You are using SpaCy, a natural language processing library, and loading the English language model (`en\_core\_web\_sm`). This model is pre-trained on a large corpus and includes word vectors, part-of-speech tagging, and lemmatization capabilities.

2. Preprocessing Function: The `preprocess\_phrase` function takes a DataFrame (`document`) as input.

3. Iterating Over Text Data: It appears that your DataFrame has a column named 'text'. The function iterates over the values in this column.

4. Tokenization and Lemmatization: For each text in the 'text' column, you use SpaCy to tokenize the text into individual words (`token.is\_alpha` checks if the token is an alphabetic character) and extract the lemmatized form of each word (`token.lemma\_`).

5. Filtering Non-Alphabetic Tokens: The condition `if token.is\_alpha` ensures that only alphabetic tokens are considered. This filtering step removes any non-alphabetic characters or punctuation marks.

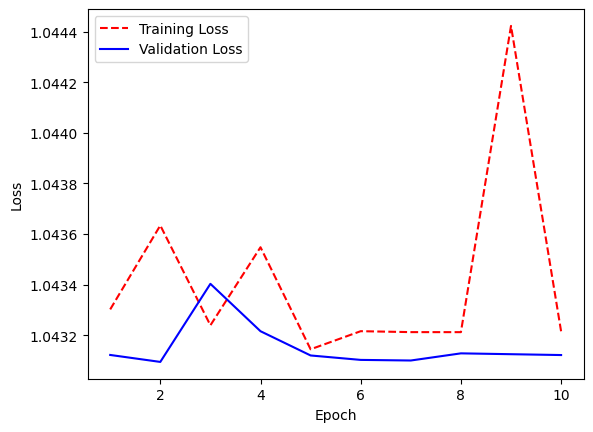
6. Joining Lemmatized Tokens: The lemmatized tokens are then joined back into a single string using `" ".join(result)"`. This creates a preprocessed version of the original text, where words are represented by their lemmatized forms and separated by spaces.

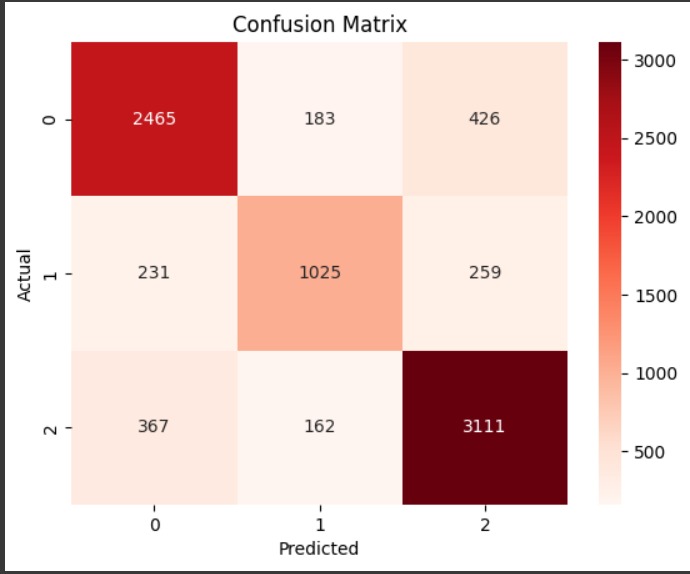
7. Collecting Preprocessed Texts: The preprocessed text for each original text is appended to the `preprocessed\_texts` list.

8. Returning Preprocessed Texts: Finally, the function returns the list of preprocessed texts.

This preprocessing step is common in natural language processing (NLP) tasks to convert raw text data into a format that is more suitable for machine learning models. Lemmatization reduces words to their base or root form, and tokenization breaks the text into individual words, which can help in capturing the essence of the text while reducing dimensionality. The output of this function can be used as input features for training a machine learning model, such as a text classification model.

Data Visualisation

1. Validation loss training los. Graph

2.Confusion Matrix

Accuracy

The accuracy which I achieved using this LSTM model is 86.91%

**Alternate Improved Solutions For Future Scope**

So these are basically some of the models that could have been used to improve the solution for the problem

1. BERT (Bidirectional Encoder Representations from Transformers): Developed by Google, BERT is a transformer-based model that considers the entire context of a sentence by looking at words bidirectionally. It is pre-trained on large corpora and has achieved state-of-the-art results in various natural language processing (NLP) tasks. For multi-label sentiment analysis, BERT can capture complex relationships and dependencies within a text, making it suitable for understanding nuanced sentiment expressions.

2. XLNet: Jointly developed by Google and Carnegie Mellon University (CMU), XLNet builds upon the BERT architecture by introducing a permutation language modeling objective. It leverages the advantages of both autoregressive and autoencoding models. XLNet excels in capturing long-range dependencies in text, which can be beneficial for sentiment analysis tasks that require understanding the overall sentiment context.

3. XLM (Cross-lingual Language Model): Developed by Facebook, XLM is designed to handle multilingual tasks. It is pre-trained on multiple languages and can generalize well across different languages. For multi-label sentiment analysis in a multilingual context, XLM could be useful in capturing sentiment patterns across various languages, providing a more inclusive approach to sentiment classification.

4. RoBERTa (Robustly optimized BERT approach): Also developed by Facebook, RoBERTa is an optimized version of BERT with modifications such as dynamic masking and removing the Next Sentence Prediction objective. It has demonstrated improved performance on various benchmarks. RoBERTa can be advantageous for multi-label sentiment analysis due to its robustness and enhanced training techniques.

5. DistilBERT: Developed by Hugging Face, DistilBERT is a smaller and faster version of BERT, designed for resource-efficient applications. It retains much of BERT's performance while reducing computational requirements. For projects with constraints on computational resources, DistilBERT can be a suitable choice for multi-label sentiment analysis, providing a good trade-off between efficiency and performance.

When choosing a model for your specific multi-label sentiment analysis task, consider factors such as the size of your dataset, available computational resources, and the desired balance between accuracy and efficiency. Additionally, fine-tuning and model selection should be based on the characteristics of your sentiment analysis problem.