Georgia Institute of Technology

ISYE/CSE/MGT 6748 Applied Analytics Practicum

Patient Health Risk Prediction

Michael Guo, Vazirani Yogesh & Sarfroz Nawaz Katiyar Hyder Ali

1. DISTRIBUTION OF WORK

Michael Guo – Exploratory Data Analysis, Determining Predictor Variables, Data Cleaning and Wrangling, Dataset Variable Investigation, Auto-Selection Method and Function, and Logistic Regression Model

Yogesh Vazirani – Exploratory Data Analysis, Determining Predictor Variables, Neural Network Model

Sarfroz Nawaz Kasiyar Hyder Ali – Exploratory Data Analysis, Determining Predictor Variables, Random Forest Model

2. INTRODUCTION

Capgemini is a multinational IT services and consulting company that is focused on partnering with companies to transform and manage business through the power of technology. As part of its efforts within the medical field it wants to:

- Identify correlated patient journeys and improve treatment pathways – Using historical data to detect patterns and predict outcomes for patients with similar health conditions
- Enable personalization of interventions through early detection and prediction of clinical risks and thus reduce the risk of future hospital readmissions

Our work on this project is focused on predicting patient health risk, that is, given a history of patient diagnoses, determine which patients are likely to develop a particular illness in the future.

The biggest benefit this project correlates directly with is that access to patient health risk conditions allows health care providers, insurance companies and others to provide preventive guided care to patients. There are many clinical models available which are possibly more suitable for this use case based on a patient's clinical information. However, for this project we are employing a novel method wherein we use medical claims (non-clinical) information to assess and predict patient risk conditions.

3. DEFINITION TABLE

Term	Definition
Accuracy	$\frac{TP + TN}{TP + FN + TN + FP}$
Sensitivity	$\frac{TP}{TP + FN}$
Balanced Accuracy	$\frac{Sensitivity + Specificity}{2}$
Area Under Curve (AUC)	The degree or measure of separability between the binary classes, that is whether the diagnosis for the condition is positive or negative
ICD10	International Classification of Disease – Catalog clinical information to help health care industry, insurance and government carriers know what diagnoses patients have gone through. Also used for Claims billing.

4. PROBLEM STATEMENT

If high-risk disease can be predicted by early recognition, it can reduce the risk of life-threatening conditions and prevent costs associated with treatment of disease. Given a history of patient diagnoses (provided in claims data), identify how likely the patients are to develop certain illnesses in the future with medical and pharmacy claims data.

5. ASSUMPTIONS/CONSTRAINTS

- Our model is constrained in the prediction of future health developments in patients because of several factors such as the unavailability of clinical information.
- Current market trends do not tend to use claims information to estimate patient health risk.
- We believe the focus should be on predicting patient conditions/diseases that contribute most to the mortality rate in the United States because we feel that will have the most impact to improving the current state.
- Because the dataset is limited to portions of the Eastern United States, the solution may not be scalable across the US.
- We believe that the model performance on the chosen ICD10 Base Codes (C18, C50, E11, I10, I25, and N18) are indicative of our models' performance on other ICD10 Base Codes.

Predicted ICD10 Base Code	Description
C18	Malignant neoplasm of colon
C50	Malignant neoplasm of breast.
E11	Type 2 Diabetes Mellitus
I10	Essential (primary) hypertension
I25	Chronic ischemic heart disease

N18	Chronic Kidney Disease (CKD)

Table 1: *ICD10 Base Codes predicted for this report*

6. APPROACH AND METHODOLOGY

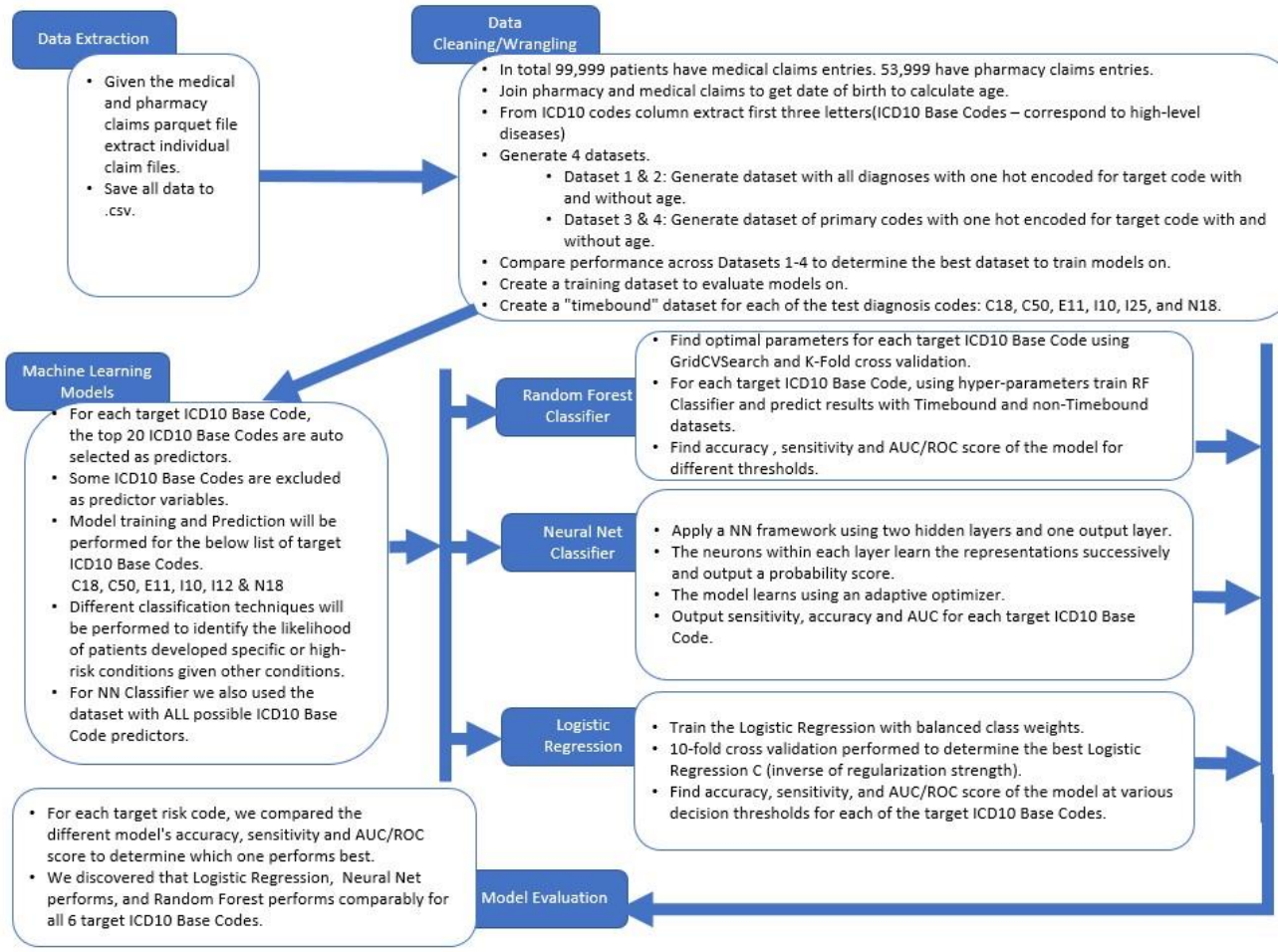

Fig 1: *High Level Process Flow Diagram*

We decided to provide Capgemini with a function that will auto-generate models to predict a target ICD10 Base Code that they can input into it. We also used ICD10 Base Codes (categories) as predictors instead of full ICD10 diagnosis codes. We define an ICD10 Base Code as the first three characters of an ICD10 code. These three initial characters provide the general category of the disease. ICD10 Base Codes are used, because we are only interested in whether a patient was diagnosed with the general category of disease and do not require other vital details provided in the rest of the code.

Our function is divided into two portions: auto-selecting ICD10 Base Codes to train our model(s) on and the actual modeling itself. The first portion chooses the most highly correlated ICD10 Base Codes to be used in predicting a patient's likelihood of developing the target

disease. It then alters the training data used to train the model(s) in the second half appropriately.

For the auto-selection portion of our code, we created an edges file that shows the co-occurrence of different ICD10 Base Codes within patients and an exclusions file of ICD10 Base Codes to ignore. For the modeling part we created a dataset of one-hot encoded ICD10 Base Codes with a few other predictor variables being created. During the data wrangling process, we investigated how the inclusion/exclusion of age and use of All Diagnoses or only Primary Diagnoses when generating one-hot encoded datasets affects model effectiveness. Ultimately, we decided to include Age as variable and use all diagnoses when training our models.

We investigated three potential models: Logistic Regression, Random Forest, and Neural Network. Due to concerns that our initial models were using diagnosis claims caused by the diagnosis of interest to predict the target diagnosis, we also evaluated investigated model performance when trained on "Timebound" datasets. These Timebound datasets are designed to limit training data to entries chronologically on or before the diagnosis predicted.

7. EXPLORATORY DATA ANALYSIS

Capgemini presented us with two relevant sets of data to work with this project: a Medical Claims Dataset and a Pharmacy Claims Dataset. The two datasets consist of claims data for 99,999 Member Life IDs (patients) primarily concentrated in the Eastern United States from 2015 to 2018. Slightly more than half of Member Life IDs (53,999) in the Medical Claims Dataset had corresponding entries in the Pharmacy Claims Dataset.

