Alvin Ho
Kenrick Lam
Steven Tran

CMPT 459 Project Milestone 2 Report

2.1 Splitting Dataset

We split the data into training and validation dataset using scikit-learn with a 80-20 split, and also set random_state, shuffle, and stratify values to ensure we got a deterministic result.

2.2 Build Models

For this section we considered accuracy to be our primary criterion, interpretability as our second criterion, and robustness as our third criterion. Therefore, we used the following 3 models: XGBoost, K-Nearest Neighbours, and Random Forests. We chose two ensemble methods: XGBoost and Random Forests, because they are robust to outliers and non-linear data, and they produce results with high classification accuracy. As for KNN, we chose it because we felt the results were very simple to interpret and it is robust regarding search space since classes do not have to be linearly separable like in SVM.

2.3 Evaluation

As we considered accuracy to be our primary criterion when choosing the models, we used prediction accuracy as the main metric to evaluate our models, especially due to the problem statement for this project being able to accurately predict outcomes of COVID patients. For XGBoost, K-Nearest Neighbours, and Random Forests, the accuracy of our model's prediction values compared to the actual test data's outcomes were 87.89%, 88.71%, and 89.19% respectively.

We also used scikit-learn's classification report to get each model's precision and recall. From the results listed below, it appears that for all 3 models the accuracy and recall for quite high were every class label outside of Recovered, which had lower precision values than the others. They also notably had extremely low recall, indicating high amounts of false negatives. Part of this may be due to the low number of Recovered entries in our dataset compared to the other labels, as can be seen in the support column.

XGBoost	Precision	Recall	F1-Score	Support
Recovered	0.7302	0.1029	0.1804	894
Hospitalized	0.7998	0.8969	0.8456	25067
Non-hospitalized	0.9912	0.9947	0.9929	29978
Deceased	0.8040	0.6949	0.7455	17536
Accuracy			0.8789	73475

K-Nearest Neighbours	Precision	Recall	F1-Score	Support
Recovered	0.5839	0.0973	0.1668	894
Hospitalized	0.8391	0.8811	0.8596	25067
Non-hospitalized	0.9790	0.9929	0.9859	29978
Deceased	0.7975	0.7551	0.7757	17536
Accuracy			0.8871	73475

Random Forests	Precision	Recall	F1-Score	Support
Recovered	0.5419	0.1230	0.2005	894

Hospitalized	0.8365	0.8915	0.8631	25067
Non-hospitalized	0.9880	0.9925	0.9902	29978
Deceased	0.8102	0.7598	0.7842	17536
Accuracy			0.8919	73475

2.4 Overfitting

For the classification models we trained, we did not observe any overfitting. To find overfitting, we would have to find where training accuracy becomes stronger while the validation accuracy gets weaker. However, within our models, we noticed that instead the accuracy tended to stagnate for both the training and validation data after further increase of the hyperparameters. We tested different values for each model's respective hyperparameter to find the most optimal values for them and take precautions towards overfitting.