Tuesday, 1 August 2023

# Run Submission report

Methodology used:

1. *Data Loading and Preprocessing:*

   - The code starts by loading the training and test datasets using `pandas`.

   - It preprocesses the data by converting the labels into a list of lists format, where each label is a separate element in a list.

   - The labels are then transformed into a binary representation using `MultiLabelBinarizer`, converting them into a format suitable for multi-label classification.

2. *Feature Engineering:*

   - The code uses `TfidfVectorizer` to convert the tweet text into numerical representations (TF-IDF vectors).

   - The vectorizer converts the text data into a matrix of numerical features, representing the importance of each word in the text.

3. *Model Creation:*

   - The TensorFlow model is created using the `Sequential` API.

   - It consists of three dense layers with activation functions (`relu` for the hidden layers and `sigmoid` for the output layer).

   - The input layer's dimension is set to the number of features obtained from the TF-IDF vectorizer, and the output layer's dimension is set to the number of unique labels in the dataset.

4. *Model Training:*

   - The model is compiled with `adam` optimizer and `binary_crossentropy` loss, appropriate for multi-label binary classification.

   - The training data (`X_train` and `y_encoded`) is used to train the model with a batch size of 32 and 20 epochs.

- A validation split of 20% is used during training to monitor the model's performance on unseen data.

5. *Model Prediction and Submission:*

- After training the model, it is used to predict the labels for the test dataset.

- The predicted labels are converted back to a list of lists format using the `MultiLabelBinarizer.inverse_transform`.

- The predictions are then saved to a CSV file in the required submission format with "id" and "preds" columns.

6. *Evaluation Metrics:*

- The code computes the accuracy, macro F1 score, and micro F1 score using `accuracy_score` and `f1_score` functions from `sklearn.metrics`.

- These metrics provide insights into the model's performance on the test dataset.

7. *User Interaction for New Tweets:*

- The code allows the user to input new tweets dataset and predicts their labels using the trained model.

- The predictions are then saved in file submission.csv

It's important to note that the code aims to solve a multi-label, multi-class classification problem where each tweet can have multiple labels (concerns towards vaccines). The TensorFlow model is trained on the provided training data to predict the concerns in the test data and also allows users to input new tweets dataset to get predictions in a civ file.

**note: while the code follows a specific methodology, the accuracy and F1 scores may vary depending on the data and model tuning.