We investigated the following variables as predictor variables in our models:

- **ICD10 Base Codes:** There are three types of ICD10 Diagnosis codes provided in the Medical Claims Dataset: Primary, Secondary, and Tertiary. The Primary Diagnosis codes represent the reason a patient visited a hospital and what the hospital is billed for. The Secondary and Tertiary Diagnosis codes supply more information to complement the primary diagnosis. The group was interested in how using all codes or solely using Primary Diagnosis codes when creating training data impacts model effectiveness. So, before model evaluation, we evaluated whether using all codes or only primary diagnosis codes would yield more effective models.
- **Age** (Calculated as the year difference between first Header From Service Date and Member Life ID Birth Date): The prevalence of multiple diseases has been shown to increase as patients age. However, since not all the Member Life IDs in the Medical Claims Dataset have corresponding entries in the Pharmacy Claims Dataset (which includes Member Life ID Birth Date), we would need to exclude a sizable portion of the Medical Claims Dataset (46,000 Member Life IDs). To determine whether the benefits of including Age as a variable outweighs the cost of losing those 46,000 Member Life IDs, we evaluated their effect on a Logistic Regression model's performance.

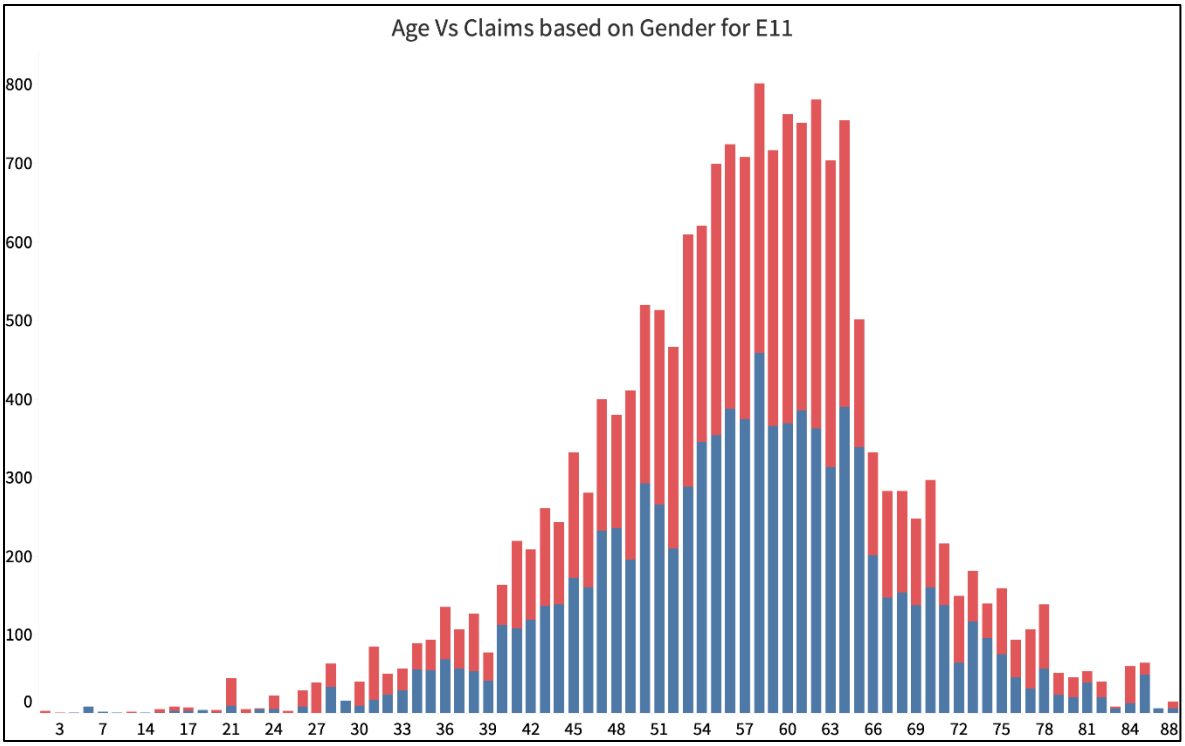

Fig 2: *Age contribution to number of Claims (E11)*

- **Biological Gender (Sex)**: We decided to use Sex as a predictor variable, because some diseases have been known to be more prevalent in one sex as compared to the other.

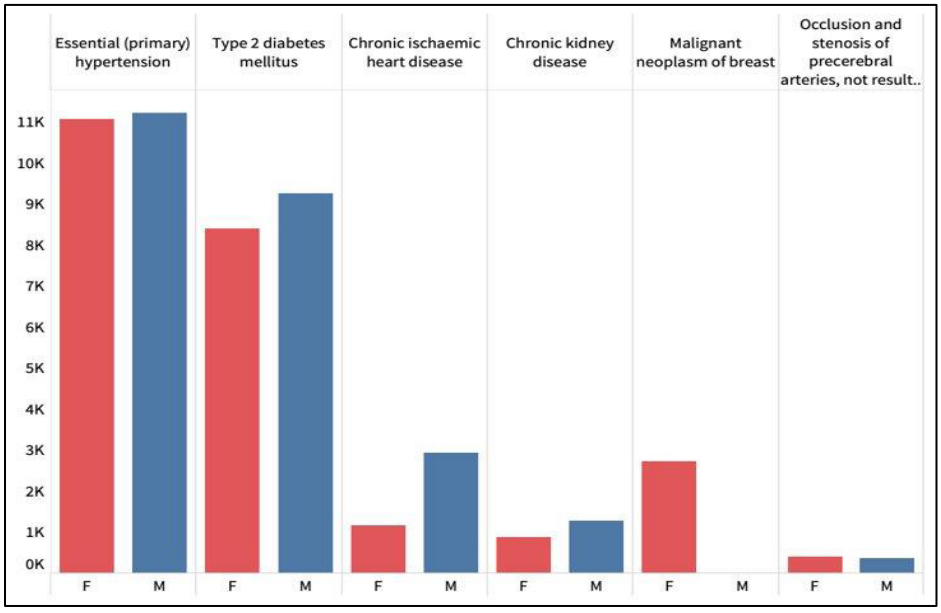

Fig 3: *Number of Claimants by Gender for certain Chronic Conditions*

- **Subscriber Zip Code:** We considered Subscriber Zip Code as a potential variable, because geographical location can be an indicator of disease status for a variety of factors including socioeconomic status of the areas as well as environmental factors

(World Health Organization, 2016). However, we decided against using it as a predictor variable when we discovered that the dataset is limited to the eastern United States.

8.DATA CLEANING/WRANGLING

8.1. EDGES AND EXCLUSIONS LIST FOR AUTO-SELECTION

To facilitate our Auto-Selection method, we generated an undirected graph and an exclusions list. For the undirected graph, the nodes represent individual ICD10 Base Codes, and the edges represent whether the nodes in a pair are both observed with at least a single Member Life ID. The weight of the edges is the number of Member Life IDs where a relationship between both ICD10 Base Code nodes is observed.

Since ICD10 diagnosis codes do not exclusively report disease diagnoses (they also capture things such as encounters for immunization and symptoms) and some temporary mild diseases (such as colds) cannot be reasonably be expected to contribute to the development of more serious chronic conditions, an exclusion list of ICD10 Base Codes that should be skipped when selecting predicting variables.

This list was created by repeatedly running a function to identify the top 20 ICD10 Base Codes to use as predictor variables for a target ICD10 Base Code multiple times. Each time it was run, we manually recorded the ICD10 Base Codes to exclude in the exclusion list. We continued to run the function while adding to the list of exclusions until it produced 20 ICD10 Base Codes (sans excluded ICD10 Base Codes) that could reasonably be expected to contribute to development of the target ICD10 Base Code. This process was performed for each of the target ICD10 Base Codes we predicted with our models: C18, C50, E11, I10, I25, and N18. The same exclusion list was kept through each iteration of this process and was used to identify ICD10 Base Codes to exclude for all models reported. The final top 20 predictor ICD10 Base Codes identified excluding items in the exclusions list are shown in Table 2.

Y	X1	X2	X3	X4	X5	X6	X7	X8	X9	X10	X11	X12	X13	X14	X15	X16	X17	X18	X19	X20
C18	I10	E78	K63	D12	M79	Z86	D64	M25	C78	K57	Z45	K62	K76	M54	N39	K21	E11	I25	Z90	E66
C50	I10	E78	M25	M79	M54	N63	Z90	Z80	N64	K21	L82	H25	D05	E03	M85	Z86	F41	E11	N39	D48
E11	I10	E78	M25	M54	M79	E66	H25	G47	I25	K21	E03	B35	H40	N39	M47	Z86	D64	M17	M19	F41
I10	E78	M25	M54	E11	M79	E66	K21	H25	G47	I25	F41	L82	E03	M47	M17	Z86	N39	M19	L57	J30
I25	I10	E78	M25	E11	M79	M54	Z95	H25	K21	G47	L82	Z86	I48	L57	M47	D64	I50	I51	I49	M19
N18	I10	E78	E11	I12	M25	M79	M54	I25	E87	D64	N17	N28	H25	K21	N39	I50	G47	Z86	I51	B35

Table 2: *Target Condition(Y) – With its respective predictors*

8.2. TRAINING DATA PREDICTOR INVESTIGATION - DATASET PREPARATION

We prepared 4 datasets to investigate the effects of including an Age predictor variable and of using All Diagnoses vs. Primary Diagnoses. In all 4 datasets, we started off with one row for each of the unique Member Life IDs and columns for each of the unique ICD10 Base Codes in the Medical Claims Dataset. Then in the All Diagnoses datasets (Dataset 1 and Dataset 2), we one-hot encoded columns for each of the ICD10 Base Codes if it was seen in that Member Life ID (1 if present, 0 if not present). In the Primary Diagnoses Only dataset, we only have one hot encoded ICD10 Base Codes if they were seen in that Member Life ID and were a Primary Diagnosis Code (Dataset 3 and Dataset 4). A breakdown of the factors affecting each dataset can be seen in Table 3.

