Report: Multi-label Classification of ICD-10 Codes Using Embeddings

1. Introduction

This report outlines the approach taken to solve the multi-label classification problem for predicting ICD-10 codes using embeddings provided for medical charts. The dataset contains approximately 200,000 training samples and 99,500 test samples, with each embedding being a 1024-dimensional vector and there are 1400 labels. The evaluation metric for this competition is micro-F2 score.

2. Data Engineering

2.1 Data Preprocessing

- **Input Features**: 1024-dimensional embeddings provided for each sample.
- **Target Labels**: 1,400 unique ICD-10 codes were treated as binary labels in a multi-label classification setting.

Steps:

- Converted ICD-10 codes to a binary matrix where each row corresponds to a sample, and each column represents whether a specific ICD-10 code applies to that sample.
- Normalized the embeddings to ensure they are on a similar scale, aiding the model's convergence.
- Converted data to float tensors for neural network model.

2.2 Missing Values

- No missing values were observed in the embeddings or labels.

2.3 Weighted Sampling

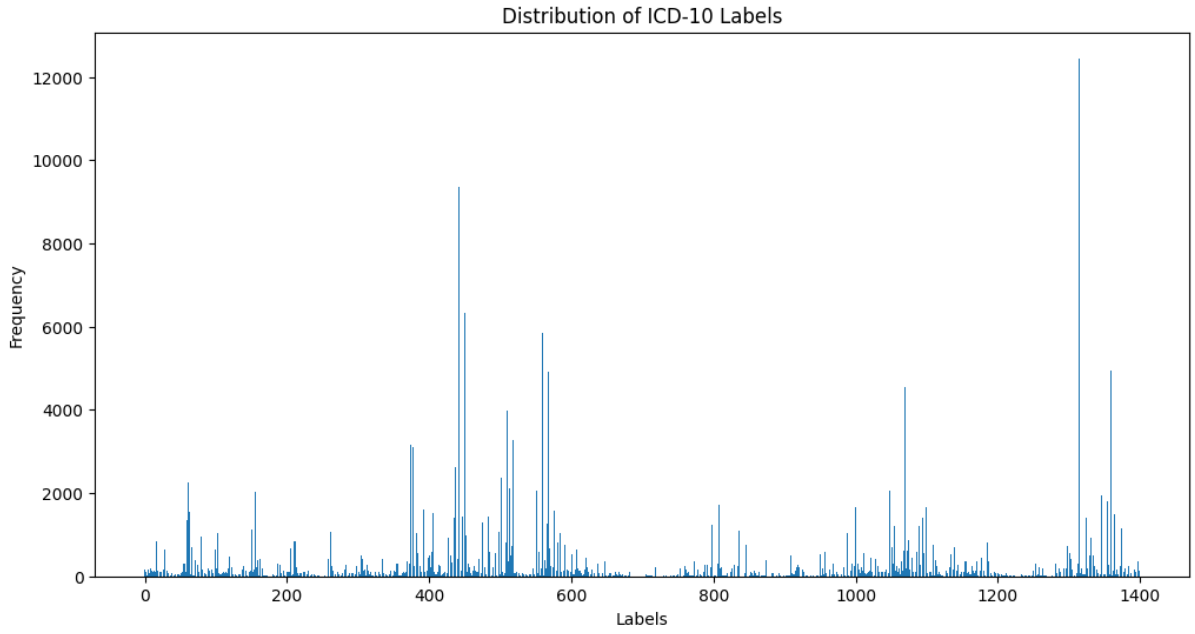
To address class imbalance:

- Calculated label frequencies and, computed per-class weights using inverse label frequencies, then derived per-sample weights for balanced training.
- Used WeightedRandomSampler for data loading.

