**DistilBERT Model Training Report:-**

**1. Code Analysis**

**->Label Encoding:**

• The label values are encoded into numeric format using LabelEncoder from scikit-learn, which converts the label names into integer labels.

**-> Custom Dataset Class:**

• A CustomDataset class is defined to handle text tokenization and the creation of tensors suitable for input into the DistilBERT model.

• The \_\_getitem\_\_ method in this class uses DistilBERT tokenizer to tokenize the text posts and pads or truncates the sequences to a maximum length of 10 tokens for consistency across inputs.

**-> Model Definition:**

• The DistilBERT model is used for sequence classification. It is initialized from a pre-trained model (distilbert-base-uncased), and the final layer is fine-tuned to output a prediction for each post, corresponding to one of the possible labels.

• The model is fine-tuned to classify the input text into one of the label categories based on the encoded labels.

**-> Training the Model:**

• The model is trained using Trainer from the Transformers library. The training configuration is defined using TrainingArguments:

• 1 epoch is specified to start with, and optimization strategies such as gradient accumulation are set to simulate a larger batch size.

• Weight decay and warm-up steps are used to optimize training efficiency and prevent overfitting.

**->Saving and Reloading the Model:**

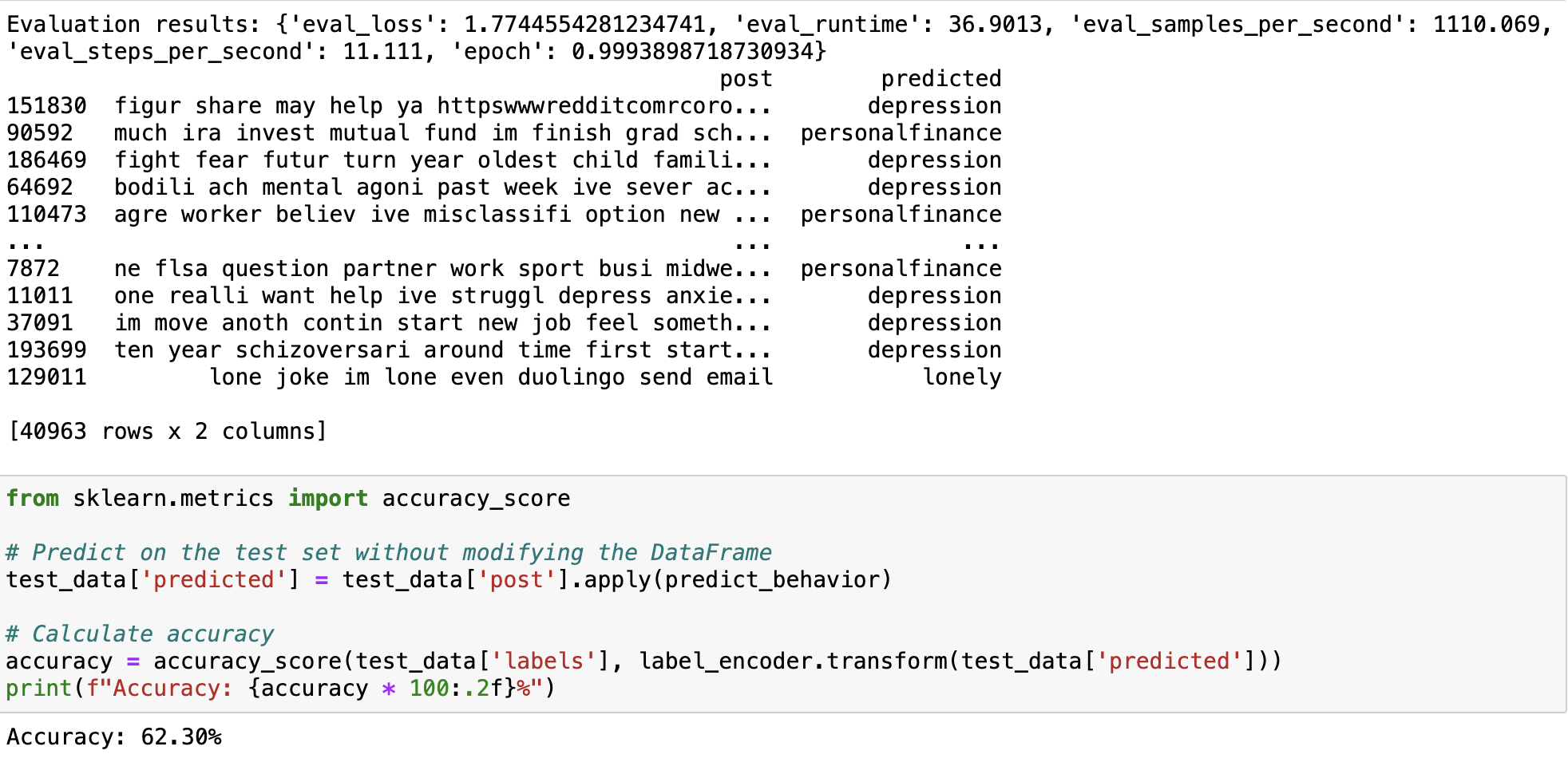
• After training, the model, tokenizer, and label encoder are saved to disk using save\_pretrained and joblib.

• These saved components are reloaded for inference on new data.

**-> Inference:**

• The predict\_behavior function is used to make predictions for new text, returning the predicted label based on the trained model. The function applies the tokenizer to the input text, performs the forward pass through the model, and decodes the output label back to the original label name.

• The predict\_on\_test\_set function processes all posts in the test dataset, applying the predict\_behavior function to each.



**2. Insights**

**->Text Classification Performance:**

• The fine-tuned DistilBERT model provides a strong baseline for text classification tasks. Its ability to classify text into specific categories (labels) is enhanced by using the pre-trained model, which already understands language patterns.

**->Optimization for Large Datasets:**

• Techniques like gradient accumulation and batch size optimization can help when working with larger datasets. We experimented with batch sizes and other training hyperparameters to optimize performance further.

**-> Accuracy Evaluation:**

• After making predictions on the test set, the accuracy of the model is evaluated by comparing the predicted labels with the true labels. The calculated accuracy provides insight into how well the model performs on unseen data.

**3. Error Analysis**

**->Potential Errors:**

• Insufficient Preprocessing: If the text preprocessing is not thorough enough, certain noise (e.g., unprocessed special characters) can interfere with model performance.

• Model Overfitting: If the model is trained for more epochs or on a small dataset, it might overfit. Regularization techniques like weight decay can help mitigate this.

**->Out-of-Vocabulary (OOV) Handling:**

• The model relies on pre-trained embeddings, but it doesn’t handle completely unseen words (OOV) well, which might cause issues with rare or novel words in the text.

**4. Observations**

**->Effect of Hyperparameters:**

• The number of epochs and batch size significantly affect model training. Increasing the number of epochs may improve the model’s understanding but might also lead to overfitting if not controlled properly.

• The maximum token length used in the tokenizer (set to 10 here) could be adjusted for longer or shorter posts to better capture contextual information.

**->Potential Improvements:**

• Fine-tuning the model for more epochs or using larger batch sizes could have lead to improved results.

• Experimenting with learning rates or different pre-trained models (e.g., BERT, RoBERTa) might have provided better performance but only for a few specific tasks.

**->Quality of Predictions:**

• The quality of the text classification predictions can be evaluated by examining the accuracy score.The label accuracy is one of the indications how well the model learned to distinguish between label categories.