Logbook: A Machine Learning Approach for Predicting Length of Stay at the Emergency Department

August 19th - September 5th
Research regarding various medical diseases and disorders as well as statistics and machine learning methods were conducted. After narrowing down a medley of ideas, we settled on a project on using Random Forest to predict the characteristics of patients with Atrial Fibrillation. Initial project proposal on said project was written.

September 7th, 2023
Sent emails to school science teachers asking for assistance on reviewing the project proposal, and were recommended to email the University of Calgary for further support.

September 13th, 2023
Emailed University of Calgary for support regarding the initial project proposal.

September 14th, 2023
Received response from Ms. Catherine Eastwood of the University of Calgary, who looked through the initial proposal and expressed interest in working with us on our initial project.

September 26th, 2023
First meeting with University of Calgary professors, including Ms. Eastwood, to discuss the possibilities of the initial project. They warned that gathering the data will likely be the most difficult part, and that under the time constraints of the CYSF project, obtaining the desired data may not be possible.

October 11th, 2023
Second meeting with professors at the University of Calgary. We were informed that it is impossible to get data for our initial idea because of privacy concerns and the current time restraints. Professors suggested the idea of using a previously made dataset on the Emergency Department, which could potentially be used for data analysis. After considerations and discussions, we decided to analyze the ED data because it is also applicable to modern societal issues, and is simpler for us to learn machine learning coding with.

October 12th - 19th, 2023
Started a new project proposal on predicting the length of stay at ED using the machine learning algorithm of Random Forest.

October 20th, 2023
Third meeting, where we met up with a professor at the university, Dr. Jessalyn Holodinsky, and we had further discussion about our project proposal and machine learning methods, as well as where to begin this project from here.

October 30th, 2023
A NDA on the new ED dataset was signed and we successfully obtained the ED dataset from Alberta Health Services.

November 9th, 2023
Officially started learning RandomForest models; we went to Dr. Jessalyn's office to learn the codes.

November 10th - November 16th, 2023
First time coding a random forest model! We used 1% of the entire data (7000 data points) as a sample dataset as practice and to run on our own laptops.

```
# import libraries for reading, tidyr, random forest packages

library(readr)
library(tidyr)
library(randomForest)

# change name of variable
names(ED_OUT_cleaned_sample)[names(ED_OUT_cleaned_sample)=="_3yrhospitalaztion"] <- "prev_hosp"

# making factor variables (categorize them)
ED_OUT_cleaned_sample$age_group <- as.factor (ED_OUT_cleaned_sample$age_group)
ED_OUT_cleaned_sample$sex <- as.factor (ED_OUT_cleaned_sample$sex)
ED_OUT_cleaned_sample$inst_id <- as.factor (ED_OUT_cleaned_sample$inst_id)
ED_OUT_cleaned_sample$triage code <- as.factor (ED_OUT_cleaned_sample$triagecode)
ED_OUT_cleaned_sample$prev_hosp <- as.factor (ED_OUT_cleaned_sample$prev_hosp)
ED_OUT_cleaned_sample$icd10_cat <- as.factor (ED_OUT_cleaned_sample$icd10_cat)
```

```
# cut the predicting variable (long vs short) instead of continuous numerical values
ED_OUT_cleaned_sample$long_stay <- cut (ED_OUT_cleaned_sample$ed_departure_totriage,
                                    breaks = c(-Inf, 240, Inf),
                                    labels = c("short","long"))

# random Forest try it out! na.action is for omitting missing data
rf_2variables <- randomForest(long_stay ~ prev_hosp + icd10_cat,
                        data = ED_OUT_cleaned_sample,
                        importance = TRUE,
                        na.action = na.omit,
                        ntree = 500,
                        mtry = 2)


# to show results of random Forest model
print (rf_2variables)
```

Began to write weekly write-ups regarding what we have learned throughout the week, including definitions for machine learning-related terminology. The write up was as follows:

Out-of-bag error is a way to measure the prediction error of RandomForest models. It utilizes a method called bootstrap aggregating (bagging) where two independent subsets are created. The “in-the-bag” (bootstrap sample) set is used to train the model, while the “out-of-bag” (OOB) set is defined as all other data. While the OOB sets can be combined into one whole dataset, each actual sample can only be considered OOB for the specific tree that it is not included in. The samples in the OOB set are then aggregated to find the majority prediction results. Subsequently, the error rate of the OOB set is then compared with the error rate of the bootstrap sample, which provides the out-of-bag error score: the number of wrongly classified samples in the OOB set.

The *confusion matrix* is a measurement for the model’s overall performance. It is a 2 by 2 table that shows whether or not the predictions are accurate for the positive and negative results. For the purposes of this project, the results are whether the patient stayed in the ED for more or less than 6 hours. The first row includes the true positive (TP) and false positive (FP) results. The TP is the correct number of patients who stayed in the department for more than 6 hours, while the FP is the incorrect number predicted for those positive values. Underneath, in the second row, are the false negatives (FN) and true negatives (TN) values. FN is the number of correct predictions for people who stayed in the department for less than 6 hours, while TN is the number of incorrect predictions for that.

To calculate the total *accuracy* of a model, the formula is the number of correct predictions in total (TP + TN) divided by the number of total predictions (TP + FP + TN + FN). Thus, the confusion matrix is essential for assessing the performance of the models.

Over the course of this past week, we have constructed three models: a 3-variable model, a 4-variable model, and a 5-variable model. During last week's meeting, we learned that the ntree value is the number of trees used in the model, and the mtry value is the number of variables that is tried as candidates at each split. Without specifying the mtry value for a classification-type model, the value will default to sqrt(p), where p is the number of variables. Accuracy assessments were done for models with different mtry values, which increased in increments of 100 from mtry = 100 to mtry = 1000. The accuracy assessments themselves were done via the out-of-bag error and the confusion matrix results, which have been explained above.

In order to maximize the model's performance, the accuracy is assessed for different numbers of trees as well. For each mtry for the different models, the OOB and confusion matrix values are recorded in order to calculate the accuracy for them. Doing so will allow us to see what the best number of trees to have for each of the models.

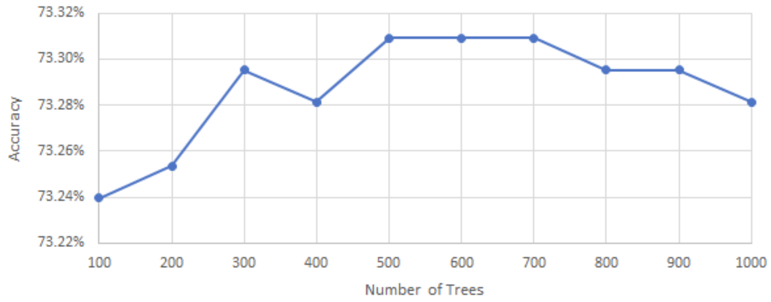
The 2-variables Model

The variables used for the 2-variables model were the previous hospitalization status and ICD score. Since there are only 2 variables, the mtry = 1. The number of trees ranges from 100 to 1000, each increasing by 100. For better understanding, the ntree range was narrowed down more precisely to see if the performance can go up even more. From the range of 500 to 700 trees, the model's accuracy was assessed at 20 trees increase intervals. The mean value and mean values are shown in the table below.

Table 1. The Mean and Maximized Performance for Out-Of-Bag Error and Total Accuracy of Different mtry Values for a 2-Variables RandomForest Model, Using ntree = 100 - 1000.

	mtry = 1
Mean OOB Error	26.71%
Mean Accuracy	73.29%
Lowest OOB Error	26.69%
Highest Accuracy	73.31%

Graph 1. Model Accuracy with 2 Variables: 100 to 1000 Trees



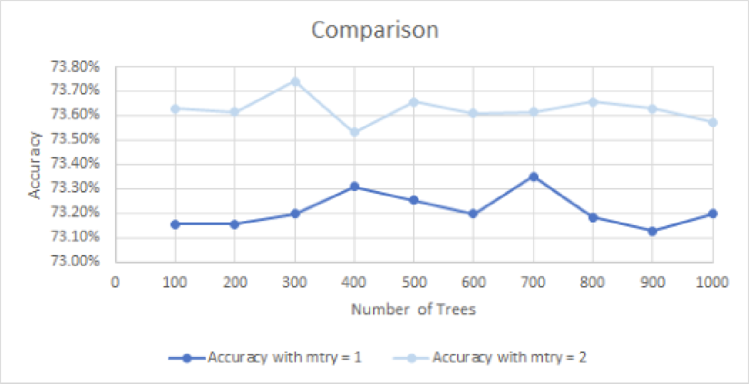
The 3-variables Model

The variables used for the 3-variables model were the previous hospitalization status, ICD score, and triage code. The number of trees ranges from 100 to 1000, each increasing by 100. The accuracy was assessed at mtry values of 1 and 2 to determine the best mtry value. The results of the tests are as follows:

Table 2. The Mean and Maximized Performance of Out-of-bag Error and Total Accuracy of Different mtry Values for a 3-variable RandomForest Model, Using ntree = 100 - 1000.

	mtry = 1	**mtry = 2**
Mean OOB Error	26.79%	26.61%
Mean Accuracy	73.21%	73.63%
Lowest OOB Error	26.65%	26.28%
Highest Accuracy	73.35%	73.74%

Graph 2. Comparison between 3-variables Model's Accuracy with mtry of 1 and 2: 100 to 1000 Trees.



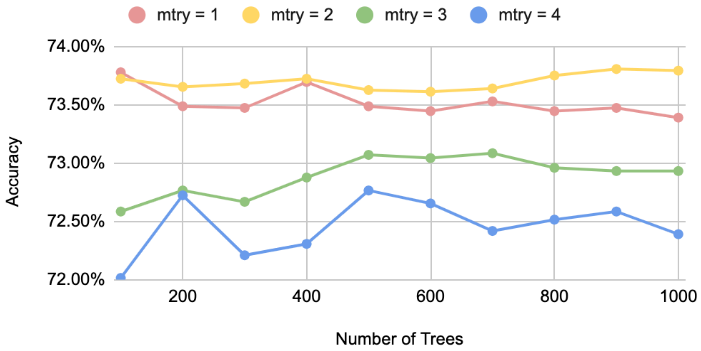
The 4-variables model

The variables used for the 4-variable model were the previous hospitalization status, the ICD score, the CTAS code, and the hospital ID. In this model, we tested mtry = 1, 2, 3, 4, with each set of mtry models done with ntree = 100-1000 (increasing by 100). The results of the tests are compiled below:

Table 3. The Mean and Maximized Performance of Out-of-bag Error and Total Accuracy of Different mtry Values for a 4-variable RandomForest Model, Using ntree = 100 - 1000.

	mtry = 1	**mtry = 2**	**mtry = 3**	**mtry = 4**
Mean OOB Error	26.48%	26.30%	27.11%	27.54%
Mean Accuracy	73.52%	73.70%	72.89%	72.46%
Lowest OOB Error	26.22%	26.19%	26.91%	27.23%
Highest Accuracy	73.78%	73.81%	73.09%	72.77%

Graph 3. Comparison of Accuracy of Different mtry Values for a 4-Variable Model.



As shown in Table 3, the mtry value with the best mean and maximized results is mtry = 2. Graph 3 also displays this, showing how for every ntree value (with the exception of ntree = 1), the models under mtry = 2 have the highest total accuracy. Referring back to the default mtry value (2 = sqrt(4)) the results of these tests have provided evidence to show how mtry = sqrt(p), where p is the number of variables, may be the best choice for deciding the mtry value for a model.