	With Ages	**Without Ages**
All Diagnoses	Dataset 1	Dataset 2
Primary Diagnoses Only	Dataset 3	Dataset 4

Table 3: *Breakdown of the predictor investigation dataset factors*

We augmented all the datasets by including Biological Gender (sex assigned at birth) according to Member Life ID. The With Ages datasets were additionally augmented with an Age column (calculated by joining in Header Service From Date and Birth Date and computing the difference in years). Since not all Member Life IDs in our data had birthdates, the With Ages datasets had to be limited to rows where Birth Date could be joined in.

In a final pass of all the generated datasets, we removed all Member Life IDs with multiple Biological Genders and multiple ages. We believe these instances arose due to input error and decided to simply remove all entries of the affected Member Life IDs because we cannot distinguish which Biological Gender and Ages are the true value.

8.2.1. DATASET INVESTIGATION

Following the creation of the Datasets 1-4, we then used the Logistic Regression function we created on each of them to generate models that predict the probabilities of patients being diagnosed with C18, C50, E11, I10, I25, and N18. The performance of these models is summarized in Table 4 and Table 5 with the better performing model's cell highlighted in green.

Even though the Accuracy and AUC were the same for the No Age and With Age datasets, we ultimately decided to use Age in our training datasets because the With Ages dataset produced more sensitive models and the medical field prioritizes detecting True Positives over minimizing False Positives.

	Accuracy		Sensitivity		Area Under Curve	
Predicted Base Code	**No Age**	**With Age**	**No Age**	**With Age**	**No Age**	**With Age**
C18	0.917852	0.910155	0.76087	0.807692	0.943166	0.934748
C50	0.860449	0.857382	0.806931	0.888889	0.922667	0.947159
E11	0.806502	0.793738	0.790524	0.82239	0.870923	0.877168
I10	0.807403	0.794202	0.760961	0.799487	0.856636	0.881387
I25	0.840563	0.829694	0.845395	0.856707	0.923054	0.919049
N18	0.893959	0.875128	0.824742	0.846154	0.940761	0.938274

Table 4: *No Age vs. With Age Performance Metrics*

We observed markedly worse performance with the Primary Only dataset vs the All Diagnoses dataset on all metrics. As a result, we ruled out using only Primary Diagnoses in our training datasets.

	Accuracy		Sensitivity		Area Under Curve	
Predicted Base Code	**All Diagnoses**	**Primary Only**	**All Diagnoses**	**Primary Only**	**All Diagnoses**	**Primary Only**
C18	0.910155	0.858311	0.807692	0.545455	0.934748	0.83407
C50	0.857382	0.841029	0.888889	0.86747	0.947159	0.929354
E11	0.793738	0.7287	0.82239	0.815203	0.877168	0.839081
I10	0.794202	0.743287	0.799487	0.793821	0.881387	0.840792
I25	0.829694	0.798662	0.856707	0.84	0.919049	0.890352
N18	0.875128	0.859054	0.846154	0.817308	0.938274	0.920413

Table 5: *All Diagnoses vs. Primary Diagnoses (With Age) Performance Metrics*

After examining the efficacy of including Age as a predictor and of using only Primary Diagnoses from the Medical Claims Dataset, we decided to train our models on datasets generated while considering all diagnosis types and that include Age as a predictor variable.

8.3. DATA CLEANING/WRANGLING PROCESS FLOW DIAGRAM

A flowchart of the flow used to create the final "normal" training dataset can be found below in Figure 4.

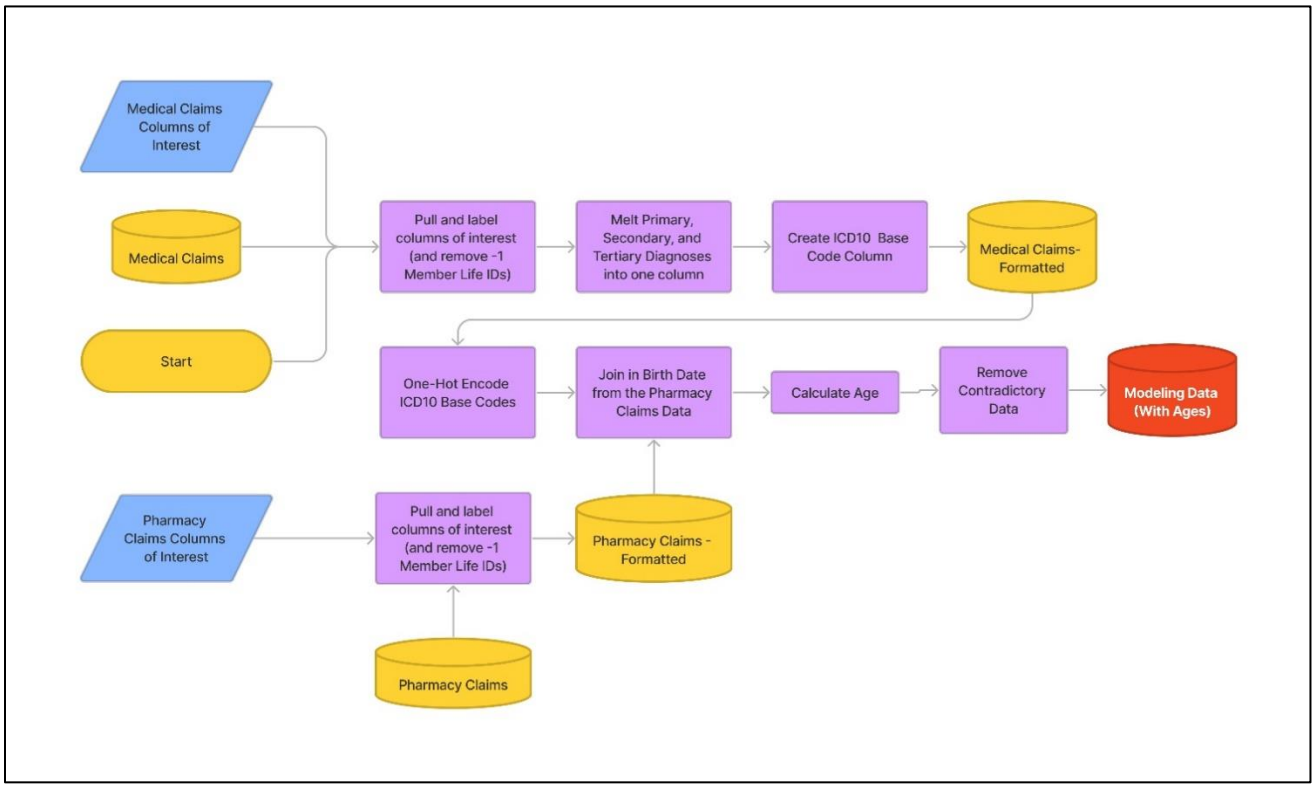

Fig 4: *Flowchart for creating final training data for models.*

8.4. TIMEBOUND ALTERNATIVE DATASETS

After seeing remarkably high accuracy and AUC (Area Under Curves) from our initial models, we became concerned that the model was making its predictions based off diagnoses caused by the target diagnoses rather than the target diagnoses being caused by the predictor diagnosis.

To combat this potential bias, we created an alternate modeling flow in our function where we removed ICD10 data which chronologically came on the same date or after the target ICD10 Base Code we want to predict from the Medical Claims Dataset. That modified dataset is then used to perform the one-hot encoding of predictor columns in lieu of the unaltered Medical Claims Dataset. The dataset then continues down the rest of the normal preprocessing pathway until we obtain a final training dataset. This altered dataset is then used to train the model of interest.

This method is a significant departure from the original method, because it requires that the Medical Claims Dataset be altered for each ICD10 Base Code the user wants to predict. They cannot simply use a completely preprocessed dataset like the originally planned method. It was noted that creating Timebound datasets that one-hot encode for all possible predictor ICD10

Base Codes is very time-intensive (~19 minutes to run each time). Therefore, we will provide Capgemini with alternate Timebound modeling functions which cut down on the computation time by only creating one-hot encoded columns for the top 20 predictor variables (according to our ICD10 Base Code selection function) during the one hot-encoding part of pre-processing. This is a departure from the original method which simply truncates a pre-generated one-hot encoded dataset which has columns for all ICD10 Base Codes.