3. Exploratory Data Analysis (EDA)

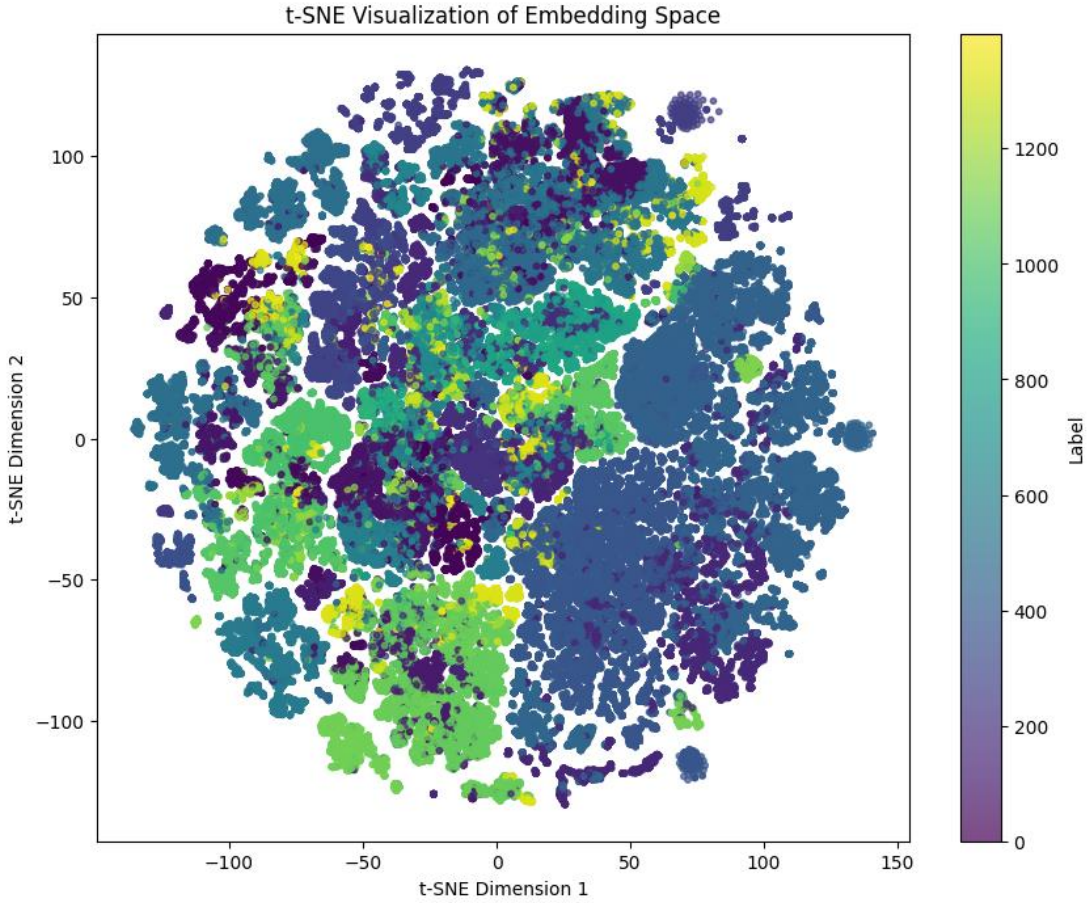
3.1 Label Distribution

- Most ICD-10 codes were highly imbalanced, with many codes appearing in <1% of samples.
- 1365 out of 1400 labels are occurring less than 1%, and 303 labels occurring less than 0.01% of samples.
- Visualize how the labels are distributed in the dataset.



3.2 **Visualization of Embeddings**

- Applied **PCA** and **t-SNE** to visualize the 1024-dimensional embeddings in a 2D space. These visualizations provided insights into clusters and potential separability of ICD-10 codes.



3.3 Co-occurrence of Labels

- Plotted a **heatmap** showing the co-occurrence of ICD-10 codes.
    - Certain codes often appeared together, suggesting underlying patterns that the model could leverage.

4. Model Selection

4.1 XGBoost Classifier

- Applied PCA (n_components = 100) for dimensionality reduction.
- Trained with MultiOutputClassifier with a XGBoost base estimator, but took 2:30 hrs to run the model even in GPU.
- Got training F1 score around 0.996 and validation score around 0.78, that is the model is too overfitting, then added some regularizations like max_depth = 5, n_estimators = 75, colsample_bytree=0.8, etc, but again overfitted. (Kaggle score 0.352)
- Hyperparameter tuning with GridSearchCV or RandomSearchCV was not a good option because the model already taking too much time.
- Then tried linear search with limited number of parameters for 100 labels and observed that most of the labels have validation F1 score more than 0.999 and train score 1.0, but the combined F1score on validation set was 0.79 and on train set around 0.98 (take approximately 2.5 hrs to run). So converted to neural Network.

4.2 Neural Network

MLPClassifier

- The **MLPClassifier** was chosen for its simplicity and compatibility with tabular data:
    - Hidden Layer 1: 1,024 units, BatchNorm, LeakyReLU (aplpha = 0.3), Dropout (0.1).
    - Output Layer: 1,400 units, Sigmoid activation for multi-label probabilities.
    - Input Layer: 1,024 units (embedding size), number of batches = 64.

Loss Function

- **Binary Cross-Entropy Loss (BCELoss)**: Optimized for multi-label classification tasks.