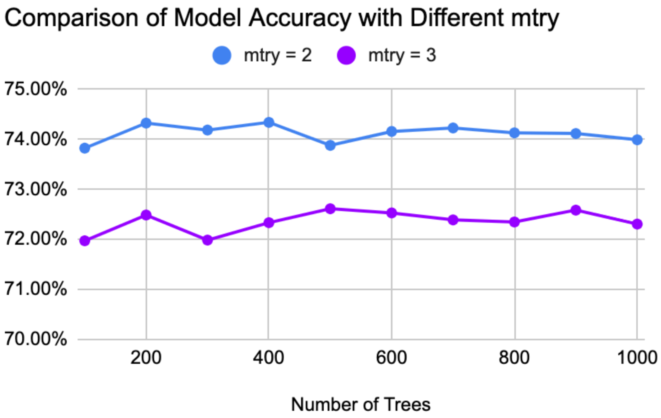
The 5-variables model

The variables used for the 5-variable model were the previous hospitalization status, the ICD score, the CTAS code, the hospital ID, and the age groups. As the fifth variable for this model was not specified to us, we chose to do the age groups as it has clinically proven connections to the health of the human body. Each set of mtry models were done with ntree = 100-1000 (increasing by 100). As concluded above, sqrt(p) may be the best mtry value for a model. Thus, for a model with p = 5 (and thus sqrt(p) = 2.24), we tested mtry = 2, 3 specifically to determine the boundaries of the default mtry formula. The results of the tests are compiled below:

Table 4. The Mean and Maximized Performance of Out-of-bag Error and Total Accuracy of Different mtry Values for a 5-variable RandomForest Model, Using ntree = 100 - 1000.

	mtry = 2	**mtry = 3**
Mean OOB Error	25.88%	27.64%
Mean Accuracy	74.12%	72.36%
Lowest OOB Error	25.66%	27.39%
Highest Accuracy	74.34%	72.61%

Graph 4. Comparison of Accuracy of Different mtry Values for a 5-Variable Model.



As shown in Table 4, the results of mtry = 2 are much more efficient than those of mtry = 3. Furthermore, Graph 4 displays how every single one of the mtry = 2 models provided better results than any of the mtry = 3 models. Thus, for the next tests and models created, we will be determining mtry via the default formula of mtry = sqrt(p), where p is the number of variables.

November 17th, 2023

Another meeting with Jessalyn. We spoke about the progress we made in the past week, such as the change in accuracy when the ntree or mtry value changes.

November 18th - 30th, 2023

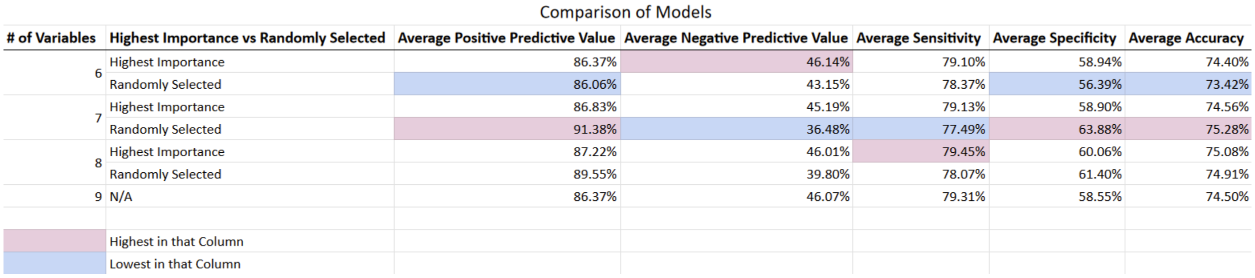
We added more variables to the model and found that the accuracy increases. Continued to write our weekly write-ups on our discoveries and progress:

Positive predictive value refers to the total number of true positives in all of the positive predictions, which is calculated by TP / (TP + FP).
Negative predictive value is the total number of true negatives in all of the negative predictions, calculated using the formula TN / (TN + FN).
Sensitivity is calculated using the formula TP / (TP + FN). A highly sensitive test is an indicator that there are few false negative results, thus fewer cases of long stay are missed. It also means that it has less false negatives.
Specificity uses the formula TN / (TN + FP). A high specificity indicates that the model is effective at correctly identifying instances of the negative class. It also means that the number of false positives is lower.

Model with High Accuracy

Through experimenting with the type and number of variables, as well as the ntree value, we concluded that the model with 7 randomly chosen variables (institutional peer group, the patient district code, main ICD code, sex, previous hospital status, and CTAS score) has the highest accuracy.

Model Comparison



Comparison of Models

# of Variables	Highest Importance vs Randomly Selected	Average Positive Predictive Value	Average Negative Predictive Value	Average Sensitivity	Average Specificity	Average Accuracy
6	Highest Importance	86.37%	46.14%	79.10%	58.94%	74.40%
	Randomly Selected	86.06%	43.15%	78.37%	56.39%	73.42%
7	Highest Importance	86.83%	45.19%	79.13%	58.90%	74.56%
	Randomly Selected	91.38%	36.48%	77.49%	63.88%	75.28%
8	Highest Importance	87.22%	46.01%	79.45%	60.06%	75.08%
	Randomly Selected	89.55%	39.80%	78.07%	61.40%	74.91%
9	N/A	86.37%	46.07%	79.31%	58.55%	74.50%
🟪	Highest in that Column					
🟦	Lowest in that Column					

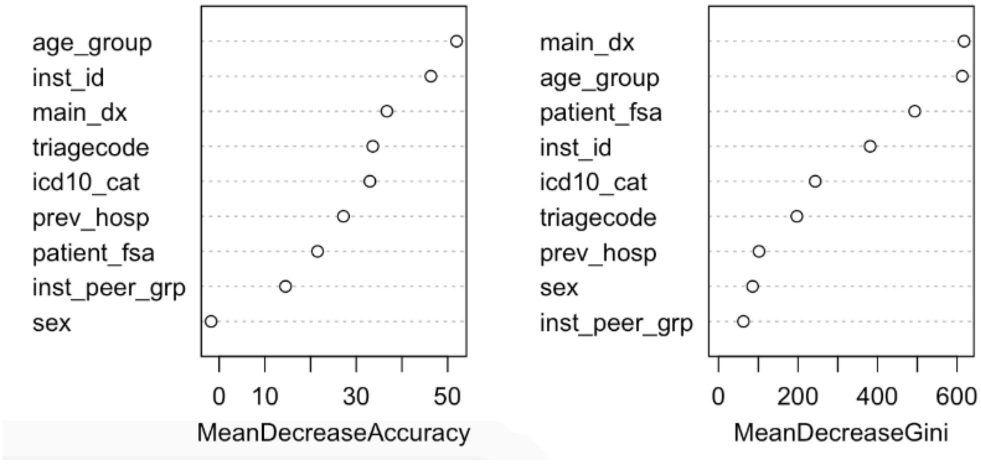
As shown in the data table above, the model with 7 randomly selected variables does have the highest accuracy (**75.28%**). It is likely because of its high PPV and Specificity. This represents that the model has a high ability to determine true positive cases and a lower false positives rate. However, looking at the sensitivity, this model actually has the lowest sensitivity compared to all the other ones. This is not ideal because a lower sensitivity would mean that the model is more likely to miss positive cases. When more cases of long stays are missed, it may lead to insufficient resources and time planned for that patient, potentially causing inconveniences. Considering the sensitivity, the model with the highest percentage (**79.45%**) is the one with 8 most important variables. This would mean that using this model, less cases of positives would be missed, which may be more useful for providing sufficient resources.

Hypothesis: If the number of variables in a RandomForest model increases, the accuracy will increase too.

As shown by previous tests (in the write-up of November 9th - 17th), the best mtry value is sqrt(p), where p is the number of variables. Thus that is the baseline condition we had for these tests. Moreover, our previous tests indicate that the best ntree value for our purposes is from 300 to 500 trees; however as the results were overall similar, we decided to include tests of ntree = 700 and ntree = 1000 as well. Previously, we had also done a model with 2, 3, 4, and 5 variables, so this time we chose to test models of 6 variables to 9 (the maximum number of) variables.

To select the variables, we decided to do a variable importance graph for the model with all the possible variables. The variable importance graph is shown below:

Graph 1: The variable importance plot for a 9-variable RandomForest model.



Thus for the first set of 6-variable models, we chose to use the top six variables, and for the 7-variable model, the top seven variables, and so on.

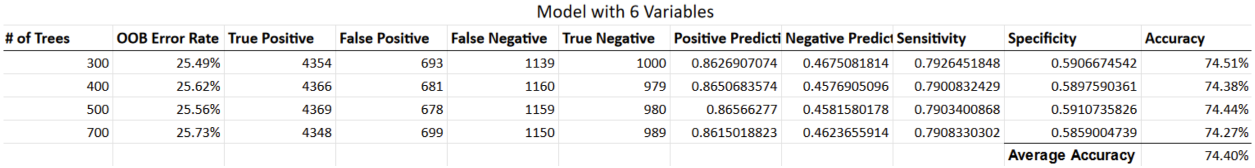
However, as we were not completely certain that these variables are optimal, we also chose to make 6 - 9 variable models that utilized randomly chosen variables as well.

The 6-Variable Models

Highest Importance Variables Models

For this model, the variables are age group, triage code, previous hospital admission, main ICD code, the ICD chapter, and hospital ID.

Table 1. The accuracy assessment of a 6-variable RandomForest model with highest importance variables.



Model with 6 Variables

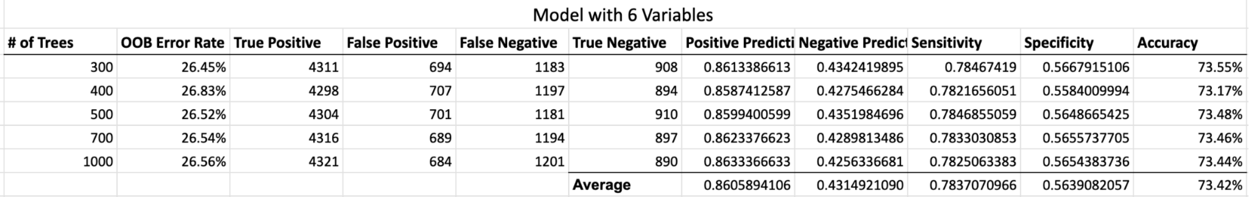
# of Trees	OOB Error Rate	True Positive	False Positive	False Negative	True Negative	Positive Predicti	Negative Predict	Sensitivity	Specificity	Accuracy
300	25.49%	4354	693	1139	1000	0.8626907074	0.4675081814	0.7926451848	0.5906674542	74.51%
400	25.62%	4366	681	1160	979	0.8650683574	0.4576905096	0.7900832429	0.5897590361	74.38%
500	25.56%	4369	678	1159	980	0.86566277	0.4581580178	0.7903400868	0.5910735826	74.44%
700	25.73%	4348	699	1150	989	0.8615018823	0.4623655914	0.7908330302	0.5859004739	74.27%
									Average Accuracy	74.40%

As shown in the table above, the average accuracy yielded by this 6-variable model was **74.40%**, with the highest accuracy going to the model of ntree = 300 with **74.51%**.

Random Variables Models

This model had randomly chosen variables, which were age group, patient district code, sex, main ICD code, the ICD chapter, and hospital ID.

Table 2. The accuracy assessment of a 6-variable RandomForest model with randomly chosen variables.



Model with 6 Variables

# of Trees	OOB Error Rate	True Positive	False Positive	False Negative	True Negative	Positive Predicti	Negative Predict	Sensitivity	Specificity	Accuracy
300	26.45%	4311	694	1183	908	0.8613386613	0.4342419895	0.78467419	0.5667915106	73.55%
400	26.83%	4298	707	1197	894	0.8587412587	0.4275466284	0.7821656051	0.5584009994	73.17%
500	26.52%	4304	701	1181	910	0.8599400599	0.4351984696	0.7846855059	0.5648665425	73.48%
700	26.54%	4316	689	1194	897	0.8623376623	0.4289813486	0.7833030853	0.5655737705	73.46%
1000	26.56%	4321	684	1201	890	0.8633366633	0.4256336681	0.7825063383	0.5654383736	73.44%
					Average	0.8605894106	0.4314921090	0.7837070966	0.5639082057	73.42%

As seen in this table above, the average accuracy wound out to be **73.42%**, with the highest accuracy being ntree = 300 with **73.55%**.

