2.1: Splitting Dataset

The dataset was split into an 80:20 ratio using sklearn's *train_test_split* with the random state set to 42.

2.2 Build Models

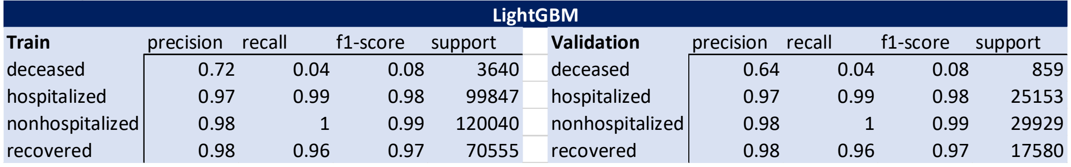
Some columns were dropped from the dataset. \`Combined_Key\`, \`latitude\`, \`longitude\`, \`Province_State\`, \`Country_Region\`, \`Lat\`, and \`Long_\` were removed because they held duplicated data that was already included in the columns \`province_filled\` and \`country_filled\`. Other columns such as \`new_date_confirmation\`, \`additional_information\`, \`source\`, and \`Last_Update\` were removed because they held metadata that was irrelevant to the classification task.

LightGBM was chosen to be the boosting model. Gradient boosting was chosen instead of adaptive boosting because AdaBoost is not optimized for speed or handling noisy data. Since a significant number of values were imputed (200,000+ imputations for missing \`age\` values alone), it is possible that the Covid-19 dataset is noisy and would not be handled well by AdaBoost. Gradient boosting is also more flexible than AdaBoost since more hyperparameters can be tuned to improve performance. Of the gradient boosters, LightGBM was chosen over XGBoost because it can handle categorical data (such as the \`country\` and \`state\` columns in the dataset) efficiently without converting them with one-hot encoding.

Random Forest was chosen to be the second model. By producing multiple different possible outputs from the multiple decision trees that were generated, Random Forests is a highly accurate classifier that can produce great results even without hyper-parameter tuning. In terms of accuracy, Random Forests outperforms other faster classifiers such as Decision Trees, Naive Bayes, Logistic regression, KNN, etc. However, more accurate classifiers such as MLP, Kernel SVM, and Random Forests all suffer from the problem of being incredibly slow and consuming a lot of processing power. In the end, the Random Forests classifier was chosen among these accurate classifiers due to its flexibility in handling numerical/categorical data and being robust when handling outliers, noise, and missing values. Considering that the dataset given in the project is composed of both numerical and categorical type data, classifiers such as Linear regression or Linear SVMs would have a difficult time working with this dataset. Additionally, the speed of Random Forests is not the worst as very accurate classifiers that can outperform Random Forests, such as MLP, are usually slower and consume more resources.

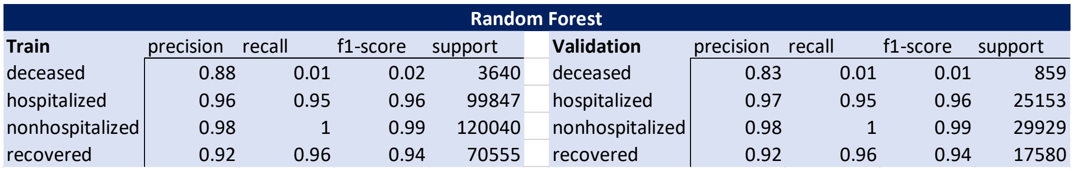
2.3 Evaluation

The F1 score was used to evaluate the models because it combines the precision and recall values to get a holistic look at the models' performance. See below for the full set of scores for the LightGBM and Random Forest models:



LightGBM

Train	precision	recall	f1-score	support		**Validation**	precision	recall	f1-score	support
deceased	0.72	0.04	0.08	3640		deceased	0.64	0.04	0.08	859
hospitalized	0.97	0.99	0.98	99847		hospitalized	0.97	0.99	0.98	25153
nonhospitalized	0.98	1	0.99	120040		nonhospitalized	0.98	1	0.99	29929
recovered	0.98	0.96	0.97	70555		recovered	0.98	0.96	0.97	17580



Random Forest

Train	precision	recall	f1-score	support		**Validation**	precision	recall	f1-score	support
deceased	0.88	0.01	0.02	3640		deceased	0.83	0.01	0.01	859
hospitalized	0.96	0.95	0.96	99847		hospitalized	0.97	0.95	0.96	25153
nonhospitalized	0.98	1	0.99	120040		nonhospitalized	0.98	1	0.99	29929
recovered	0.92	0.96	0.94	70555		recovered	0.92	0.96	0.94	17580

As seen in the f1-score column for both the train and validation sets, the models performed very well at correctly classifying cases that were `hospitalized`, `non-hospitalized`, and `recovered`. However, they performed poorly for classifying `deceased` cases. Confusion matrices were generated (see *confusion_matrix_train_gbd.png*, *confusion_matrix_val_gbd.png, confusion_matrix_train_rf.png,* and *confusion_matrix_val_rf.png* in the /plots directory) to see what was happening with the misclassifications. The matrices showed that the majority of `deceased` cases were being misclassified as `hospitalized`. The misclassification could be because the `deceased` and `hospitalized` outcomes are similar in that they are the two most severe of the possible outcomes. It could also be because there are fewer `deceased` cases so the models did not sufficiently weigh them during training.
An interesting observation was made after determining the feature important attributes for the Random Forests model (see *feature_importance_rf.png*). It was discovered that around 33% of the rows with `deceased` labels had the top 4 feature important attributes to be identical or have very similar values as the most frequent values for the respective columns in rows with `hospitalized` outcome labels. Considering the number of similar values in these significant columns between the `hospitalized` and `deceased` cases and the fewer number of `deceased` cases, it is possible that the classifier is misclassifying `deceased` because a given deceased row contains too many common values that would highly suggest a `hospitalized` outcome.

Stepping back and looking at the meaning of the data, it could be reasonably argued that the recall of the `deceased` label is an extremely important evaluation measure as it is the percentage of correctly classifying a severe case of Covid-19. If these models were used to predict a patient's response to Covid-19, they would do well to predict whether the patient would have to be hospitalized or not, but would do poorly at predicting their chances of survival.

2.4 Overfitting

The maximum depth for LightGBM was varied between 2 and 18 and the accuracy scores were plotted to see if the model was overfitting (see *overfitting_check_gdb.png*). The accuracy for the validation set did not decrease as the maximum depth increased, which would have indicated overfitting. Therefore, the LightGBM model did not overfit, but the accuracy did plateau for maximum depth >= 8. It is also interesting to note that the accuracy score for the validation set was consistently higher than that of the train set. This could be due to the split of the dataset, since the train set had a higher percentage of `deceased` outcomes than the validation set (by chance), and it is the `deceased` cases that are often misclassified.

Similarly, the maximum depth for Random Forests was set to be between 10 and 100 and the accuracy scores were plotted to see if the model was overfitting (see *overfitting_check_rf.png*). By nature, random forests are designed to be less prone to overfitting as they use random creations of forests to reduce overfitting within decision trees. Therefore, an interesting result was that the accuracy for the training set remained the same while the validation set fluctuated between the values of 0.959 and 0.960 but never went into a steady decrease which would indicate overfitting. This means that the Random Forest model is not overfitting because the accuracy score stabilizes near the first few trees as more trees are added to the model. Similar to LightGBM, the accuracy score for the validation set was always higher than the train set which could also be correlated to the chance that more `deceased` outcomes were in the training set than the validation set.