However, for this report, we generated datasets with all the potential ICD10 Base Codes as columns for our model investigations (Figure 5). This was done so we were not limited to using the subset of ICD10 Base Codes identified by the variable auto-selection function in our modeling investigations. The resulting models trained on this data are not different from those that would be generated by the function described in the previous paragraph. We created Timebound data sets where C18, C50, E11, I10, I25, and N18 were presumed to be the predicted ICD10 Base Codes to train our Timebound investigative models.

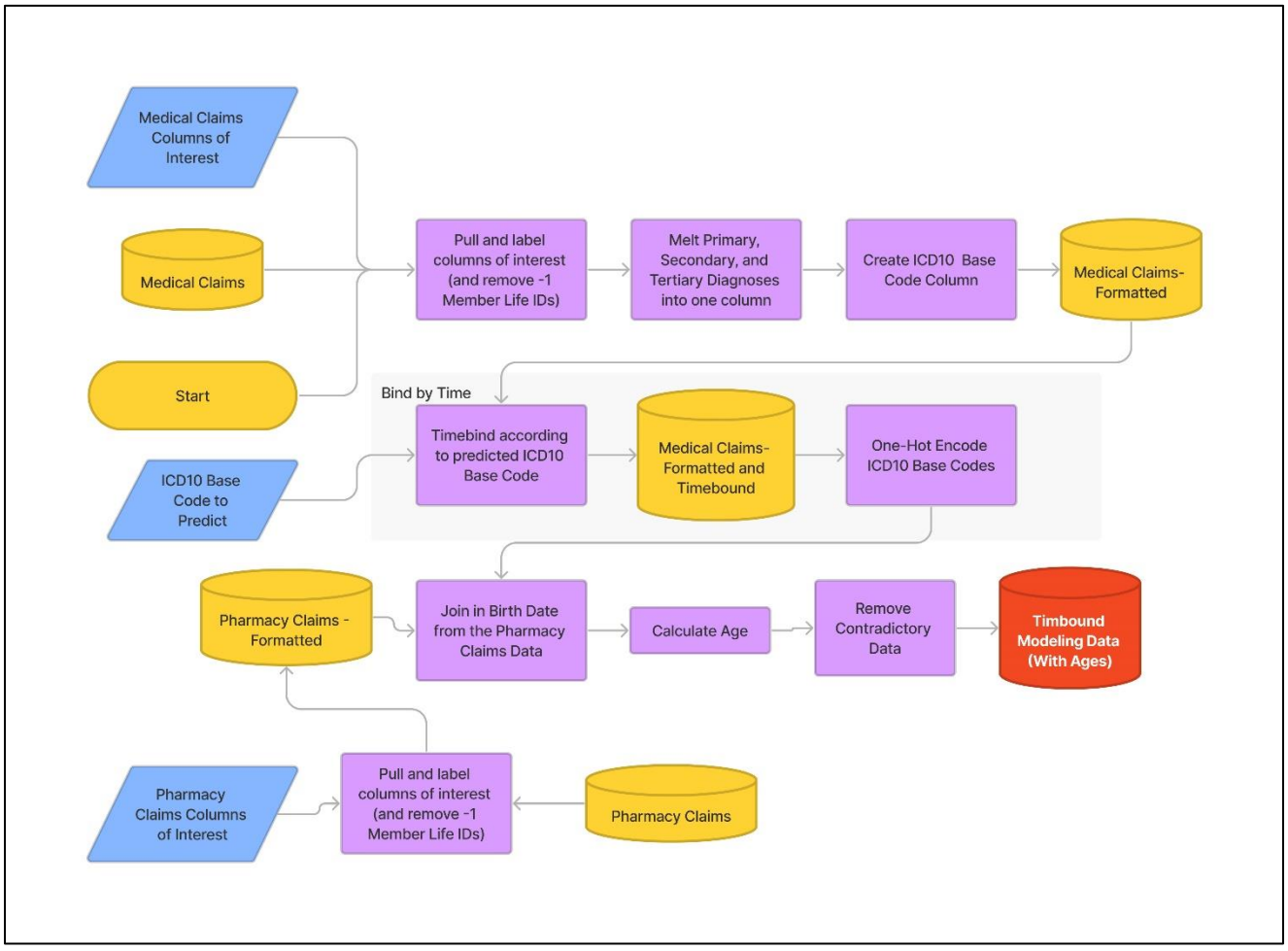

Fig 5: *Flowchart for creating final Timebound training data for models.*

9.MACHINE LEARNING MODELS

9.1. LOGISTIC REGRESSION

The Logistic Regression function has four required inputs:

- Input Dataset – a pandas data frame containing one-hot encoded ICD10 Base Code data with Biological Gender and (possibly Age) also included.
- Input Code – a text string indicating the desired ICD10 Base Code to predict.
- Edges – a pandas data frame containing information on the edges between nodes of the undirected graph generated for this model.
- Exclusion List - a list of ICD10 codes to ignore when selecting ICD10 Base Code predictor variables.

This function finds the top 20 ICD10 Base Codes with the highest edge weights from the Edges data frame where the Input Code is in the other node. These top 20 ICD10 Base Codes are then used to truncate the Input Dataset so it includes only the Input Code, top 20 ICD10 Base Codes, and Biological Gender and Age.

This trimmed down dataset is then used to generate a Cross-Validated Elastic Net Logistic Regression Model (from the Scikit-learn package) predicting Input Code and using the rest of the variables as predictors. The model is trained using balanced weighting. Cross-validation selects the model with the highest balanced accuracy. The function outputs this new model out along with the model accuracy, predictor ICD10 Base Codes used, and a dictionary holding the Training Dataset and a Test Dataset for further evaluating the model.

This function was then used to generate Logistic Regression models predicting the likelihood of patients developing C18, C50, E11, I10, I25, and N18 training on the non-Timebound and Timebound datasets. The Logistic Regressions trained during our investigation had their data divided into an 80/20 train/test split. Their accuracy, sensitivity, and area under curve were generated against this test set.

9.1.1. LOGISTIC REGRESSION RESULTS

The Logistic Regression results are presented in Table 6 and Table 7. The results are in line with what we would expect from a good predictive model. Reducing the decision threshold from 0.5 to 0.3 to increase the sensitivity is accompanied by a decrease in accuracy. However, at each threshold the accuracy is still reasonable, except when predicting E11 with a Timebound dataset at the 0.3 threshold (~60% accuracy).

Not surprisingly, the performance metrics of the Logistic Regression trained on Timebound data were a little lower than on the non-Timebound data. However, the models trained on the Timebound data still had relatively satisfactory performance in all three metrics (Accuracy, Sensitivity, and Area Under Curve).

	Threshold = 0.5		Threshold = 0.4		Threshold = 0.3		
Predicted Base Code	**Sensitivity**	**Accuracy**	**Sensitivity**	**Accuracy**	**Sensitivity**	**Accuracy**	**Area Under Curve**
C18	0.807692	0.910155	0.923077	0.873641	0.923077	0.808511	0.934748
C50	0.888889	0.857382	0.933333	0.820682	0.955556	0.780173	0.947159
E11	0.822390	0.793738	0.866523	0.756759	0.904198	0.700548	0.877168
I10	0.799487	0.794202	0.860624	0.756945	0.915348	0.698783	0.881387
I25	0.856707	0.829694	0.914634	0.785469	0.942073	0.731488	0.919049
N18	0.846154	0.875128	0.878205	0.830531	0.910256	0.777385	0.938274

Table 6: *Logistic Regression Performance Metrics (Not Timebound)*

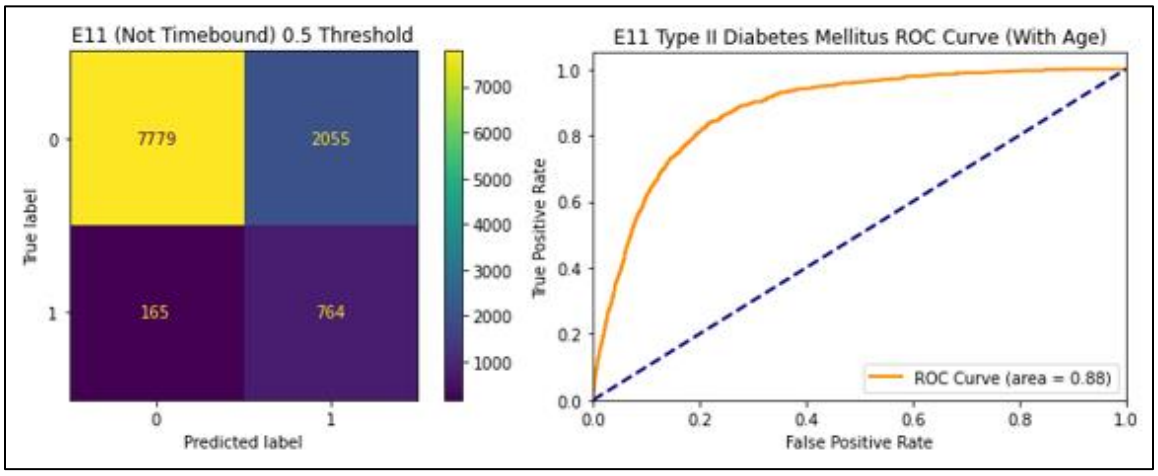

Fig 6: *E11 Confusion Matrix and ROC Curve at a 0.5 Threshold (Not Timebound)*

	Threshold = 0.5		Threshold = 0.4		Threshold = 0.3		
Predicted Base Code	**Sensitivity**	**Accuracy**	**Sensitivity**	**Accuracy**	**Sensitivity**	**Accuracy**	**Area Under Curve**
C18	0.611111	0.798476	0.777778	0.741243	0.777778	0.673325	0.790900
C50	0.834862	0.846976	0.87156	0.812413	0.917431	0.773762	0.929935
E11	0.781659	0.751579	0.855895	0.684132	0.902838	0.597362	0.847616
I10	0.792324	0.785137	0.859275	0.746307	0.909595	0.686298	0.869374
I25	0.814286	0.767072	0.877143	0.696739	0.917143	0.621109	0.872895
N18	0.721519	0.790393	0.816456	0.713184	0.911392	0.622038	0.845434