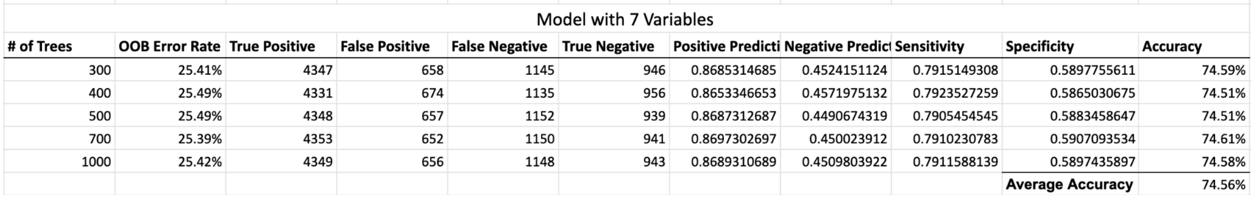
When comparing the average accuracy of the two 6-variable models, it can be seen that there is a **0.98% difference** between the two of them, with the **highest variable importance model having a higher accuracy**. Even when comparing the models with highest accuracy in each category, the former model had **0.96%** higher accuracy than the latter.

The 7-Variable Models

Highest Importance Variables Models

Due to the procedure shown above, the variables we had for this model were age group, the hospital ID, the main ICD code, CTAS score, the ICD chapter, previous hospital admission status, and the patient's district code.

Table 3. The accuracy assessment of a 7-variable RandomForest model with highest importance variables.



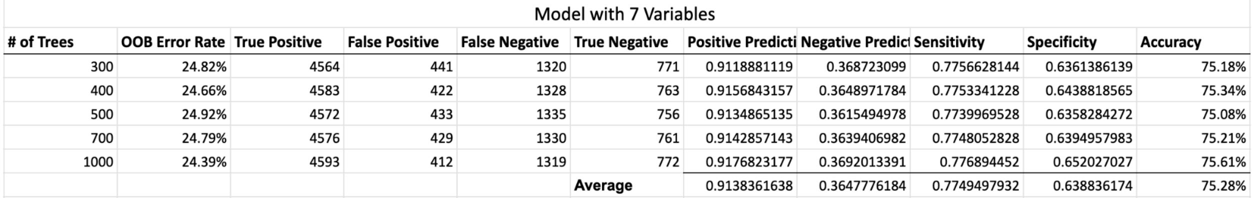
Model with 7 Variables										
# of Trees	OOB Error Rate	True Positive	False Positive	False Negative	True Negative	Positive Predicti	Negative Predict	Sensitivity	Specificity	Accuracy
300	25.41%	4347	658	1145	946	0.8685314685	0.4524151124	0.7915149308	0.5897755611	74.59%
400	25.49%	4331	674	1135	956	0.8653346653	0.4571975132	0.7923527259	0.5865030675	74.51%
500	25.49%	4348	657	1152	939	0.8687312687	0.4490674319	0.7905454545	0.5883458647	74.51%
700	25.39%	4353	652	1150	941	0.8697302697	0.450023912	0.7910230783	0.5907093534	74.61%
1000	25.42%	4349	656	1148	943	0.8689310689	0.4509803922	0.7911588139	0.5897435897	74.58%
									Average Accuracy	74.56%

In this table, it is displayed that the average accuracy for a 7-variable model across ntree = 300, 400, 500, 700, and 1000, is **74.56%**. Likewise, the average OOB rate is 25.44%. Out of these tests, the one that had the highest efficiency was using ntree = 700, at **74.61%**.

Random Variables Models

The variables included for these models were institutional peer group, the patient district code, main ICD code, sex, previous hospital status, and CTAS score.

Table 4. The accuracy assessment of a 7-variable RandomForest model with random variables.



Model with 7 Variables										
# of Trees	OOB Error Rate	True Positive	False Positive	False Negative	True Negative	Positive Predicti	Negative Predict	Sensitivity	Specificity	Accuracy
300	24.82%	4564	441	1320	771	0.9118881119	0.368723099	0.7756628144	0.6361386139	75.18%
400	24.66%	4583	422	1328	763	0.9156843157	0.3648971784	0.7753341228	0.6438818565	75.34%
500	24.92%	4572	433	1335	756	0.9134865135	0.3615494978	0.7739969528	0.6358284272	75.08%
700	24.79%	4576	429	1330	761	0.9142857143	0.3639406982	0.7748052828	0.6394957983	75.21%
1000	24.39%	4593	412	1319	772	0.9176823177	0.3692013391	0.776894452	0.652027027	75.61%
					Average	0.9138361638	0.3647776184	0.7749497932	0.638836174	75.28%

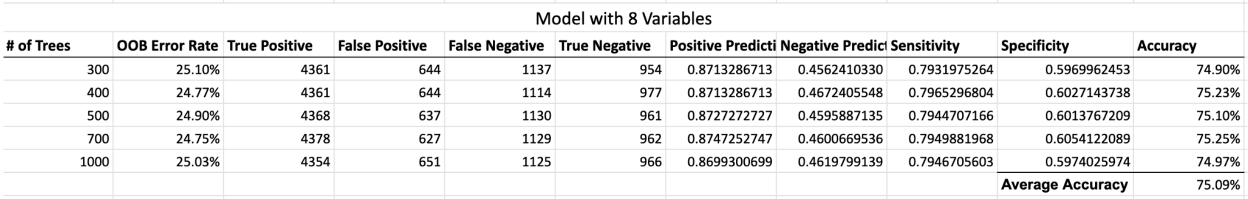
It can be seen in Table 4 that the average accuracy of these models was **75.28%**, with the highest accuracy model being the one using ntree = 1000 at **75.61%** accuracy.

When comparing the two 7-variable models, the random variables model had a **0.73%** higher average accuracy than the highest importance variables model. Similarly, the model with the highest accuracy out of the random variable models had a **1.00%** higher accuracy than the highest accuracy of the random variable models.

<ins>The 8-Variable Models</ins>

Highest Important Variables Models

Table 5. The accuracy assessment for a 8-variable RandomForest model with highest importance variables.



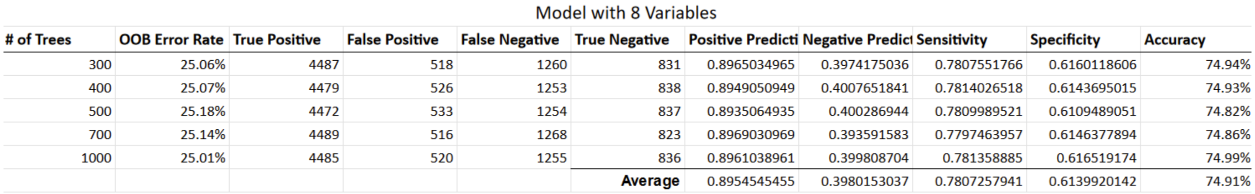
Model with 8 Variables

# of Trees	OOB Error Rate	True Positive	False Positive	False Negative	True Negative	Positive Predicti	Negative Predic	Sensitivity	Specificity	Accuracy
300	25.10%	4361	644	1137	954	0.8713286713	0.4562410330	0.7931975264	0.5969962453	74.90%
400	24.77%	4361	644	1114	977	0.8713286713	0.4672405548	0.7965296804	0.6027143738	75.23%
500	24.90%	4368	637	1130	961	0.8727272727	0.4595887135	0.7944707166	0.6013767209	75.10%
700	24.75%	4378	627	1129	962	0.8747252747	0.4600669536	0.7949881968	0.6054122089	75.25%
1000	25.03%	4354	651	1125	966	0.8699300699	0.4619799139	0.7946705603	0.5974025974	74.97%
									Average Accuracy	75.09%

Of the models made in this set of tests, the one with the highest accuracy was shown also at ntree = 700 (with a value of **75.25%**), with 75.23% of ntree = 400 following closely behind it. The average accuracy here, for the 8-variable model, was at **75.09%**. Likewise, the OOB error rate was averaged at 24.91%.

Random Variables Models

Table 6. The accuracy assessment for a 8-variable RandomForest model with randomly selected variables.



Model with 8 Variables

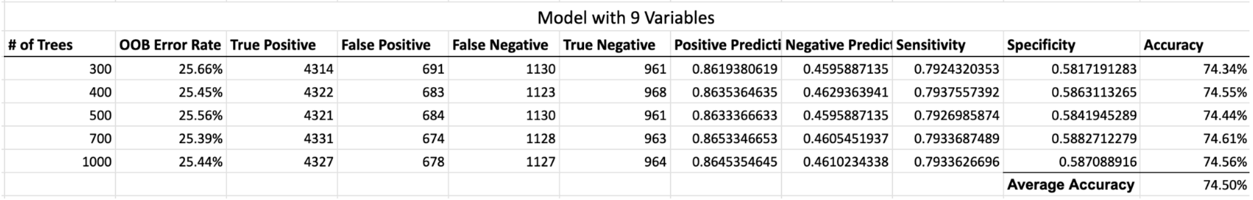
# of Trees	OOB Error Rate	True Positive	False Positive	False Negative	True Negative	Positive Predicti	Negative Predic	Sensitivity	Specificity	Accuracy
300	25.06%	4487	518	1260	831	0.8965034965	0.3974175036	0.7807551766	0.6160118606	74.94%
400	25.07%	4479	526	1253	838	0.8949050949	0.4007651841	0.7814026518	0.6143695015	74.93%
500	25.18%	4472	533	1254	837	0.8935064935	0.400286944	0.7809989521	0.6109489051	74.82%
700	25.14%	4489	516	1268	823	0.8969030969	0.393591583	0.7797463957	0.6146377894	74.86%
1000	25.01%	4485	520	1255	836	0.8961038961	0.399808704	0.781358885	0.616519174	74.99%
					Average	0.8954545455	0.3980153037	0.7807257941	0.6139920142	74.91%

The average accuracy for this model is **74.91%**, which is lower compared to the model with 7 randomly selected variables. The highest accuracy for this model is at ntree = 1000, with **74.99%**.

Comparing the two 8 variable models, it can be seen that the first model with the first 8 most important variables has **0.17%** higher accuracy.

The 9-Variable Model

Table 7: The accuracy assessment for a 9-variable model.



Model with 9 Variables

# of Trees	OOB Error Rate	True Positive	False Positive	False Negative	True Negative	Positive Predicti	Negative Predic	Sensitivity	Specificity	Accuracy
300	25.66%	4314	691	1130	961	0.8619380619	0.4595887135	0.7924320353	0.5817191283	74.34%
400	25.45%	4322	683	1123	968	0.8635364635	0.4629363941	0.7937557392	0.5863113265	74.55%
500	25.56%	4321	684	1130	961	0.8633366633	0.4595887135	0.7926985874	0.5841945289	74.44%
700	25.39%	4331	674	1128	963	0.8653346653	0.4605451937	0.7933687489	0.5882712279	74.61%
1000	25.44%	4327	678	1127	964	0.8645354645	0.4610234338	0.7933626696	0.587088916	74.56%
									Average Accuracy	74.50%

The average accuracy for the 9-variable models was at **74.50%.** This proves against the hypothesis presented as it means that the accuracy is lower than that of a n 8-variable model. Out of the models, the one with the highest accuracy was also at ntree = 700: **74.61%.**

Dec. 1st, 2023
Another meeting. We spoke about what partial dependence and variable importance graphs are.

