AIQoD ML Assessment

Problem Statement:

Develop a machine learning model to predict text classification probabilities across multiple classes using features from text nGrams. The dataset includes 145 features encompassing cryptographic hash, parsing, spatial, and relational information. This multilabel classification task requires handling samples associated with multiple classes. Techniques such as Bag of Words, tf-idf, and word embeddings will be employed, with an explanation of the use of hash values. The process involves training with train.csv, aligning labels from “trainLabels.csv”, testing with “test.csv”, and submitting predictions in the format provided by “sampleSubmission.csv”.

Solution :

1. Data Preprocessing:

Load the Data:

- Utilize pandas to import train.csv and test.csv, generating data frames for processing.

Handle Missing Values:

- Null values are dropped from both training and test datasets.

2. Feature Extraction:

- Feature extraction is based on the transformation of hash values to integers and encoding of categorical variables. The features are then used to train the machine learning model.
- Hash values in object-type columns are converted to integers.
- Features and labels are separated from the merged training data.
- Yes/No columns are encoded using `LabelEncoder`.

3. Model Building:

- **Choose a Model:** Implement a RandomForestClassifier within a MultiOutputClassifier framework to address multilabel classification.
- **Train the Model:** Train using the combined features and labels from train_data and train_label.
- **Cross-Validation:** Essential for ensuring the model's generalizability, though not depicted in the initial steps.

4. Model Evaluation:

- **Evaluation Metrics:** Compute accuracy, F1 score, precision, and recall with sklearn.metrics, appropriate for multilabel classification assessment.

5. Predictions and Submission:

- **Generate Predictions:** Predict the test set probabilities, preparing them for submission.
- **Create Submission File:** Format the predictions to align with the structure in output.csv, associating each probability with the correct sample and label.

6. Model Optimization:

- While not included in the initial outline, hyperparameter tuning or feature selection would involve methods like grid search or random search to enhance the model's performance.

7. Final Thoughts:

Performance Metrics:

- Model Accuracy: 0.7960837272113437
- Model F1 Score: 0.8241157133124756
- Model Precision: 0.9090684604063571
- Model Recall: 0.7977598008711886

Code file :

https://drive.google.com/file/d/1BuxnTf-MBxTFH46p-jIddpR5sRfdzFty/view?usp=sharing.

Output Csv :

https://drive.google.com/file/d/1Kz8Ba2LvO0beEdV9dv-kcANPwBqDHvDV/view?usp=sharing.