Table 7: *Logistic Regression Performance Metrics (Timebound)*

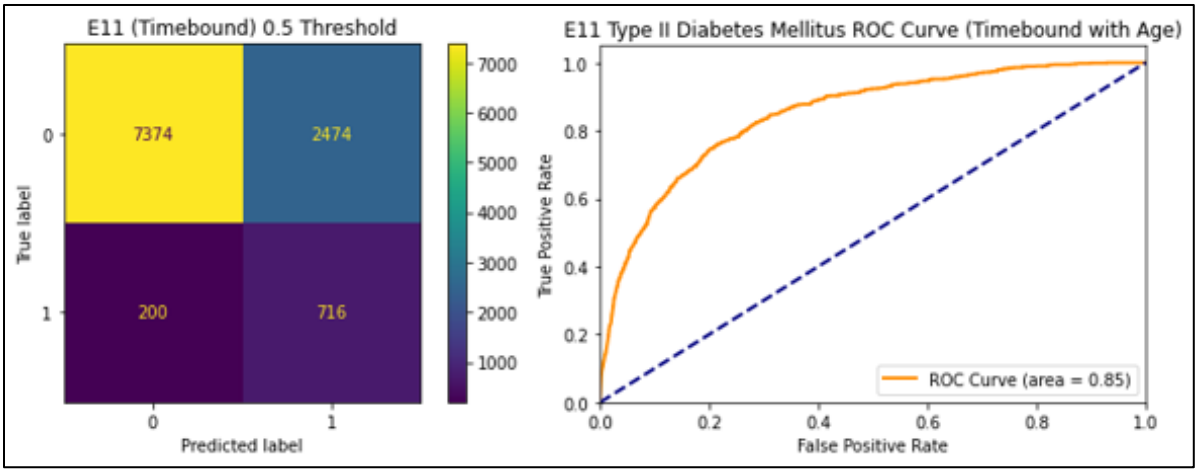

Fig 7: *E11 Confusion Matrix and ROC Curve at a 0.5 Threshold (Timebound)*

9.2. RANDOM FOREST CLASSIFIER

Approach

As in the Logistic Regression models, the predictors for the Random Forest Classifier are also selected automatically based on the strength of their association with the target ICD10 Base Code.

Inputs

- A dataset with one-hot encoded variables for all the predictor ICD10 Base Codes, biological gender, and age.
    - Filtered during process for top 20 predictor ICD10 Base Codes that contribute towards target risk condition.
- Undirected edges between different ICD10 Base Codes – Used to find the top 20 predictors based on weights
- List of ICD10 Base Codes to ignore from predictors - High association with target but unlikely to contribute to the target ICD10 Base Code.

Output

- Different Random Forest Classifiers to predict the target ICD10 Base Code given the other predictor ICD10 Base Codes

9.2.1. RF CLASSIFIER – ALL DATA WITHOUT TIMEBOUND

9.2.1.1. TUNING HYPERPARAMETERS:

Prior to the learning process, we optimized the parameters to help create the model with the most accurate predictions. So, we focused on adjusting and selecting the hyperparameters. There are many parameters to tune for better results, however in this model we concentrated on n_estimators (number of trees in the model) and max_depth (depth of each tree in the model).

- The hyperparameter_tuning function was created. It accepts the one-hot encoded dataset for all the ICD10 Base Codes, target ICD10 Base Code to predict, and list of predictor ICD10 Base Codes to use from the input dataset.
- A full dataset with Gender, Age and ICD10 Base Codes (without Timebound) is built. Then it is split into training and test dataset (80% training and 20% Test)
- Based on the GridSearchCV (Scikit-learn) and given the range of parameters to tune, the random Forest Classifier is trained for each combination and results in optimized hyperparameters.

The above steps were run for all the target ICD10 Base Codes and optimized hyperparameters were captured.

	n_estimators	max_depth
E11	8	300
C18	2	100
C50	8	100
I10	8	400
I25	8	300
N18	6	400

Table 8: *Hyperparameter RF Classifier – Without Timebound*

9.2.1.2. MODEL TRAINING:

As discussed above with the Logistic Regression Model, to generalize the solution to predict the likelihood of a patient developing the target risk conditions, the predictor variables are automatically selected based on the strength (weights) of the ICD10 Base Codes contributing towards the target ICD10 Base code.

1. Given the list of predictors (ICD10 Base Codes) to predict target IC10 Base Codes, filter out the predictors not needed for the experiment.
2. For the model evaluation, train the model with 80% of input dataset and test with 20% of the data.
3. Given the hyper-parameters and training dataset, run the Random Forest Classifier
4. We used 10-Fold cross validation to train the model for better prediction ability.
5. As far as the health risk conditions go, it is very important to call out patients prone to health risks early and so it is acceptable to have more false positives. So, in this model we have tried with different threshold levels (.3 to .5) to flag any patients as risky to target condition.

9.2.1.3. MODEL EVALUATION:

The Random Forest Model is evaluated based on the Accuracy, Sensitivity and AUC/ROC Score/Curve.

	Threshold = 0.5		Threshold = 0.4		Threshold = 0.3		
Predicted Base Code	**Sensitivity**	**Accuracy**	**Sensitivity**	**Accuracy**	**Sensitivity**	**Accuracy**	**Area Under Curve**
C18	0.69231	0.92465	0.92308	0.76577	1.00000	0.47645	0.93050
C50	0.90000	0.81483	0.90000	0.78798	0.94444	0.77516	0.86068
E11	0.83208	0.77506	0.87298	0.74124	0.93003	0.66979	0.86957
I10	0.78709	0.80191	0.90124	0.71978	0.94656	0.64815	0.87460
I25	0.83232	0.82431	0.90854	0.77859	0.93293	0.72489	0.90577
N18	0.83974	0.85840	0.91026	0.78733	0.93590	0.72712	0.93022

Table 9: *RF Classifier Performance Metrics (Not Timebound)*

For different thresholds and prediction of ICD10 Base Codes, the accuracy and sensitivity of the RF Non-Timebound model seems promising, however the key result is sensitivity.

For the observed sensitivity of the model, it is clear low thresholds for predicting risky ICD10 Base Codes yield better ratios; however, the model sensitivity goes down for the higher threshold values. Here a low threshold of .3 is suggested to predict patients prone to the disease accurately. Even though the accuracy score is low, the overall model is acceptable. For reference, the confusion matrix and AUC/ROC Curve of E11 are given below.

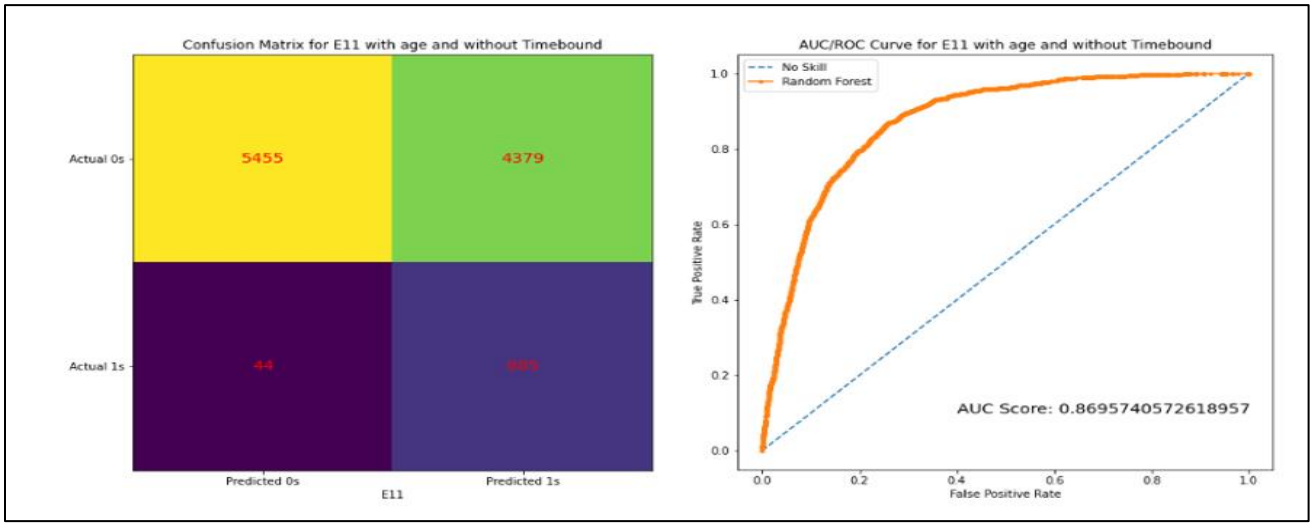

Fig 8: *E11 RF Classifier Model Confusion Matrix & AUC/ROC – With Age and Without Timebound (.5 Threshold)*

From what we observed from the above confusion matrix, AUC score and sensitivity, the model seems to be performing well.