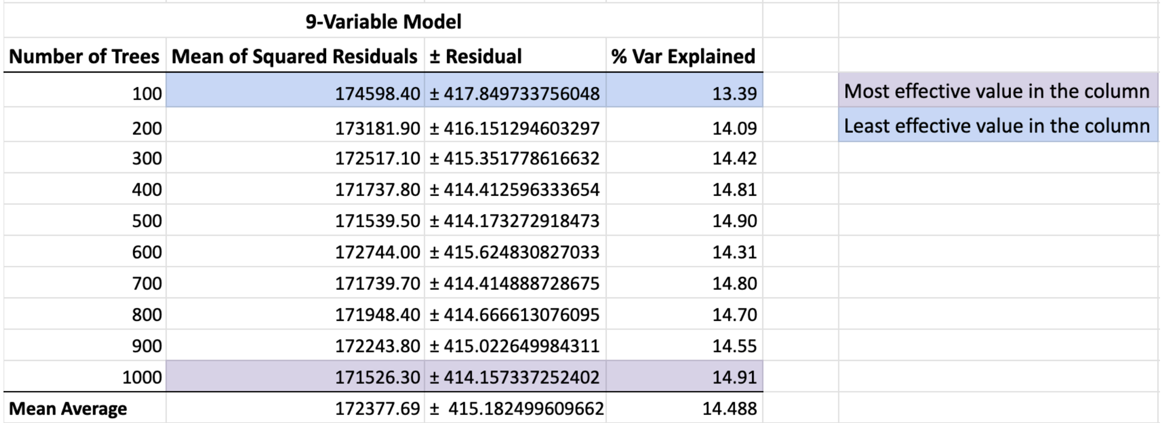
Dec 2nd - 7th, 2023
We started coding for some variable importance and partial dependence plots, and also started to learn a bit about random forest regression models. Our code was as below:

```
# variable importance
varImpPlot (rf_cat)

# partial dependence
partial(rf_regress, pred.var = "icd10_cat", data = test, plot = TRUE)
dev.off()
```

And our weekly write-up is as follows:

<u>9-Variable Model</u>

Table 1. The accuracy assessment of the 9-variable regression RandomForest model.

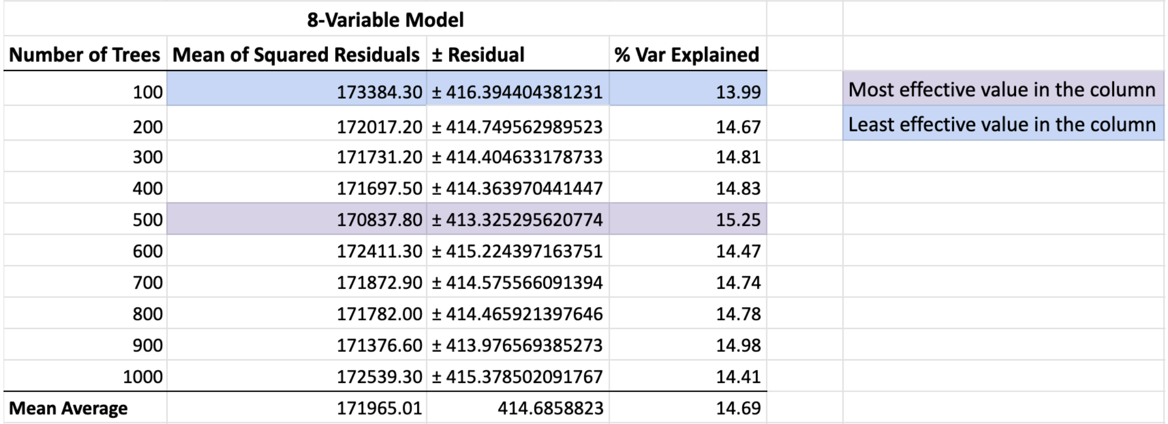


9-Variable Model					
Number of Trees	Mean of Squared Residuals	± Residual	% Var Explained		
100	174598.40	± 417.849733756048	13.39		Most effective value in the column
200	173181.90	± 416.151294603297	14.09		Least effective value in the column
300	172517.10	± 415.351778616632	14.42		
400	171737.80	± 414.412596333654	14.81		
500	171539.50	± 414.173272918473	14.90		
600	172744.00	± 415.624830827033	14.31		
700	171739.70	± 414.414888728675	14.80		
800	171948.40	± 414.666613076095	14.70		
900	172243.80	± 415.022649984311	14.55		
1000	171526.30	± 414.157337252402	14.91		
Mean Average	172377.69	± 415.182499609662	14.488		

<u>8-Variable Model</u>

Because the variables main_dx and icd10_cat are similar (icd10_cat is just main_dx categorized), we created an 8-variable model where icd10_cat was not included to test if it made a difference.

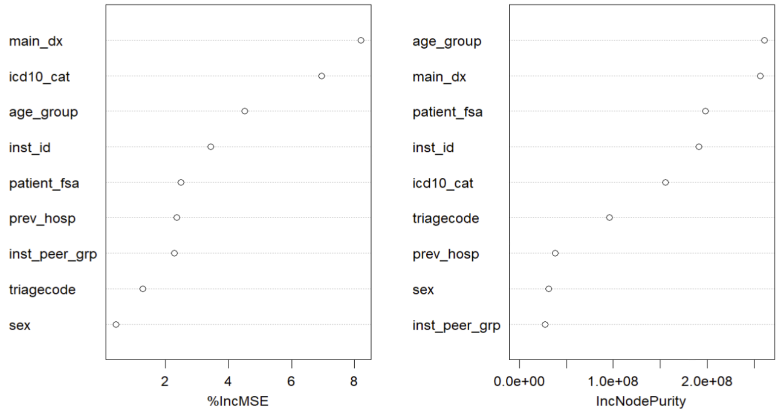
Table 2. The accuracy assessment of the 8-variable regression RandomForest model.



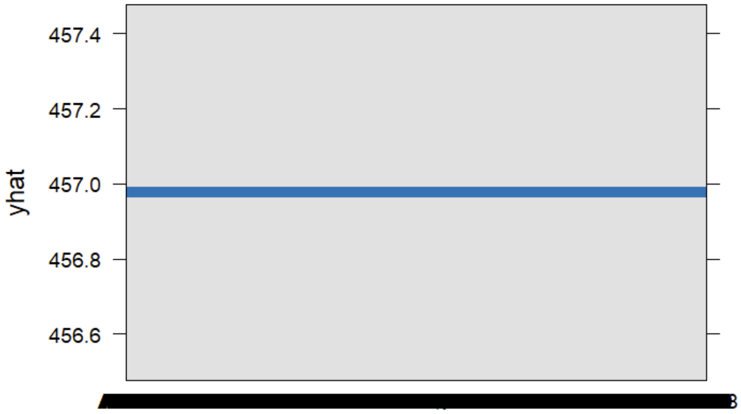
8-Variable Model					
Number of Trees	Mean of Squared Residuals	± Residual	% Var Explained		
100	173384.30	± 416.394404381231	13.99		Most effective value in the column
200	172017.20	± 414.749562989523	14.67		Least effective value in the column
300	171731.20	± 414.404633178733	14.81		
400	171697.50	± 414.363970441447	14.83		
500	170837.80	± 413.325295620774	15.25		
600	172411.30	± 415.224397163751	14.47		
700	171872.90	± 414.575566091394	14.74		
800	171782.00	± 414.465921397646	14.78		
900	171376.60	± 413.976569385273	14.98		
1000	172539.30	± 415.378502091767	14.41		
Mean Average	171965.01	414.6858823	14.69		

9-Variable Model

Graph 1. The variable importance plot for a 9-variable regression RandomForest model.

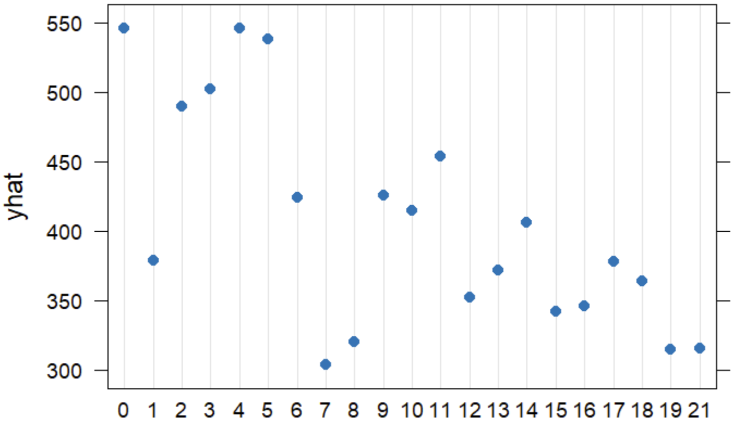


Graph 2. The partial dependence plot for the variable main_dx in relation to the model



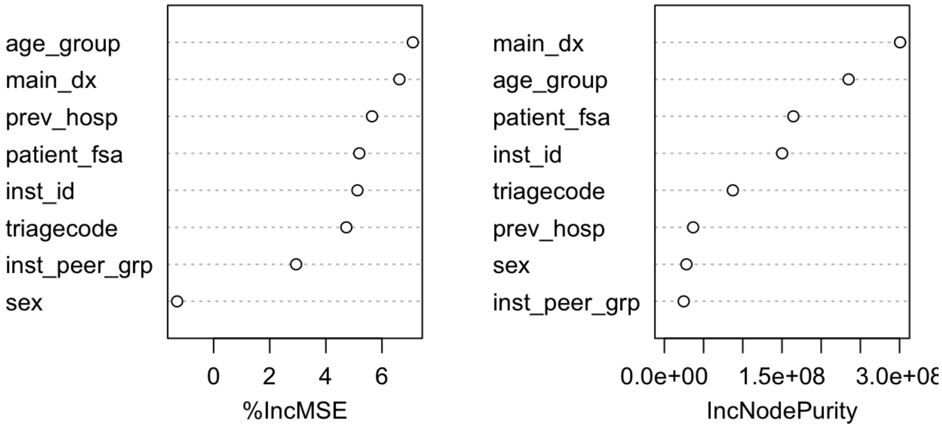
* There were way too many different patient fsa, so they are all condensed on the x-axis, and cannot be seen clearly.

Graph 3. The partial dependence plot for the variable icd10_cat in relation to the model.

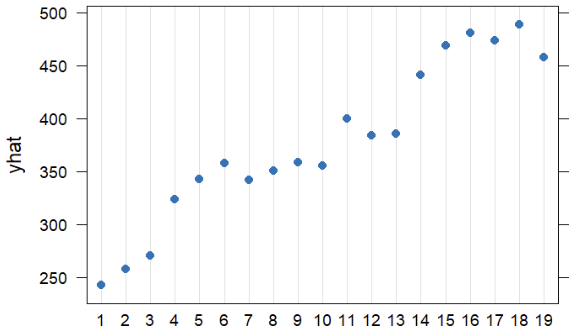


8-Variable Model

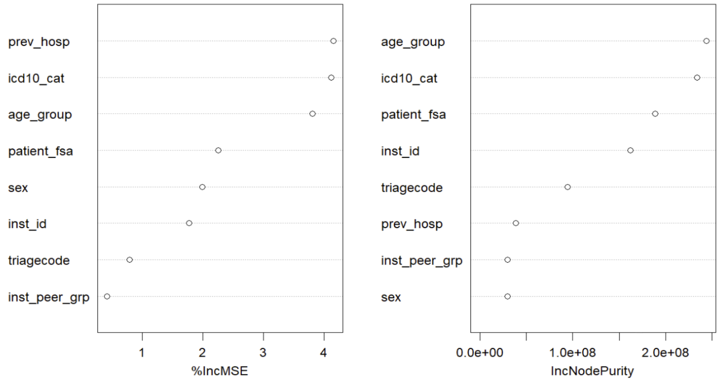
Graph 4. The variable importance plot for an 8-variable regression RandomForest model, using main_dx instead of icd10_cat.



