Infrrd Assignment

I have gone through with the provided input data and invested some time to understand and analyze the data. After a lot of analysis and tweaking i finally came with the following outcomes:

1. Exploratory Data Analysis(EDA):

The project began with an initial exploration of the provided TSV files and corresponding images to understand the data structure and preparation. Following this, the data preprocessing phase involved adding appropriate labels to each column to enhance visualization and model interpretability. Additionally, the bounding box information was used to derive box height and width, which added valuable features to improve model training.

Given the imbalanced nature of the dataset, where the '**OTHER**' field was overrepresented, model selection became crucial. I initially employed the **RandomForestClassifier** due to its robustness and ability to handle varied data types. While the results were acceptable, I sought further improvements by researching models better suited for imbalanced datasets. After reviewing several resources and official model documentation, I identified **XGBoost** as a promising alternative, particularly for its ability to handle imbalanced data effectively.

Hyperparameter tuning of the **XGBoost** model yielded results similar to the **RandomForestClassifier**, with **XGBoost** excelling in recall, while **RandomForest** was superior in precision. To leverage the strengths of both models, I implemented a **Voting Classifier** that combined the predictions from both models by averaging their outputs. This approach balanced the precision-recall trade-offs and optimized overall performance.

The process of reviewing documentation and exploring models for imbalanced data was integral to selecting the appropriate solution for this project.

2. Error Analysis:

Once the model was trained, I was able to get the evaluation of the trained model. Following is my error analysis:

High-Performing Fields:

- ssnOfEmployee
- einEmployerIdentificationNumber
- box1WagesTipsAndOtherCompensations
- box2FederalIncomeTaxWithheld
- box3SocialSecurityWages
- box4SocialSecurityTaxWithheld
- employerName
- employerAddressStreet_name
- employerAddressCity
- employerAddressState
- employerAddressZip
- taxYear

The above fields are very high performing, ranging their **F1-score** from **0.9-1.0**. This means that model predictions for these fields are very reliable.

Good-Performing Fields:

- employeeName
- box17StateIncomeTax
- box16StateWagesTips

The above fields are performing great as well but lack because the recall value is not up to the mark ranging their **F1-score** between **0.85-0.9**. Although this is a great F1-Score as well, the performance is not the same as compared to the other fields because the model captures some false positive values as well due to slight imbalance in data after preprocessing.

References:

1. https://scikit-learn.org/dev/modules/generated/sklearn.ensemble.RandomForestClassifier.html\
2. https://www.geeksforgeeks.org/random-forest-classifier-using-scikit-learn/
3. https://youtu.be/4Im0CT43QxY?si=ZrrmIRUY7N0pR0rl
4. https://xgboost.readthedocs.io/en/stable/python/python_api.html
5. https://xgboosting.com/xgboost-compare-n_jobs-vs-nthread-parameters/
6. https://youtu.be/gPciUPwWJQQ?si=KDK-sePjZIKMsUn2
7. https://youtu.be/06aqa-W5YU4?si=74ntOh5VYgEg5njy
8. https://scikit-learn.org/1.5/modules/generated/sklearn.ensemble.VotingClassifier.html
9. https://www.geeksforgeeks.org/ml-voting-classifier-using-sklearn/
10. https://www.kaggle.com/code/marcinrutecki/voting-classifier-for-better-results