9.2.2. RF CLASSIFIER – ALL DATA WITH TIMEBOUND:

9.2.2.1. TUNING HYPERPARAMETERS:

As we did with the non-Timebound dataset, we optimized our hyperparameters for the Timebound dataset and below are the results.

	n_estimators	max_depth
E11	2	100
C18	2	100
C50	8	300
I10	8	300
I25	8	100
N18	6	300

Table 10: *Hyperparameter RF Classifier – With Age and With Timebound*

9.2.2.2. MODEL EVALUATION:

The overall objective of running the model again with the Timebound datasets is to improve the model performance and below are the accuracy, sensitivity, and AUC results after running the model with the Timebound Dataset.

	Threshold = 0.5		Threshold = 0.4		Threshold = 0.3		
Predicted Base Code	**Sensitivity**	**Accuracy**	**Sensitivity**	**Accuracy**	**Sensitivity**	**Accuracy**	**Area Under Curve**
C18	0.72222	0.72229	0.83333	0.54446	1.00000	0.06271	0.86068
C50	0.76147	0.86175	0.82569	0.82031	0.89908	0.77869	0.92840
E11	0.80786	0.71683	0.96507	0.30639	1.00000	0.09662	0.84127
I10	0.80853	0.76730	0.87633	0.71667	0.93348	0.64933	0.86870
I25	0.77714	0.77757	0.85429	0.69990	0.90000	0.62650	0.86639
N18	0.70253	0.77553	0.83544	0.64768	0.93038	0.52550	0.84351

Table 11: *RF Classifier Performance Metrics (Timebound)*

Compared to the model without the Timebound dataset, model accuracy and sensitivity increased/remained same for most of the part. Overall, both the models provide adequate

evidence to use it to predict target ICD10 Base Codes. For reference, the confusion matrix and AUC/ROC Curve of E11 are given below.

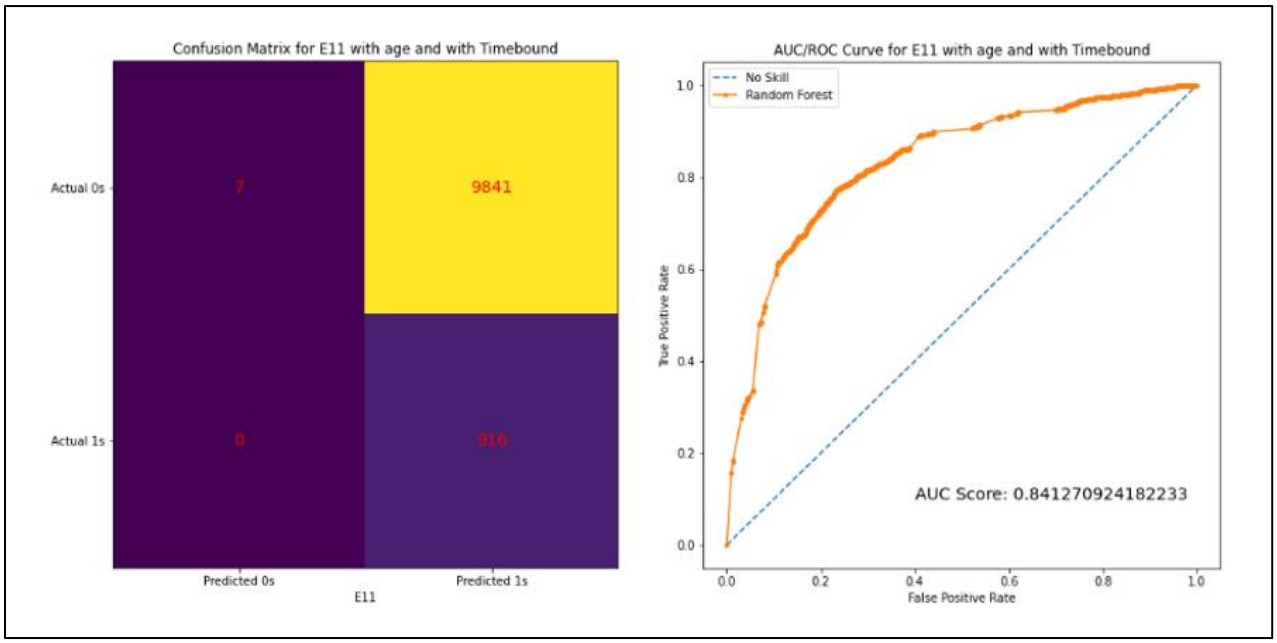

Fig 9: *E11 RF Classifier Model Confusion Matrix & AUC/ROC – With Timebound (.5 Threshold)*

9.3. NEURAL NETWORK MODEL

A neural network (NN) is a deep-learning framework that uses the concept of connected nodes (or artificial neurons) that are loosely based on the biological human brain.

Deep learning is an enhancement of the traditional machine learning algorithms wherein the learning representations from data are transformed through successive layers. The two main characteristics are that each layer in a deep model allows learning of increasingly complex representations, and all layers are learned jointly. We considered using deep learning because it has been seen that in practice applying the typical ML models repeatedly to emulate the effects of deep learning results in diminishing returns.

We implement our neural network model using TensorFlow. A tensor is a generalization of vectors and matrices to higher dimensions, a container for numerical data. TensorFlow uses Keras, an open-source library as its interface to implement the Neural Network.

The basic building blocks of a Neural Network are:

- Layers, which are a collection of nodes (neurons) combined into a network.
- Input data and target.
- Loss function, a feedback signal used for learning.
- Optimizer, which determines how the learning proceeds.

Our sequential model uses the one Input layer, the training data of predictors and target variable, two hidden Dense layers and one Output layer. Each layer uses a non-linear activation function *sigmoid* which is represented as:

$$g(z) = \frac{1}{1+e^{-z}} \text{ where } z = w^T x + b.$$

The *w* here is the vector of weights applied to each neuron within the layer. The loss function controls the output using a distance score from the target value and the adjustment to the weights is performed by the optimizer using **backpropagation** algorithm.

For all the NN classifiers below we have used the one-hot encoded datasets that were discussed above. In each model, we predict one of the six target ICD10 Base Codes identified above – C18, C50, E11, I10, I25, N18. For each model, using different thresholds, we output the Sensitivity (or Recall), the accuracy and the AUC. We used an 80/20 train-test split on all our datasets

An important point to consider here is that our datasets are imbalanced, the target ICD10 Base Codes have skewed distributions of classes. There are many more negative observations (0) than positive ones (1). ML algorithms are designed to improve accuracy and typically will not account for the proportion of classes. As a result, we calculated and applied class weights to our training data.

Here is an example of the difference in sensitivity between an unbalanced and balanced dataset for predicting the "E11" ICD10 category:

Unbalanced	0.2634
Balanced	0.7289

Table 12: *Example comparison of sensitivity between unbalanced and balanced dataset for E11*

Since we are building a binary classifier, the loss or cost function we used for our NN models is ***binary cross-entropy*** which is also called Log Loss and is represented as:

$$-\frac{1}{N}\sum_{i=1}^{N} y_i \ \log\big(p(y_i)\big) + (1-y_i)\log\big(1-p(y_i)\big)$$

where $p(y_i)$ is the probability of a positive diagnosis

The optimizer we used to generate our results is Adam which uses an adaptive learning rate for each parameter unlike the classical stochastic gradient descent algorithm which uses a single learning rate.

9.3.1. NN CLASSIFIER FOR USING ALL PREDICTORS

Unlike the Logistic Regression and Random Forest models, we considered each of the six target ICD10 Base Codes identified above and used all other ICD10 Base Codes as the predictors including Age and Gender. Here are the results obtained:

	Threshold = 0.5		Threshold = 0.4		Threshold = 0.3		
Predicted Base Code	**Sensitivity**	**Accuracy**	**Sensitivity**	**Accuracy**	**Sensitivity**	**Accuracy**	**Area Under Curve**
C18	0.2666	0.997	0.3	0.997	0.3	0.997	0.6743
C50	0.8241	0.972	0.8241	0.97	0.8461	0.966	0.9313
E11	0.7289	0.824	0.790	0.795	0.8261	0.762	0.8659
I10	0.8141	0.785	0.849	0.755	0.8817	0.722	0.8748
I25	0.717	0.865	0.733	0.846	0.7563	0.825	0.8645
N18	0.7562	0.923	0.756	0.912	0.7687	0.899	0.882

Table 13: *Neural Network Classifier Performance Metrics (all predictors)*

The model shows reasonable sensitivities and good AUC values for all conditions except C18. All metrics show improvement at lower thresholds.