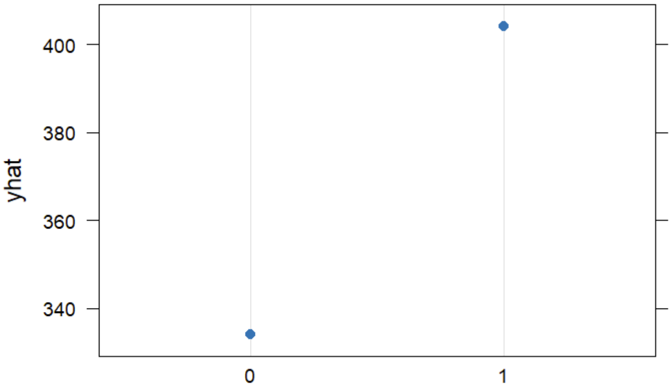
Graph 5. The partial dependence plot for the variable age_group in relation to the model.



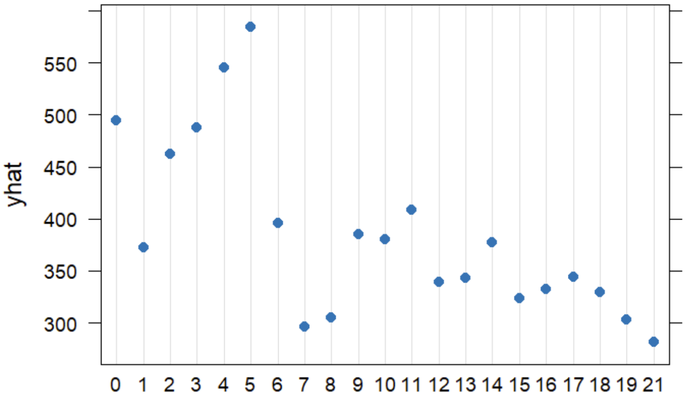
Graph 6. The variable importance plot for an 8-variable regression RandomForest model, using icd10_cat instead of main_dx.



Graph 7. The partial dependence plot for the variable prev_hosp in relation to the model.



Graph 8. The partial dependence plot for the variable icd10_cat in relation to the model.

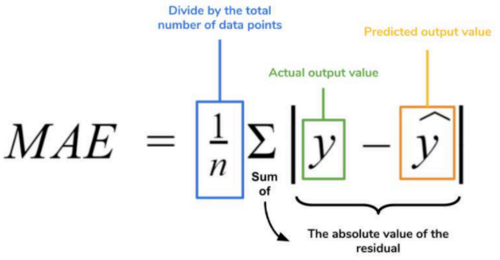


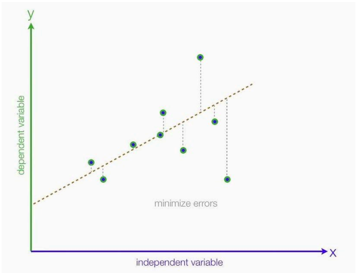
A linear regression is often used to find the relationship between variables, where a “line of best fit” is made to allow for better prediction. The predictions made by a linear progression model may not be highly accurate (compared to some nonlinear models), but they are consistently good and the line of best fit works for future predictions as well.

The *residual* is the difference between the predicted value (on the line of best fit) and the actual value of the data, where, for each datapoint, there is a different residual.

One way of assessing the accuracy of the model is to calculate the *root mean square error*, which is the average distance between the predicted values and the actual values of a model. Because some of the residuals may sometimes be negative (and thus will cancel positive values out), it is squared to make the values positive, and averaged to create the mean of squared residuals. Root mean square error is when the mean of squared residuals is then square rooted.

Mean absolute error refers to the mean difference in the predicted time and the actual time. On a linear regression, it would be the mean distance between each data point and the line of best fit. Usually, the closer the MAE is to zero, the better the model performs.


Divide by the total number of data points
Predicted output value
Actual output value
$MAE = \frac{1}{n}\sum \left| y - \hat{y} \right|$
Sum of
The absolute value of the residual



Bias, in a linear regression model, is a systematic error that occurs in the machine learning model itself. It is the inability for a model to capture the true relationship that exists in the data set. There is a relative high amount of bias in linear progression models because a linear line often cannot fit many data points.

% Variability Explained describes how much variance of the output variable can be explained by the input variable. The higher the % variable explained, the more the model is able to describe the variation in the data, and vice versa.

Dec. 8th, 2023

Another meeting, we discussed the variable importance and partial dependence plots created, they may potentially imply some social trends. We are also planning to build classification models that have more than 2 categories.

Dec 9th - Dec 14th

Accuracy assessment for different time breaks for binary classification and changing the number of categories (it was a little disappointing to see that increasing the number of categories made the accuracy decrease).

2 categories and different time breaks: The general trend is that as the time break gets larger, the accuracy increases.

Time Break (hour)	Accuracy
1	95.77%
2	82.67%
3	72.18%
4	68.39%
5	69.35%
6	74.41%
7	78.98%
8	82.10%
9	85.37%
10	87.69%
15	93.93%
20	96.11%

More than 2 categories: As the number of categories increases, the accuracy decreases.

# of Categories	Accuracy
3	54.61%
4	39.54%
6	32.90%
8	24.67%
10	19.56%

Dec. 15th, 2023

Another meeting, talked about the accuracy assessments we did, and also learned how to code for regression!

```
# regression model code
rf_regress <- randomForest(visit_los_minutes ~ age_group + inst_id + inst_peer_grp + sex + patient_fsa +
triagecode + icd10_cat + prev_hosp,
            data = train,
            importance = TRUE,
            na.action = na.omit)
```

```
print(rf_regress)

# testing
test$predict <- predict(rf_regress, newdata = test)
ED_sample_no_mv$predict <- predict(rf_regress, newdata = ED_sample_no_mv)
test$residual <- test$predict - test$visit_los_minutes
test$abs_residual <- abs(test$residual)
summary(test$abs_residual)
```

Dec 16th, 2023 - Jan 8th, 2024

Merry Christmas + Happy New Year!!!

We made an initial outline for the CYSF presentation, including what to put in the slides, in the paper, and on the trifold. We also figured out how to code for the baseline table.

Classification Models

- 2 categories
    - Below vs above 4 hours (50th percentile) median split
    - Trendline for the accuracy as the split increases (accuracy decreases as the split increases)
- 4 categories
    - Below 25th percentile, 25th - 50th percentile, 50th - 75th percentile, above 75th percentile
- 10 categories
    - Below 10th percentile, 10th - 20th percentile, …, 90th percentile, above 90th percentile
- General trendline for the accuracy as the number of categories increase
    - Explain the reason of why
    - Interestingly, the errors made in models with many categories appear to be concentrated categories with bigger numbers (longer stay)

Regression Model

- Explain results
- Variable importance
- Partial dependence (with 1 and 2 variables)
- Potential patterns in worst predictions

<u>Baseline Characteristics</u>

- Summary command - baseline characteristics (and a bunch of other R packages)
    - https://cran.r-project.org/web/packages/table1/vignettes/table1-examples.html

- https://rstudio.github.io/cheatsheets/html/data-visualization.html

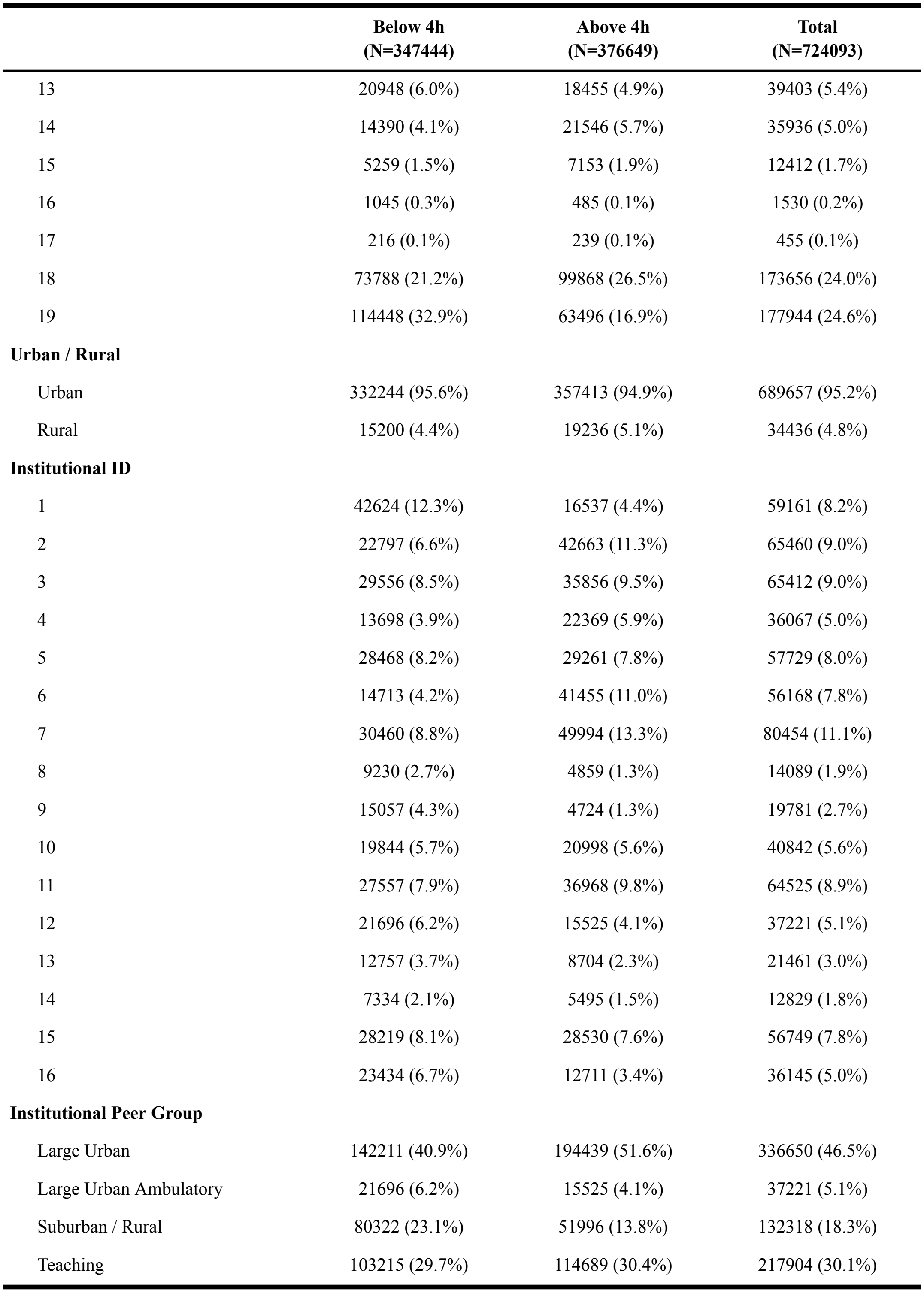
```
# Code for baseline table
# categorize visit_los_minutes (0-4 hours, 4-8 hours, 8-12 hours, Above 12 hours, Total)
ED_sample_no_mv$visit_los_category <- cut(ED_sample_no_mv$visit_los_minutes,
                                    breaks = c(0, 240, Inf),
                                    labels = c("Below 4 Hours", "Above 4 Hours"),
                                    include.lowest = TRUE)


# categorize the postal districts
ED_sample_no_mv$fsa_category <- factor(substr(ED_sample_no_mv$patient_fsa, 1, 2),
                                    levels = unique(substr(ED_sample_no_mv$patient_fsa, 1, 2)))


# Factorize the new variable
ED_sample_no_mv$visit_los_category <- factor(ED_sample_no_mv$visit_los_category,
                                    levels = c("Below 4 Hours", "Above 4 Hours"))


# add variables, all categorical (sex, age_group, prev_hosp, triagecode, icd10_cat, patient_fsa, inst_id,
inst_peer_grp)
ED_sample_no_mv$sex <-
  factor(ED_sample_no_mv$sex, levels=c(1,0),
            labels=c("Male", "Female"))

ED_sample_no_mv$age_group <-
  factor(ED_sample_no_mv$age_group, levels=c(1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19),
            labels=c("1 - 5", "6 - 10", "11 - 15", "16 - 20", "21 - 25", "26 - 30", "31 - 35", "36 - 40", "41 - 45", "46 -
50", "51 - 55", "56 - 60", "61 - 65", "66 - 70", "71 - 75", "76 - 80", "81 - 85", "86 - 90", "90 - 95"))

ED_sample_no_mv$prev_hosp <-
  factor(ED_sample_no_mv$prev_hosp, levels=c(1,0),
            labels=c("Yes", "No"))

ED_sample_no_mv$triagecode <-
  factor(ED_sample_no_mv$triagecode, levels=c(1, 2, 3, 4, 5),
            labels=c("Level 1", "Level 2", "Level 3", "Level 4", "Level 5"))

ED_sample_no_mv$icd10_cat <-
  factor(ED_sample_no_mv$icd10_cat, levels=c(1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21),
            labels=c("I", "II", "III", "IV", "V", "VI", "VII", "VIII", "IX", "X", "XI", "XII", "XIII", "XIV", "XV",
"XVI",
            "XVII", "XVIII", "XIX", "XX", "XXI"))

ED_sample_no_mv$fsa_category <-
  factor(ED_sample_no_mv$fsa_category, levels=c("T0", "T1", "T2", "T3", "T4", "T5", "T6", "T7", "T8", "T9"),
            labels=c("T0", "T1", "T2", "T3", "T4", "T5", "T6", "T7", "T8", "T9"))
```

```
ED_sample_no_mv$inst_peer_grp <-
  factor(ED_sample_no_mv$inst_peer_grp, levels=c("Large Urban Ambulatory", "Large Urban", "Teaching",
"Suburban / Rural"),
         labels=c("Large Urban Ambulatory", "Large Urban", "Teaching", "Suburban / Rural"))

ED_sample_no_mv$inst_id <-
  factor(ED_sample_no_mv$inst_id, levels=c(1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16),
         labels=c("1", "2", "3", "4", "5", "6", "7", "8", "9", "10", "11", "12", "13", "14", "15", "16"))


# change some names
label(ED_sample_no_mv$sex) <- "Biologically Assigned Sex"
label(ED_sample_no_mv$age_group) <- "Age (years)"
label(ED_sample_no_mv$prev_hosp) <- "Previous Hospital Admission"
label(ED_sample_no_mv$triagecode) <- "CTAS Score"
label(ED_sample_no_mv$icd10_cat) <- "ICD 10 Category"
label(ED_sample_no_mv$fsa_category) <- "Patient Postal District"
label(ED_sample_no_mv$inst_peer_grp) <- "Hospital Type"
label(ED_sample_no_mv$inst_id) <- "Hospital ID"


# add table title
caption  <- "Baseline Characteristics"


# print results
table1(~  sex + age_group + prev_hosp + triagecode + icd10_cat + fsa_category +  inst_peer_grp + inst_id |
visit_los_category, data=ED_sample_no_mv,
         overall=c(right="Total"), caption=caption)
```