Optimizer

- **Adam Optimizer** with:
    - Learning rate: 5e-5.
    - Weight decay: 2.2e-7 (L2 regularization).

Learning Rate Scheduler

- **ReduceLROnPlateau** to reduce the learning rate on plateauing validation loss.

Hacks and Workarounds

- **Weighted Sampling**: Improved training stability on imbalanced labels.
- **Regularization Tuning**: Adjusted dropout rates, negative slope of the leakyRELU activation function and weight decay to prevent overfitting.
- Also tried by adding/removing hidden layers of 512 units and 256 units and changing the batch sizes. Optimized number of Epochs using validation score.
- **Dynamic Thresholding**: Explored dynamic thresholds for optimizing F1-score, but fixed threshold (0.5) yielded competitive results.
- Tried ensembling the different **MLPClassifiers** with different input sizes of data to prevent overfitting.

5) Training & Validation Performance

1) XGBoost

	Micro F1 score
Train	0.996
Validation	0.78

2) 1st MLP Classifier

Params: Hidden Layers
layer1 - 1024 units, Dropout – 0.1, ReLu,
layer2 - 512 units, Dropout – 0.1, ReLu,
layer3 - 256 units, Dropout – 0.1, ReLu,
epochs = 20, batch size – 64

	Micro F1 score	Accuracy	Precision	Recall
Train	0.9327	0.724	0.926	0.9211
Validation	0.79	0.52	0.79	0.792

3) Final MLP Classifier

Params: Hidden Layer – 1024 units, Dropout – 0.1, LeakyRelu – 0.3,

weight decay – 2.2*1e-7, epochs = 25, batch size - 64

	Micro F1 score	Accuracy	Precision	Recall
Train	0.945	0.845	0.96	0.95
Validation	0.8221	0.624	0.845	0.810