Here is a sample confusion-matrix output for the prediction of ICD10 condition- Diabetes (E11)

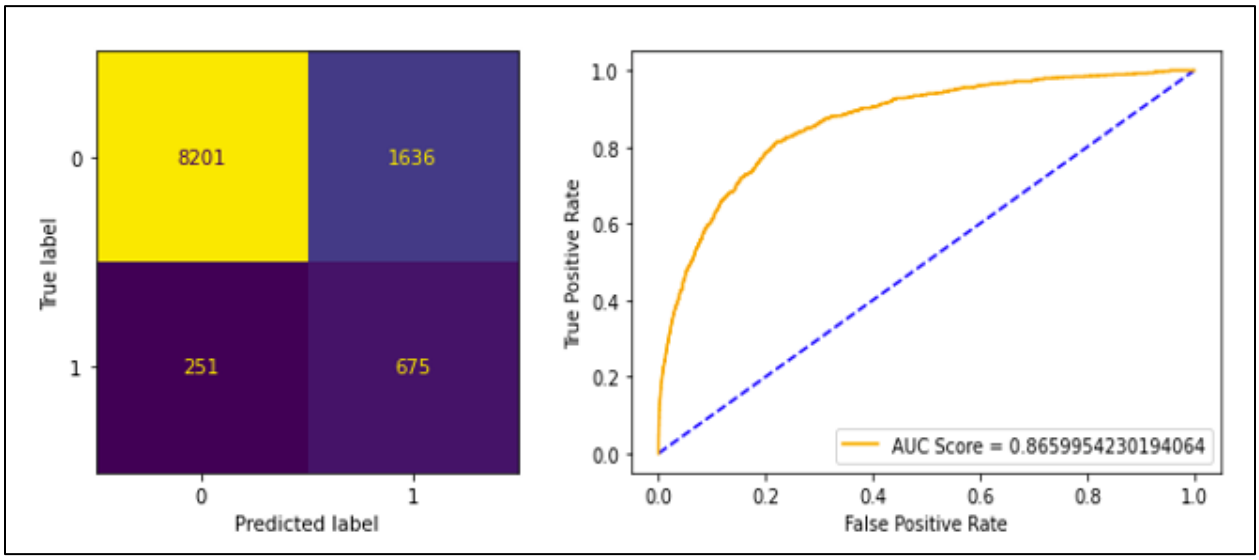

Fig 10: *E11 NN Classifier Model Confusion Matrix & AUC/ROC (All predictors)*

9.3.2. NN CLASSIFIER- AUTO SELECTED PREDICTORS

In this model we used a subset of predictor variables. They consist of the top 20 ICD10 Base Codes which were automatically selected based on the Edges data frame. Here are the results obtained:

	Threshold = 0.5		Threshold = 0.4		Threshold = 0.3		
Predicted Base Code	**Sensitivity**	**Accuracy**	**Sensitivity**	**Accuracy**	**Sensitivity**	**Accuracy**	**Area Under Curve**
C18	0.9	0.8723	0.9	0.8418	0.9666	0.7725	0.9425
C50	0.7252	0.8877	0.7582	0.8451	0.8131	0.7765	0.8671
E11	0.8552	0.7379	0.8844	0.7153	0.9038	0.6938	0.8646
I10	0.7524	0.7785	0.8107	0.7421	0.8549	0.6892	0.8291
I25	0.8963	0.7717	0.9355	0.706	0.9635	0.6599	0.9074
N18	0.825	0.861	0.8937	0.8051	0.95	0.7406	0.9353

Table 14: *Neural Network Classifier Performance Metrics (Auto-selected predictors)*

This model shows an improvement in sensitivity in 4 out of the 6 target ICD10 Base Codes as compared to the model using all predictors.

Here is a sample confusion-matrix output for the prediction of ICD10 condition- Diabetes (E11)

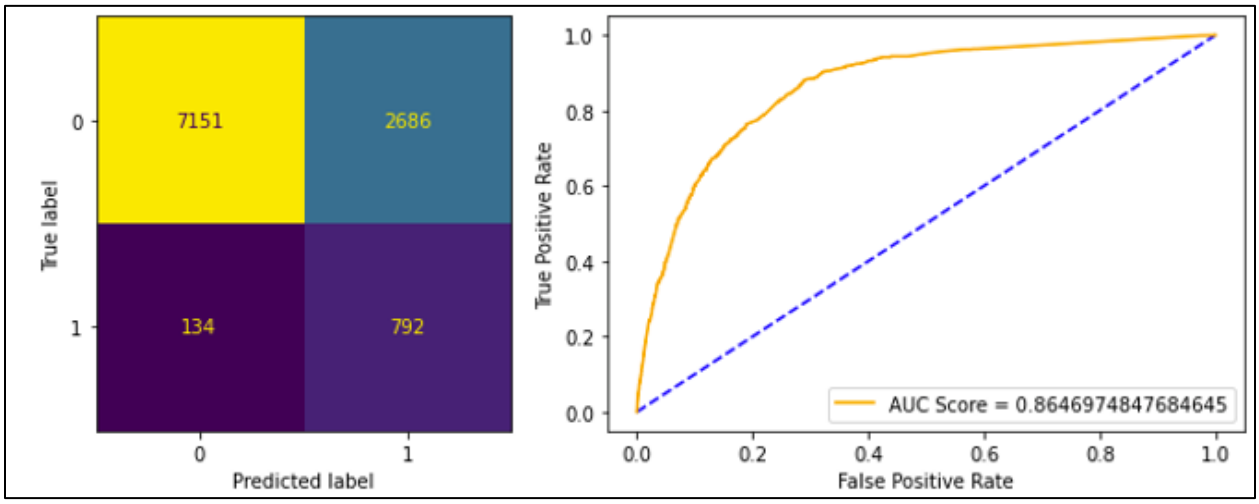

Fig 11: *E11 NN Classifier Model Confusion Matrix & AUC/ROC (Auto-selected predictors)*

9.3.3. NN CLASSIFIER USING TIMEBOUND DATA

In this model, we used all the ICD10 Base codes in the Timebound datasets. The Timebound datasets were created by removing potential predictor ICD10 Base Code that occur after or on the same date as the target ICD10 Base Code.

	Threshold = 0.5		Threshold = 0.4		Threshold = 0.3		
Predicted Base Code	**Sensitivity**	**Accuracy**	**Sensitivity**	**Accuracy**	**Sensitivity**	**Accuracy**	**Area Under Curve**
C18	0.5833	0.9721	0.5833	0.9692	0.5833	0.9655	0.8366
C50	0.632	0.946	0.6509	0.9415	0.6698	0.9331	0.8514
E11	0.8044	0.8671	0.8354	0.8355	0.8707	0.799	0.9113
I10	0.8356	0.8339	0.8733	0.8051	0.9065	0.7633	0.9141
I25	0.7238	0.8644	0.7555	0.8362	0.7809	0.8011	0.8729
N18	0.6329	0.9201	0.6518	0.9091	0.6835	0.8972	0.8612

Table 15: *Neural Network Classifier Performance Metrics (Timebound data)*

The sensitivity outputs from this dataset show improvements in only 2 out of the 6 target-based codes as compared to the model using all predictors.

Here is a sample confusion-matrix output for the prediction of ICD10 condition- Diabetes (E11)

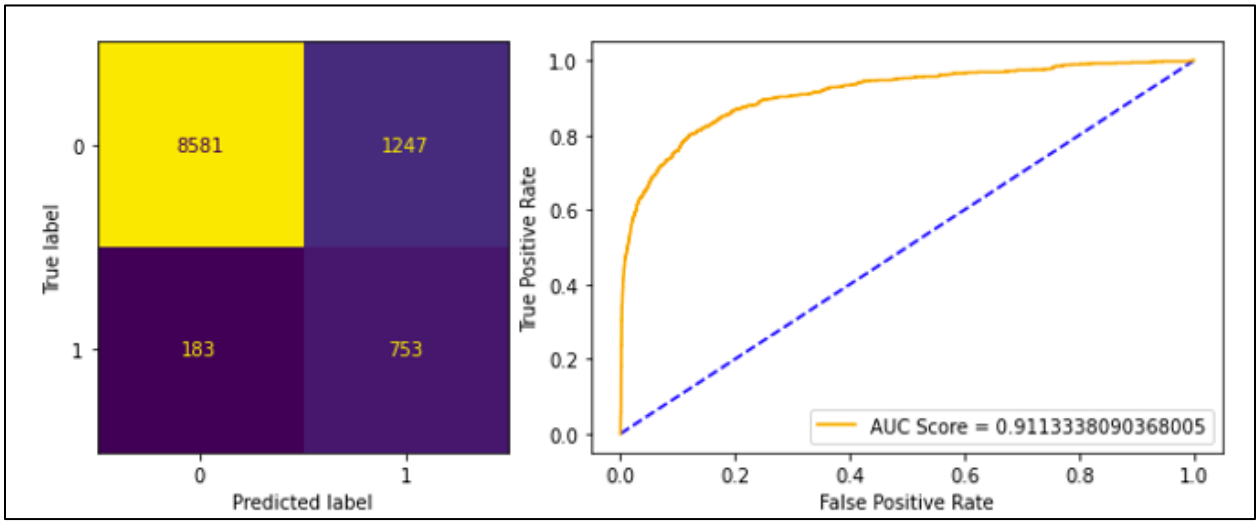

Fig 12: *E11 NN Classifier Model Confusion Matrix & AUC/ROC (Timebound data)*

There is no single optimal method for selecting the number of layers, and in general tuning the neural network requires modifying the number of layers and the neuron units to achieve the best possible outcomes. As such it has been observed that making the model more complex through layers or size will result in overfitting and diminishing returns.