Baseline Characteristics Table

	Below 4h **(N=347444)**	**Above 4h** **(N=376649)**	**Total** **(N=724093)**
Sex			
Male	171920 (49.5%)	200292 (53.2%)	372212 (51.4%)
Female	175520 (50.5%)	176348 (46.8%)	351868 (48.6%)
Intersex	4 (0.0%)	9 (0.0%)	13 (0.0%)
Age Group			
1 (1-5)	53454 (15.4%)	18581 (4.9%)	72035 (9.9%)
2 (6-10)	25677 (7.4%)	8990 (2.4%)	34667 (4.8%)
3 (11-15)	21937 (6.3%)	9591 (2.5%)	31528 (4.4%)
4 (16-20)	22195 (6.4%)	16143 (4.3%)	38338 (5.3%)
5 (21-25)	23143 (6.7%)	21674 (5.8%)	44817 (6.2%)

	Below 4h (N=347444)	**Above 4h (N=376649)**	**Total (N=724093)**
6 (26-30)	25753 (7.4%)	25886 (6.9%)	51639 (7.1%)
7 (31-35)	27041 (7.8%)	28561 (7.6%)	55602 (7.7%)
8 (36-40)	25004 (7.2%)	27893 (7.4%)	52897 (7.3%)
9 (41-45)	20202 (5.8%)	23930 (6.4%)	44132 (6.1%)
10 (46-50)	17921 (5.2%)	22918 (6.1%)	40839 (5.6%)
11 (51-55)	16655 (4.8%)	23438 (6.2%)	40093 (5.5%)
12 (56-60)	17094 (4.9%)	25319 (6.7%)	42413 (5.9%)
13 (61-65)	14361 (4.1%)	25023 (6.6%)	39384 (5.4%)
14 (66-70)	11396 (3.3%)	22379 (5.9%)	33775 (4.7%)
15 (71-75)	9191 (2.6%)	21535 (5.7%)	30726 (4.2%)
16 (76-80)	7029 (2.0%)	19263 (5.1%)	26292 (3.6%)
17 (81-85)	5152 (1.5%)	17602 (4.7%)	22754 (3.1%)
18 (86-90)	3762 (1.1%)	15568 (4.1%)	19330 (2.7%)
19 (90-95)	477 (0.1%)	2355 (0.6%)	2832 (0.4%)
Previous Hospitalization Status			
Has not been previously hospitalized	243900 (70.2%)	214451 (56.9%)	458351 (63.3%)
Has been previously hospitalized	103544 (29.8%)	162198 (43.1%)	265742 (36.7%)
Triage Code			
1	3064 (0.9%)	5653 (1.5%)	8717 (1.2%)
2	69391 (20.0%)	121737 (32.3%)	191128 (26.4%)
3	157351 (45.3%)	194476 (51.6%)	351827 (48.6%)
4	103274 (29.7%)	49619 (13.2%)	152893 (21.1%)
5	14364 (4.1%)	5164 (1.4%)	19528 (2.7%)
ICD 10 Category			
1	15899 (4.6%)	14089 (3.7%)	29988 (4.1%)
2	677 (0.2%)	3280 (0.9%)	3957 (0.5%)
3	601 (0.2%)	3160 (0.8%)	3761 (0.5%)
4	1845 (0.5%)	8972 (2.4%)	10817 (1.5%)
5	11963 (3.4%)	31793 (8.4%)	43756 (6.0%)
6	5398 (1.6%)	9003 (2.4%)	14401 (2.0%)
7	5629 (1.6%)	2065 (0.5%)	7694 (1.1%)
8	6513 (1.9%)	2687 (0.7%)	9200 (1.3%)
9	8901 (2.6%)	20734 (5.5%)	29635 (4.1%)
10	31398 (9.0%)	29045 (7.7%)	60443 (8.3%)
11	15042 (4.3%)	31685 (8.4%)	46727 (6.5%)
12	13484 (3.9%)	8894 (2.4%)	22378 (3.1%)



	Below 4h (N=347444)	Above 4h (N=376649)	Total (N=724093)
13	20948 (6.0%)	18455 (4.9%)	39403 (5.4%)
14	14390 (4.1%)	21546 (5.7%)	35936 (5.0%)
15	5259 (1.5%)	7153 (1.9%)	12412 (1.7%)
16	1045 (0.3%)	485 (0.1%)	1530 (0.2%)
17	216 (0.1%)	239 (0.1%)	455 (0.1%)
18	73788 (21.2%)	99868 (26.5%)	173656 (24.0%)
19	114448 (32.9%)	63496 (16.9%)	177944 (24.6%)
Urban / Rural			
Urban	332244 (95.6%)	357413 (94.9%)	689657 (95.2%)
Rural	15200 (4.4%)	19236 (5.1%)	34436 (4.8%)
Institutional ID			
1	42624 (12.3%)	16537 (4.4%)	59161 (8.2%)
2	22797 (6.6%)	42663 (11.3%)	65460 (9.0%)
3	29556 (8.5%)	35856 (9.5%)	65412 (9.0%)
4	13698 (3.9%)	22369 (5.9%)	36067 (5.0%)
5	28468 (8.2%)	29261 (7.8%)	57729 (8.0%)
6	14713 (4.2%)	41455 (11.0%)	56168 (7.8%)
7	30460 (8.8%)	49994 (13.3%)	80454 (11.1%)
8	9230 (2.7%)	4859 (1.3%)	14089 (1.9%)
9	15057 (4.3%)	4724 (1.3%)	19781 (2.7%)
10	19844 (5.7%)	20998 (5.6%)	40842 (5.6%)
11	27557 (7.9%)	36968 (9.8%)	64525 (8.9%)
12	21696 (6.2%)	15525 (4.1%)	37221 (5.1%)
13	12757 (3.7%)	8704 (2.3%)	21461 (3.0%)
14	7334 (2.1%)	5495 (1.5%)	12829 (1.8%)
15	28219 (8.1%)	28530 (7.6%)	56749 (7.8%)
16	23434 (6.7%)	12711 (3.4%)	36145 (5.0%)
Institutional Peer Group			
Large Urban	142211 (40.9%)	194439 (51.6%)	336650 (46.5%)
Large Urban Ambulatory	21696 (6.2%)	15525 (4.1%)	37221 (5.1%)
Suburban / Rural	80322 (23.1%)	51996 (13.8%)	132318 (18.3%)
Teaching	103215 (29.7%)	114689 (30.4%)	217904 (30.1%)

Jan 15th, 2024

Another meeting with Jessalyn, we had quite a few questions to ask her regarding some of the code and just a few other general questions.

1. Why is random forest preferred for this dataset not not machine learning algorithms?
2. Accuracy for the classification model?
3. Full baseline table?
4. Absolute residual and other results for the regression model?
5. Can we have the pdp for triage code pls?
6. When we increase the time break for the 2 category one, is that overfitting or is that just the trend of the dataset?
7. How to do partial dependence for 2 variables?

Jan 16th - 24th, 2024

A new idea came to mind: we wanted to try predicting whether the patient is discharged or admitted into hospital.

```
# factor disp_group
ED_Data$disp_group <- as.factor(ED_Data$disp_group)
levels(ED_Data$disp_group) <- c('1', '4', '2', '3', '5', '6')

ED_Data$disp_group <- as.numeric(as.character(ED_Data$disp_group))

ED_Data[ , 'disp_group'] <- NA

# A = Admitted, B = Discharged
ED_Data$disp_group <- ifelse(ED_Data$disp_group == 1, 'A',
                             ifelse(ED_Data$disp_group == 2, 'B', 'C'))

ED_Data$disp_group <- as.factor(ED_Data$disp_group)
levels(ED_Data$disp_group) <- c('Admitted', 'Discharged', 'Other')

# the model
rf_disp_group <- randomForest(disp_group ~ icd10_cat + triagecode + age_group + prev_hosp + inst_id +
inst_peer_grp + patient_fsa + sex,
                    data = train,
                    na.action = na.omit)

# print results
print(rf_disp_group)

# variable importance
varImpPlot(rf_disp_group)

# testing
test$predict_disp_group <- predict(rf_disp_group, newdata = test)
confusion_matrix_test_disp_group <- table(test$disp_group, test$predict_disp_group)
print(confusion_matrix_test_disp_group)
```

Jan, 24th, 2024

Another meeting with Jessayln, we are going to start organizing the code to run on the full dataset. Because the full dataset is so large, it cannot run on our computers, so it needs to run on a high power computer. We also talked about some interesting aspects of the tables generated from the worst predictions. It shows that in some areas the prediction is consistently worse than others. The ones with very long lengths of stays are often the most complicated.