Perfomance on different Parameters

	A	B	C	D	E	F	G	H	I	J	K
1	Dropout % of l1	Dropout % of l2	Dropout % of l3	negaitve slope	negaitve slope	negaitve slope	lr	weight decay	epochs	Val F1	Train
2	0.15	0.12	0.1	0.2	0.2	0.2	0.0001	3*e-6	18	0.79566	0.9035
3	0.15	0.12	0.1	0.25	0.25	0.25	0.0001	0.5*	18	0.8113	0.93
4	0.15	0.12	0.1	0.4	0.4	0.4	0.0001		18	0.8133	0.92649
5	0.15	0.12	0.1	0.4	0.4	0.4	0.0001	0.05	18	0.813644	0.93214
6	0.15	0.12	0.1	0.6	0.6	0.6	0.0001	0.05	18	0.81301	0.91708
7	0.15	0.12	0.1	0.3	0.3	0.3	0.0001	0.05	18	0.8125635	0.93854
8	0.15	0.12	0.1	0.3	0.2	0.2	0.0001	0.05	18	0.808278	0.9382
9	0.15	0.12	0.1	0.2	0.3	0.3	0.0001	0.05	18	0.81309	0.94191
10	0.15	0.12	0.1	0.2	0.3	0.4	0.0001	0.05	18	0.8129563	0.94224
11	0.15	0.12	0.2	0.2	0.3	0.4	0.0001	0.05	18	0.812898	0.9318
12	0.15	0.2	0.1	0.2	0.3	0.4	0.0001	0.05	18	0.811687	0.93749
13	0.15	0.01	0.1	0.2	0.3	0.4	0.0001	0.05	18	0.812241	0.95
14	0.15	0.01	0.05	0.2	0.3	0.4	0.0001	0.05	18	0.814014	0.954
15	0.02	0.01	0.05	0.2	0.3	0.4	0.0001	0.05	18	0.81	0.96237
16	0.05	0.01	0.05	0.2	0.3	0.4	0.0001	0.5	18	0.8109	0.9563
17	0.08	0.01	0.05	0.2	0.3	0.4	0.0001	1	18	0.81023	0.94792
18	0.12	0.1	0.1	0.1	0.1	0.1	0.001	0	18	0.82272	0.911326
19	0.12	0.03	0.055	0.2	0.3	0.4	0.0001	0.12	20	0.815	0.958
20	0.12	0.05	0.055	0.2	0.25	0.25	0.001(sh)	0.02	20	0.80039	0.95928
21	0.15	0.07	0.07	0.2	0.3	0.35	0.001(sh)	0.05	20	0.811	0.988
22	0.15	0.07	0.07	0.2	0.3	0.35	0.001(sh)	0.5	20		
23	0.15	0.07	0.07	0.2	0.3	0.35	0.001(sh)	0.07	25	0.813	0.99
24	0.15	0.07	0.07	0.2	0.3	0.35	0.001(sh)	0.09	25	0.813	0.9855
25	0.15	0.07	0.07	0.2	0.3	0.35	0.001(sh)	0.09	20	0.813	0.9819
26	0.2	0.1	0.1	0.2	0.3	0.35	0.001(sh)	0.3	20	0.8122	0.9818
27	0.5	0.3	0.1	0.2	0.3	0.35	0.001(sh)	0.3		0.808	0.978
28				0.2	0.3	0.35	0.001(sh)	0.3	25	0.8084	0.9823
29	0.5	0.5	0.3	0.2	0.3	0.35	0.001(sh)	0.08	25	0.81	0.98815
30		0.5		0.2	0.3	0.35	0.001(sh)		20		
31		0.5		0.2	0.3	0.35	0.001(sh)		25	0.81269	0.99
32	0.12	0.03	0.055	0.2	0.3	0.4	0.0001(sh)	0.12	20	0.808	0.9543
33	0.12	0.03	0.055	0.2	0.3	0.4	0.0001	0.12	25	0.8137	0.9667
34		0.03	0.055		0.3	0.4					
35	0.12	0.03	0.055	0.2	0.3	0.4	0.0001	0.12	21	0.817	0.96788
36	0.15	0.08	0.055	0.3	0.4	0.4	0.0001	0.12	21	0.818	0.959
37	0.15	0.08	0.055	0.3	0.4	0.4	0.0001	0.15	25	0.819	0.9642
38	0.15	0.08	0.055	0.3	0.4	0.4	0.0001	0.2	30	0.8191	0.9677
39	0.15	0.08	0.055	0.3	0.4	0.4	0.0001	0.3	30	0.818	0.966
40	0.15	0.08	0.055	0.3	0.4	0.4	0.0001	0.25	30	0.82015	0.96762
41	0.15	0.1	0.055	0.3	0.4	0.4	0.0001	0.25	30	0.819338	0.9667
42	0.15	0.1	0.055	0.3	0.4	0.4	0.0005(sh)	0.25	30	0.8179	0.98448
43	0.15	0.1	0.055	0.3	0.4	0.4	0.0005(sh)	0.25	20	0.815	0.98
44	0.15	0.2	0.055	0.3	0.4	0.4	0.0005(sh)	0.25	30	0.817	0.9834
45	0.15	0.2	0.055	0.3	0.4	0.4	0.0005(sh)	0.25	22	0.8164	0.981
46	0.15	0.2	0.055	0.3	0.4	0.4	0.0001	0.25	30	0.8164	0.962
47	0.15	0.2	0.055	0.3	0.4	0.4	0.0005	0.25	30	0.809	0.945	
48	0.15	0.2	0.055	0.3	0.4	0.4	0.00005	0.25	30	0.821889	0.95405	
49	0.15	0.2	0.055	0.3	0.4	0.4	0.00001	0.25	35	0.808	0.87	
50	0.15	0.2	0.055	0.3	0.4	0.4	0.00002	0.1	40	0.8207	0.9368	
51	0.15	0.2	0.055	0.3	0.4	0.4	0.00005	0.1	40	0.821437	0.96539	
52	0.2	0.2	0.055	0.3	0.4	0.4	0.00005	0.2	40	0.8221228	0.9606	
53	0.3	0.2	0.055	0.3	0.4	0.4	0.00005	0.22	40	0.8218		
54	0.25	0.2	0.055	0.3	0.4	0.4	0.00005	0.22	40	0.819	0.9581	
55	0.1	0.2	0.055	0.3	0.4	0.4	0.00005	0.22	40	0.8221887	0.966586	
56	0.05	0.2		0.3			0.00005	0.22	40	0.82	0.969	
57	0.1	0.2		0.3			0.00005	0.22	25	0.818	0.926	1layer(512)
58	0.1	0.2		0.3			0.00005	0.22	25	0.8215	0.948	layer1024
59	0.1	0.2		0.3			0.00005	0.1	35	0.8219	0.962	layer1024
60	0.2	0.2		0.3			0.00005	0.2	35	0.82225	0.96	layer1024
61	0.2	0.2		0.3			0.00005	0.3	35	0.8214	0.957	layer1024
62	0.2	0.2		0.2			0.00005	0.3	35	0.82205	0.9599	layer1024
63	0.2	0.2		0.1			0.00005	0.3	35	0.8218	0.963	layer1024
64	0.2	0.2		0.2			0.00005	0.3	35	0.819	0.946	laye512
65												