Tuning the optimizer

A comparison of sensitivity outputs for two different optimizers is shown below:

		Optimizer	
Predicted Base Code	**Base Code Description**	**Adam**	**SGD**
C18	Malignant neoplasm of colon	0.267	0.967
C50	Malignant neoplasm of breast	0.824	0.615
E11	Diabetes	0.729	0.867
I10	Essential (primary) hypertension	0.814	0.645
I25	Chronic ischemic heart disease	0.717	0.745
N18	Chronic kidney disease	0.756	0.794

Table 16: *Optimizer Comparison*

From the above chart we can conclude there is no one size fits all solution when using the optimizers for a NN model, The Adaptive Learning optimizer (Adam) performs significantly better in 2 of the target ICD10 Base Codes than the SGD (fixed learning rate) classifier and significantly worse in 2 others. In conclusion, since we are considering binary classification here with each target ICD10 Base Code treated separately, optimal tuning of the NN model parameters may be required.

10. MODEL COMPARISON AND CONCLUSION:

All the models' performances were better when trained on the non-Timebound dataset. However, we believe that training on ICD10 Base Codes that occur after the actual target ICD10 Base Code may artificially bias and overfit the models. Therefore, we recommend that Capgemini train models using our Timebound method.

We additionally recommend that Capgemini perform a longitudinal study where they evaluate the non-Timebound and Timebound functions' performance when the ultimate status of the patients is not known at the outset. This will help definitively conclude whether the Timebound method should truly be used over the non-Timebound method.

Out of the three models we investigated (Logistic Regression, Random Forest, and Neural Net), none performed significantly better than the others. In general, their performance metrics were very close. Here is a walkthrough of our models' performance when predicting E11:

- Logistic Regression performed best for E11 Target ICD10 Base Codes in Accuracy and AUC.
- Random Forest performed best for E11 Target ICD10 Base Codes with non-Timebound and Timebound datasets in terms of Sensitivity at the .3 threshold.
- Neural Net performed best for E11 Target ICD10 Base Codes for non-Timebound and Timebound datasets in terms of Sensitivity at the .4 & .5 thresholds.

Tables comparing the performances of our models can be found in **Appendix A.**

Therefore, we will provide Capgemini with functions that automate the creation of all three of these models along with their respective Timebound versions. However, we also note that Logistic Regression is the simplest and easiest to interpret since the coefficients of the model can be used as representations of how much a particular variable affects the end prediction.

We believe that this final solution is broadly applicable for creating models to predict patients' probabilities of developing other target ICD10 Base Codes not tested in this report.

11. FUTURE WORK

As a future work, this project can be enhanced to be more generalized based on the criteria of threshold limit, Sensitivity/Accuracy/AUC and automated to choose the model that best fits and predicts the target ICD10 Base Code.

The process for identifying ICD10 Base Code exclusions is manual. We recommend that future work include the development of an automated method for finding ICD10 Base Codes that should not be considered for inclusion in the models.

We also recommend augmenting patient data with social determinants of health to train models in order to better generalize their overall performance.

REFERENCES

- Centers of Disease Control and Prevention. (2015, November 6) - Causes of Death https://www.cdc.gov/nchs/nvss/mortality/comparability_icd.htm
- guest_blog. (2021, January 6). *Class imbalance: Handling imbalanced data using Python*. Analytics Vidhya. Retrieved November 30, 2022, from https://www.analyticsvidhya.com/blog/2017/03/imbalanced-data-classification/
- Singh, H. (2021, March 12). *Neural network: Introduction to neural network: Neural network for DL*. Analytics Vidhya. Retrieved November 30, 2022, from https://www.analyticsvidhya.com/blog/2021/03/basics-of-neural-network/
- World Health Organization. (2016). *Preventing disease through healthy environments*. Geneva, Switzerland. Retrieved November 18, 2022, from https://www.who.int/publications/i/item/9789241565196.

Appendix A – Model Comparison Tables

AUC Comparisons

Predicted Base Code	Logistic Regression	Random Forest	Neural Net	Logistic Regression (Timebound)	Random Forest (Timebound)	Neural Net (Timebound)
C18	0.93475	0.9305	0.9425	0.7909	0.86068	0.8366
C50	0.94716	0.86068	0.8671	0.92994	0.9284	0.8514
E11	0.87717	0.86957	0.8646	0.84762	0.84127	0.9113
I10	0.88139	0.8746	0.8291	0.86937	0.8687	0.9141
I25	0.91905	0.90577	0.9074	0.8729	0.86639	0.8729
N18	0.93827	0.93022	0.9353	0.84543	0.84351	0.8612

Table 17: *Area Under Curve of Models*

Accuracy Comparisons

Predicted Base Code	Logistic Regression	Random Forest	Neural Net	Logistic Regression (Timebound)	Random Forest (Timebound)	Neural Net (Timebound)
C18	0.91016	0.92465	0.8723	0.79848	0.72229	0.9721
C50	0.85738	0.81483	0.8877	0.84698	0.86175	0.946
E11	0.79374	0.77506	0.7379	0.75158	0.71683	0.8671
I10	0.7942	0.80191	0.7785	0.78514	0.7673	0.8339
I25	0.82969	0.82431	0.7717	0.76707	0.77757	0.8644
N18	0.87513	0.8584	0.861	0.79039	0.77553	0.9201

Table 18: *Accuracy of Models at 50% Threshold*

Predicted Base Code	Logistic Regression	Random Forest	Neural Net	Logistic Regression (Timebound)	Random Forest (Timebound)	Neural Net (Timebound)
C18	0.87364	0.76577	0.8418	0.74124	0.54446	0.9692
C50	0.82068	0.78798	0.8451	0.81241	0.82031	0.9415
E11	0.75676	0.74124	0.7153	0.68413	0.30639	0.8355
I10	0.75695	0.71978	0.7421	0.74631	0.71667	0.8051
I25	0.78547	0.77859	0.706	0.69674	0.6999	0.8362
N18	0.83053	0.78733	0.8051	0.71318	0.64768	0.9091

Table 19: *Accuracy of Models at 40% Threshold*

Predicted Base Code	Logistic Regression	Random Forest	Neural Net	Logistic Regression (Timebound)	Random Forest (Timebound)	Neural Net (Timebound)
C18	0.80851	0.47645	0.7725	0.67333	0.06271	0.9655

C50	0.78017	0.77516	0.7765	0.77376	0.77869	0.9331
E11	0.70055	0.66979	0.6938	0.59736	0.09662	0.799
I10	0.69878	0.64815	0.6892	0.6863	0.64933	0.7633
I25	0.73149	0.72489	0.6599	0.62111	0.6265	0.8011
N18	0.77739	0.72712	0.7406	0.62204	0.5255	0.8972

Table 20: *Accuracy of Models at 30% Threshold*

Sensitivity Comparisons

Predicted Base Code	Logistic Regression	Random Forest	Neural Net	Logistic Regression (Timebound)	Random Forest (Timebound)	Neural Net (Timebound)
C18	0.80769	0.69231	0.9	0.61111	0.72222	0.5833
C50	0.88889	0.9	0.7252	0.83486	0.76147	0.632
E11	0.82239	0.83208	0.8552	0.78166	0.80786	0.8044
I10	0.79949	0.78709	0.7524	0.79232	0.80853	0.8356
I25	0.85671	0.83232	0.8963	0.81429	0.77714	0.7238
N18	0.84615	0.83974	0.825	0.72152	0.70253	0.6329

Table 21: *Sensitivity of Models at 50% Threshold*

Predicted Base Code	Logistic Regression	Random Forest	Neural Net	Logistic Regression (Timebound)	Random Forest (Timebound)	Neural Net (Timebound)
C18	0.92308	0.92308	0.9	0.77778	0.83333	0.5833
C50	0.93333	0.9	0.7582	0.87156	0.82569	0.6509
E11	0.86652	0.87298	0.8844	0.8559	0.96507	0.8354
I10	0.86062	0.90124	0.8107	0.85928	0.87633	0.8733
I25	0.91463	0.90854	0.9355	0.87714	0.85429	0.7555
N18	0.87821	0.91026	0.8937	0.81646	0.83544	0.6518

Table 22: *Sensitivity of Models at 40% Threshold*

Predicted Base Code	Logistic Regression	Random Forest	Neural Net	Logistic Regression (Timebound)	Random Forest (Timebound)	Neural Net (Timebound)
C18	0.92308	1	0.9666	0.77778	1	0.5833
C50	0.95556	0.94444	0.8131	0.91743	0.89908	0.6698
E11	0.9042	0.93003	0.9038	0.90284	1	0.8707
I10	0.91535	0.94656	0.8549	0.9096	0.93348	0.9065
I25	0.94207	0.93293	0.9635	0.91714	0.9	0.7809
N18	0.91026	0.9359	0.95	0.91139	0.93038	0.6835

Table 23: *Sensitivity of Models at 50% Threshold*