Jan 25th - Feb. 9th, 2024

Organized all the code that we need to run on the full dataset, all models are trained with the training set, and tested on the testing set. We also started to make the presentation and trifold, it provides a rough outline of what we need to talk about.

```
# Splitting into train-test sets, cutting variables

# cutting for 2 hours
ED_sample_no_mv$two_hours <- cut(ED_sample_no_mv$visit_los_minutes,
                    breaks = c(-Inf, 120, Inf),
                    labels = c("Below 2h", "Above 2h"))

# cutting for 4 hours
# cutting for 6 hours
…

#Making Test and Training Sets
sample <- sample(c(TRUE, FALSE), nrow(ED_sample_no_mv), replace=TRUE, prob=c(0.7, 0.3))
train <- ED_sample_no_mv[sample, ]
test <- ED_sample_no_mv[!sample, ]
```

The classification models

- With different time breaks (2 hours, 4 hours, 6 hours, 8 hours, 10 hours, 12 hours)
- With different ntree (on 4 hours, 6 hours, 8 hours classification models)
- With different mtry (on 4 hours classification models)
- With different number of variables

The regression model

- Model
- Variable importance
- Partial dependence for all the variables

```
# Model
```

```
rf_regress <- randomForest(visit_los_minutes ~ age_group + inst_id + inst_peer_grp + sex + patient_fsa +
triagecode + icd10_cat + prev_hosp,
            data = train,
            importance = TRUE,
            na.action = na.omit)

print(rf_regress)

# Testing
test$predict <- predict(rf_regress, newdata = test)
ED_sample_no_mv$predict <- predict(rf_regress, newdata = ED_sample_no_mv)
test$residual <- test$predict - test$visit_los_minutes
test$abs_residual <- abs(test$residual)
summary(test$abs_residual)
```

Misclassified/Mispredicted Table for Classification Models and the Regression Model

```
# Misclassified in classification

# models (4 and 6 hours, on full dataset)

# 4 hours model
rf_four_hours_2 <- randomForest(four_hours ~ age_group + triagecode + sex + inst_peer_grp + inst_id +
patient_fsa + prev_hosp + icd10_cat,
                        data = ED_sample_no_mv,
                        importance = TRUE,
                        na.action = na.omit)

# 6 hours model
rf_six_hours_2 <- randomForest(six_hours ~ age_group + triagecode + sex + inst_peer_grp + inst_id + patient_fsa +
prev_hosp + icd10_cat,
                        data = ED_Data,
                        importance = TRUE,
                        na.action = na.omit)


# ------ Identifying Different Values ------

ED_sample_no_mv$predResults4 <- predict(rf_four_hours_2)
ED_sample_no_mv$predResults6 <- predict(rf_six_hours_2)

#long = 2, short =
levels(ED_sample_no_mv$predResults4) <- c('1', '2')
levels(ED_sample_no_mv$predResults6) <- c('1', '2')
levels(ED_sample_no_mv$four_hours) <- c('1', '2')
levels(ED_sample_no_mv$six_hours) <- c('1', '2')

ED_sample_no_mv$predResults4 <- as.numeric(as.character(ED_sample_no_mv$predResults4))
ED_sample_no_mv$predResults6 <- as.numeric(as.character(ED_sample_no_mv$predResults6))
```

```
ED_sample_no_mv$four_hours <- as.numeric(as.character(ED_sample_no_mv$four_hours))
ED_sample_no_mv$six_hours <- as.numeric(as.character(ED_sample_no_mv$six_hours))

ED_sample_no_mv[ , 'results4'] <- NA
ED_sample_no_mv[ , 'results6'] <- NA
ED_sample_no_mv[ , 'final4'] <- NA
ED_sample_no_mv[ , 'final6'] <- NA

# 1 = same, 2 = different
ED_sample_no_mv$results4 <- ifelse(ED_sample_no_mv$four_hours == ED_sample_no_mv$predResults4, '1', '2')
ED_sample_no_mv$results6 <- ifelse(ED_sample_no_mv$six_hours == ED_sample_no_mv$predResults6, '1', '2')

ED_sample_no_mv$results4 <- as.numeric(as.character(ED_sample_no_mv$results4))
ED_sample_no_mv$results6 <- as.numeric(as.character(ED_sample_no_mv$results6))

# A = Classified Right, B = Classified Wrong
ED_sample_no_mv$final4 <- ifelse(ED_sample_no_mv$results4 < 2, 'A', 'B')

# C = Classified Right, D = Classified Wrong
ED_sample_no_mv$final6 <- ifelse(ED_sample_no_mv$results6 < 2, 'C', 'D')

table(ED_sample_no_mv$final4)
table(ED_sample_no_mv$final6)

# --------- Tables ---------

# load packages
library(boot)
library(table1)

# categorize the postal districts
ED_sample_no_mv$fsa_category <- factor(substr(ED_sample_no_mv$patient_fsa, 1, 2),
                           levels = unique(substr(ED_sample_no_mv$patient_fsa, 1, 2)))

# add variables, all categorical (sex, age_group, prev_hosp, triage code, icd10_cat, patient_fsa, inst_id,
inst_peer_grp)
ED_sample_no_mv$sex <-
  factor(ED_sample_no_mv$sex, levels=c(1,0),
         labels=c("Male", "Female"))

ED_sample_no_mv$age_group <-
  factor(ED_sample_no_mv$age_group, levels=c(1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19),
         labels=c("1 - 5", "6 - 10", "11 - 15", "16 - 20", "21 - 25", "26 - 30", "31 - 35", "36 - 40", "41 - 45", "46 -
50", "51 - 55", "56 - 60", "61 - 65", "66 - 70", "71 - 75", "76 - 80", "81 - 85", "86 - 90", "90 - 95"))

ED_sample_no_mv$prev_hosp <-
  factor(ED_sample_no_mv$prev_hosp, levels=c(1,0),
         labels=c("Yes", "No"))
```

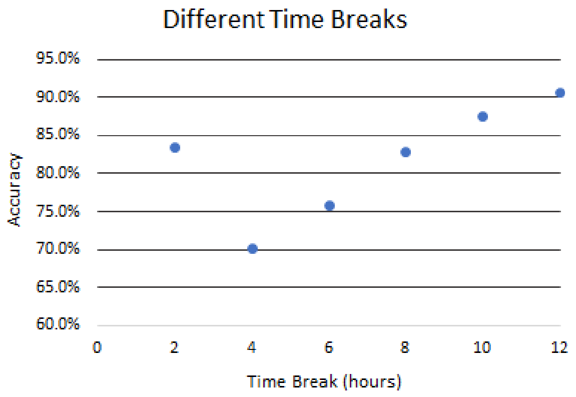
```
ED_sample_no_mv$triagecode <-
  factor(ED_sample_no_mv$triagecode, levels=c(1, 2, 3, 4, 5),
          labels=c("Level 1", "Level 2", "Level 3", "Level 4", "Level 5"))

ED_sample_no_mv$icd10_cat <-
  factor(ED_sample_no_mv$icd10_cat, levels=c(1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21),
          labels=c("Category I", "Category II", "Category III", "Category IV", "Category V", "Category VI",
"Category VII",
          "Category VIII", "Category IX", "Category X", "Category XI", "Category XII", "Category XIII", "Category
XIV",
          "Category XV", "Category XVI", "Category XVII", "Category XVIII", "Category XIX", "Category XX",
"Category XXI"))

ED_sample_no_mv$fsa_category <-
  factor(ED_sample_no_mv$fsa_category, levels=c("T0", "T1", "T2", "T3", "T4", "T5", "T6", "T7", "T8", "T9"),
          labels=c("T0", "T1", "T2", "T3", "T4", "T5", "T6", "T7", "T8", "T9"))

ED_sample_no_mv$inst_id <-
  factor(ED_sample_no_mv$inst_id, levels=c(1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16),
          labels=c("1", "2", "3", "4", "5", "6", "7", "8", "9", "10", "11", "12", "13", "14", "15", "16"))

ED_sample_no_mv$inst_peer_grp <-
  factor(ED_sample_no_mv$inst_peer_grp, levels=c("Large Urban Ambulatory", "Large Urban", "Teaching",
"Suburban / Rural"),
          labels=c("Large Urban Ambulatory", "Large Urban", "Teaching", "Suburban / Rural"))

ED_sample_no_mv$final4 <-
  factor(ED_sample_no_mv$final4, levels=c("A", "B"),
          labels=c("Correctly Predicted", "Incorrectly Predicted"))

ED_sample_no_mv$final6 <-
  factor(ED_sample_no_mv$final6, levels=c("C", "D"),
          labels=c("Correctly Predicted", "Incorrectly Predicted"))

# change some names
label(ED_sample_no_mv$sex) <- "Biologically Assigned Sex"
label(ED_sample_no_mv$age_group) <- "Age (years)"
label(ED_sample_no_mv$prev_hosp) <- "Previous Hospital Admission"
label(ED_sample_no_mv$triagecode) <- "CTAS Score"
label(ED_sample_no_mv$icd10_cat) <- "ICD 10 Category"
label(ED_sample_no_mv$fsa_category) <- "Patient Postal District"
label(ED_sample_no_mv$inst_id) <- "Hospital ID"
label(ED_sample_no_mv$inst_peer_grp) <- "Hospital Type"

# add table title
caption4  <- "Baseline Characteristics of Classification Model with 4 Hours Timebreak"

# print results
```

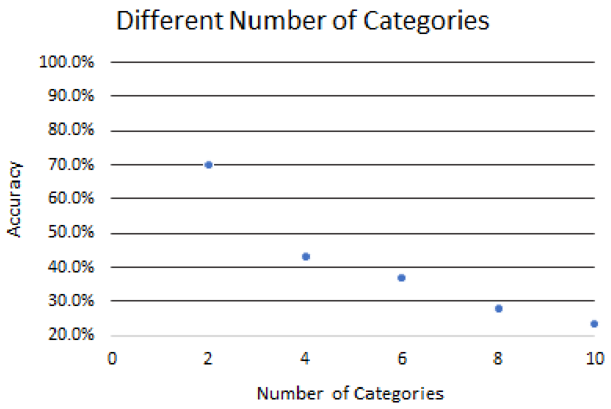
```
table1(~  sex + age_group + prev_hosp + triagecode + icd10_cat + fsa_category + inst_id + inst_peer_grp | final4,
data=ED_sample_no_mv,
       caption=caption4)

# add table title
caption6  <- "Baseline Characteristics of Classification Model with 6 Hour Timebreak"

# print results
table1(~  sex + age_group + prev_hosp + triagecode + icd10_cat + fsa_category + inst_id + inst_peer_grp | final6,
data=ED_sample_no_mv,
       caption=caption6)



# Mispredicted in regression

# Model
rf_regress_2 <- randomForest(visit_los_minutes ~ age_group + inst_id + inst_peer_grp + sex + patient_fsa +
triagecode + icd10_cat + prev_hosp,
              data = ED_sample_no_mv,
              importance = TRUE,
              na.action = na.omit)


# Finding the > 10 hours difference - Categorize the absolute residuals
ED_sample_no_mv$long_stay <- cut(ED_sample_no_mv$abs_residual,
              breaks=c(-Inf, 240, 360, 480, 600, Inf),
              labels=c("A", "B", "C", "D", "E"))

# --------- TABLE----------

# load packages
library(boot)
library(table1)

# categorize the postal districts
ED_sample_no_mv$fsa_category <- factor(substr(ED_sample_no_mv$patient_fsa, 1, 2),
              levels = unique(substr(ED_sample_no_mv$patient_fsa, 1, 2)))

# add variables, all categorical (sex, age_group, prev_hosp, triagecode, icd10_cat, patient_fsa, inst_id,
inst_peer_grp)
ED_sample_no_mv$sex <-
  factor(ED_sample_no_mv$sex, levels=c(1,0),
       labels=c("Male", "Female"))

ED_sample_no_mv$age_group <-
  factor(ED_sample_no_mv$age_group, levels=c(1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19),
       labels=c("1 - 5", "6 - 10", "11 - 15", "16 - 20", "21 - 25", "26 - 30", "31 - 35", "36 - 40", "41 - 45", "46 -
50", "51 - 55", "56 - 60", "61 - 65", "66 - 70", "71 - 75", "76 - 80", "81 - 85", "86 - 90", "90 - 95"))
```

```
ED_sample_no_mv$prev_hosp <-
  factor(ED_sample_no_mv$prev_hosp, levels=c(1,0),
            labels=c("Yes", "No"))

ED_sample_no_mv$triagecode <-
  factor(ED_sample_no_mv$triagecode, levels=c(1, 2, 3, 4, 5),
            labels=c("Level 1", "Level 2", "Level 3", "Level 4", "Level 5"))

ED_sample_no_mv$icd10_cat <-
  factor(ED_sample_no_mv$icd10_cat, levels=c(1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21),
            labels=c("Category I", "Category II", "Category III", "Category IV", "Category V", "Category VI",
"Category VII",
            "Category VIII", "Category IX", "Category X", "Category XI", "Category XII", "Category XIII", "Category
XIV",
            "Category XV", "Category XVI", "Category XVII", "Category XVIII", "Category XIX", "Category XX",
"Category XXI"))

ED_sample_no_mv$fsa_category <-
  factor(ED_sample_no_mv$fsa_category, levels=c("T0", "T1", "T2", "T3", "T4", "T5", "T6", "T7", "T8", "T9"),
            labels=c("T0", "T1", "T2", "T3", "T4", "T5", "T6", "T7", "T8", "T9"))

ED_sample_no_mv$inst_id <-
  factor(ED_sample_no_mv$inst_id, levels=c(1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16),
            labels=c("1", "2", "3", "4", "5", "6", "7", "8", "9", "10", "11", "12", "13", "14", "15", "16"))

ED_sample_no_mv$inst_peer_grp <-
  factor(ED_sample_no_mv$inst_peer_grp, levels=c("Large Urban Ambulatory", "Large Urban", "Teaching",
"Suburban / Rural"),
            labels=c("Large Urban Ambulatory", "Large Urban", "Teaching", "Suburban / Rural"))

ED_sample_no_mv$long_stay <-
  factor(ED_sample_no_mv$long_stay, levels=c("A", "B", "C", "D", "E"),
            labels=c("Mispredicted by <4 Hours", " Mispredicted by 4-6 Hours", "Mispredicted by 6-8 Hours",
"Mispredicted by 8-10 Hours", "Mispredicted by >10 Hours"))

# change some names
label(ED_sample_no_mv$sex) <- "Biologically Assigned Sex"
label(ED_sample_no_mv$age_group) <- "Age (years)"
label(ED_sample_no_mv$prev_hosp) <- "Previous Hospital Admission"
label(ED_sample_no_mv$triagecode) <- "CTAS Score"
label(ED_sample_no_mv$icd10_cat) <- "ICD 10 Category"
label(ED_sample_no_mv$fsa_category) <- "Patient Postal District"
label(ED_sample_no_mv$inst_id) <- "Hospital ID"
label(ED_sample_no_mv$inst_peer_grp) <- "Hospital Type"
label(ED_sample_no_mv$long_stay) <- "Below vs Above 10 Hours Residual"

# add table title
caption  <- "Characteristics of Mispredicted Patients in Regression Model"
```

```
# print results
table1(~  sex + age_group + prev_hosp + triagecode + icd10_cat + fsa_category + inst_id + inst_peer_grp |
long_stay, data=ED_sample_no_mv,
          caption=caption)
```

Feb 9th, 2024
This is the results meeting, we went to Jessalyn's office and saw most of the results for the models.





Feb 10th - 15th, 2024
We decided to change the style of the slides, more work on the slides is done. It has a dark background now, which makes the text stand out more. We also started writing the paper since we do have some of the results now.

Feb 16th, 2024
We asked Jessalyn to take a look at our paper. She gave a few comments regarding what to add or change. Some of the codes are still running on her high power computer.

Feb 17th - 29th, 2024
Mostly writing the paper, editing the slides, and designing the trifold.

Mar 1st, 2024

Another meeting with Jesalyn, we got results for the misclassified and mispredicted analysis. The Disposition Group model is still running. There are some interesting trends for the misclassified and mispredicted tables.

MISCLASSIFIED
Sex: accuracy is not really affected by the change in time break --> indicates higher unpredictability and its low influence compared to other variables.

Age group: When the time break increases, for the younger age groups, the incorrect prediction rate decreases. For some of the ages, compared to the 6 hours prediction, more than half of the people who were misclassified for the 4 hours model got classified correctly when it comes to the 6 hours model. For ages 61 - 95, when the time break increases, the incorrect prediction rates also slightly increase. According to the partial dependence, we see that for these age groups the length of stay is all higher than 6 hours. However, there are more people in the younger age groups, and their stay times are mostly less than the ones in the higher age groups. For the four hours model, the machine would categorize most of the older patients into above 4 hours, because most of them do stay for longer than 4 hours. But when it comes to the 6 hours model, the machine becomes less sure about whether the patient is below or above 6 hours, thus more likely to classify them into the incorrect category. This is potentially why we see a slight increase in the incorrect prediction rate when the age group gets larger.

Previous hospitalization: The incorrect predictions increase when the time break increases, this is likely because people who have been hospitalized previously are more likely to stay for longer and their conditions are likely more unpredictable. For the 4 hours model, the machine is likely to classify most of them as long stays, but for the 6 hours model, there are more people with previous hospitalization who stay for around that 6 hours time breaks, thus making it more difficult to predict.

Triage code: For levels 1 and 2, the number of incorrect predictions is fairly consistent, with a very slight increase, this is likely due to the high unpredictability of the patients who fall under this category, as the stay time is more likely to vary significantly in severe conditions. For levels 3-5, as the conditions are not as severe, they are more likely to fall under the shorter than 6 hours category for the 6 hours models, thus increasing the accuracy for these 3 levels.

ICD 10: Categories 2-5 are the ones with the longest stay times, and the number of incorrect predictions do increase as the time break increases to 6 hours. Once again, it is likely caused by the unpredictability of these categories, some patients may stay for very long while others may stay for a very short amount of time. For the 6 hours model, the machine may incorrectly classify some of the patients as under 6 hours. This is potentially why there is an increase in the number of incorrect predictions for these categories.

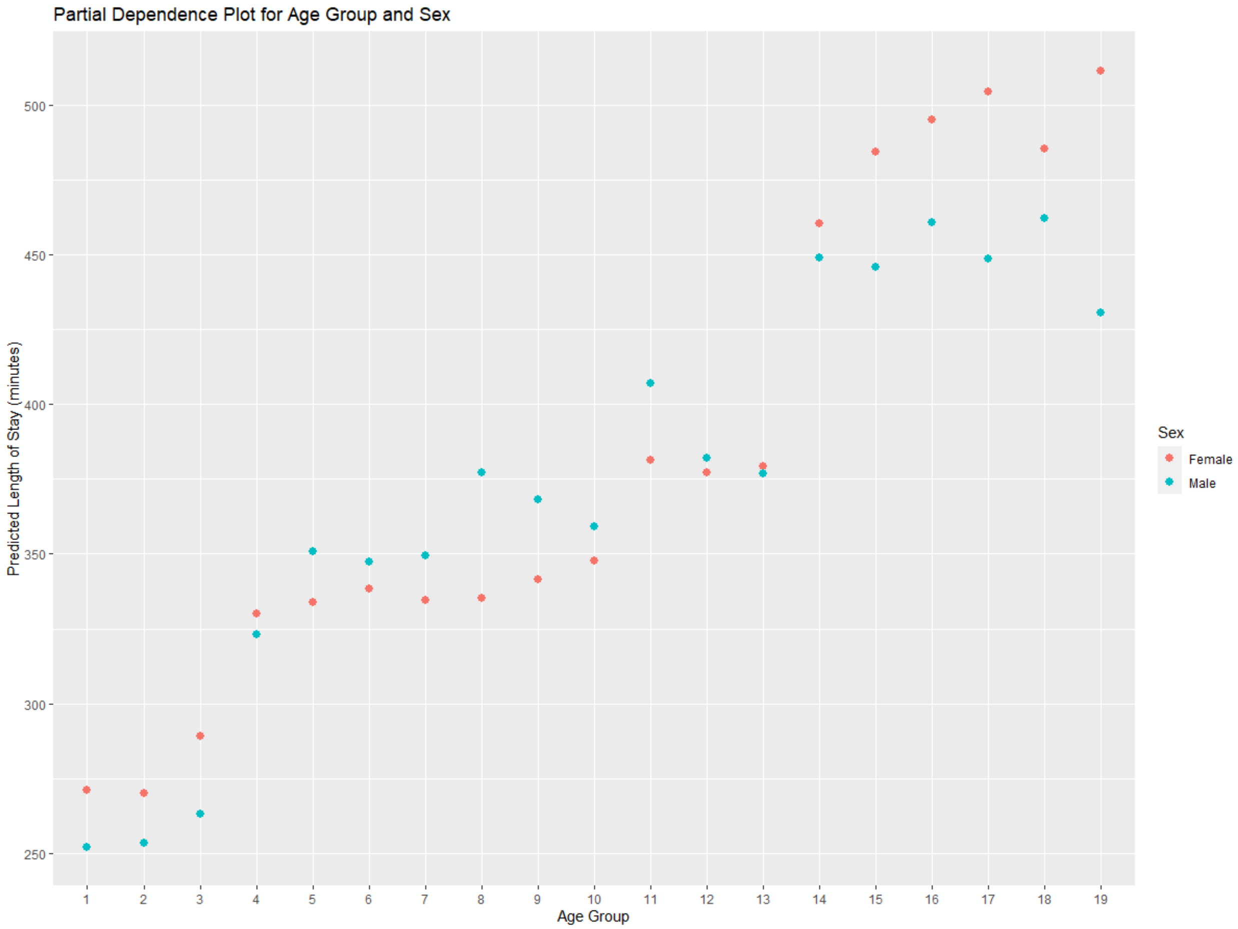
MISPREDICTED
For all of the different sections of the variables, there are the most number of people in the mispredcited by less than 4 hours category. For some of them, there are more people mispredicted in the greater than 10 hours category compared to the categories with mispredictions by 6 to 10 hours.

Mar 1st - 6th, 2024

Rough draft of the trifold is done. Learned how to code for 2 variable partial dependence. In that way, two variables can be seen and compared in one graph.

```
rf_regress_1 <- randomForest(visit_los_minutes ~ age_group + sex,
                        data = ED_sample_no_mv,
                        na.action = na.omit)

age.sex.pdp <- partial(rf_regress_1, pred.var = c("sex", "age_group"), data = ED_sample_no_mv)
```

```
    # AGE + SEX Paper Color
    age_sex_paper <- ggplot(data = age.sex.pdp, aes(x = age_group, y = yhat)) +
    geom_point(aes(colour = factor(sex)), size = 2.5) +  # Adjust size here
    labs(title = "Partial Dependence Plot for Age Group and Sex",
    x = "Age Group",
    y = "Predicted Length of Stay (minutes)") +
    scale_color_discrete(name = "Sex", labels = c("Female", "Male"))

    print(age_sex_paper)
```



2 variables PDP graphs were made for:

- ICD 10 + triage code
- Triage code + previous hospitalization status
- Age group + previous hospitalization status
- Age group + triage code
- Age group + sex

Mar 7th, 2024

Last meeting with Jessalyn for this project… Thank you.

Roughly discussed interesting trends we noticed for the 2 variable PDP graphs, we also got our disposition group model results, the accuracy is 81.2%.

Confusion Matrix

	Actual Values

		Admitted	**Discharged**	**Other**	**Predictive Values**
Predicted Values	**Admitted**	8278	52208	22	**13.7%**
	Discharged	4108	169920	128	**97.6%**
	Other	430	14708	198	**1.3%**
	Accuracy Out of Actual Values	**64.5%**	**71.7%**	**56.9%**	

Mar 8th - 14th, 2024

We got the 2 variable partial dependence graphs finished. This week the focus was primarily on the presentation aspects. On Wednesday, the slideshow was finished, video was recorded, paper was done and uploaded onto the CYSF platform.