__Classifying Risk Mortality for Patients above 65 years of age__

By:

Arpita Deshmukh

Irma Yanez Soto

Mrinalini Dey

Email: arpitapd@tamu.edu

ISTM 650 – 601 (Group 4)

Healthcare Analytics Using Texas Hospital Inpatient Discharge Data

Table of Contents

Abstract

Ensuring health care quality has always been of paramount importance for the healthcare community. The objective of our project it to classify the risk mortality of a patient over the age of 65. Classifying the patient's risk mortality could prove beneficial in optimizing the efforts to improve the current quality of healthcare delivered to patients. Our report considers Patient Age, Admitting Diagnosis, Principal Diagnosis, Type of Admission and Source of Admission to classify the patient's risk of mortality as either minor, moderate, major, or extreme. Since a majority of our predictor variables are categorical variables we transformed, cleaned and filtered the data to fit our data mining problem. Our findings suggest that the variables chosen as predictors are relevant, however we must also note that predicting risk mortality of patients might implement other factors not contained in the data set. Overall, based on our analysis of the THCIC data set, the variables used are appropriate indicators of a patient's risk of mortality.

Introduction

Survey of Healthcare analytics area

Since around the 1920s, healthcare professionals realized that documenting patient records not only benefits patients but everyone else involved, such as the medical provider, the payer and the hospital. The data and the process of documentation have evolved since then, becoming much more sophisticated, detailed and standardized. Analytics and healthcare data go hand-in-hand in this age of information technology. Even though it's complex, healthcare provides a plethora of data, which can be used to draw several insights for business problems such as improving the quality of care provided, optimizing the costs for the hospital administration or improving the utilization of hospital beds.

Interactions between the different players - Hospitals, Insurance companies (payers), Patients, Doctors and Nurses in the healthcare domain provide huge volumes of data for analytics and business intelligence. This data can be harnessed and used in an effective way to make important operational decisions to generate better revenues, increase efficiency of the hospital or that of the payer, provide faster and better medical aid when required, isolate the most painful points in the healthcare system and try to get rid of them and possibly predict future trends in the health of the patient. This data is analyzed and studied, using a host of different algorithms to determine the trends and patterns that are most prevalent, and in turn the healthcare industry can be improved all across the world.

Data Mining Problem

Goal: Classify the risk mortality of a patient

The objective here is to classify the risk mortality category of the patient. Risk Mortality denotes the possibility of dying. The categories are Minor, Moderate, Major, and Extreme. To answer this data mining question, we're using information from the following variables that were available in the THCIC data - Patient Age, Admitting Diagnosis, Principal Diagnosis, Type of Admission and Source of Admission. Additionally, we have created two derived predictor variables that cover the Number of Diagnosis and whether the codes for Admitting Diagnosis and Principal Diagnosis match

Why this problem?

We decided to choose this problem since Risk Mortality is quite an important attribute that gives us information about a patient's health and indicates if an increased level of care delivery is required. Looking at the data, we decided to use Patient Age, Principal Diagnosis, Type of Admission, Source of Admission, and Admitting Diagnosis as attributes that could help in predicting the Risk of Mortality.

First, to reduce our dataset, we are focusing on the aging population. Therefore, we use Patient Age to filter those who are above 60 years of age. We are using Admitting diagnosis and Principal diagnosis to determine the relationship between different diagnoses and Risk Mortality. As previously defined, Admitting Diagnosis and Principal diagnosis might not be the same for one individual patient. In other words, there might be a difference between the reason they arrived at the hospital and what the doctor diagnosed them for. Therefore, it is important to note those differences, if they exist. As mentioned below, there exists some correlation between the type of admission, source of admission and risk of mortality. Source and Type of Admission can be important factors that determine the risk of mortality of a patient. For instance, a patient brought in through the ER might be in a more serious condition that a patient that came in for a minor injury. Overall, we chose these variables to reduce our dataset and potentially obtain a more efficient and accurate model.

Literature review

The data involved in the analysis is from the Texas Department of State Health Services Center for Health Statistics; Public Use Data File (PUDF). The PUDF file contains the patient-level information for inpatient hospital stays and the data is extracted from DSHS's Hospital Discharge Database (HDD).

The data manual is present *here* which would help in better understanding of the data (We're using the manual for 2013)

Focus Variables

The classification problem would be analyzed based on Age, principal diagnosis, type of admission and source of admission. The following are the reasons for selecting the mentioned variables.

Effect of Age on Risk Mortality

A patient's age has some crucial implications for his/her risk of death [1]. Thus, this variable has an important application in health care systems. Moreover, aging is one of the major risk factors for most chronic diseases that include neurological diseases [1]

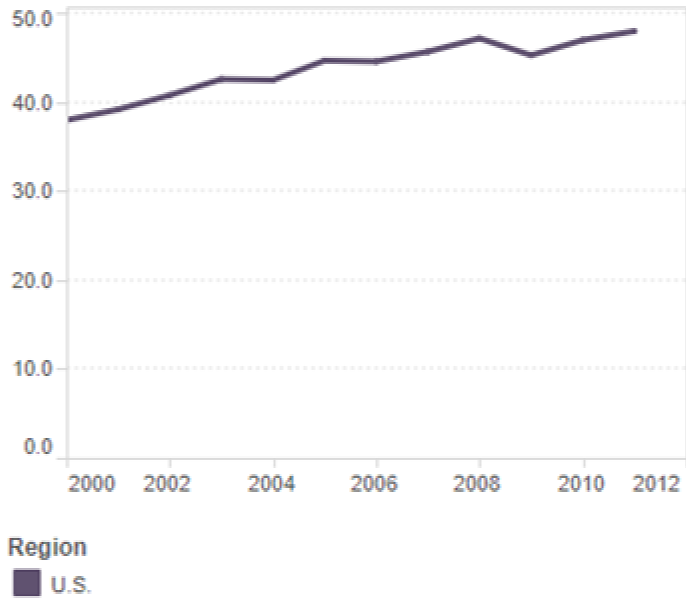
Effect of Principal Diagnosis on Risk Mortality: Neurological, Respiratory, Digestive

One of the variables we will use for our study is the principal diagnosis. The principal diagnosis is defined as the condition, after study, that occasioned admission to the hospital. Principal diagnosis, however, is not always the primary diagnosis. It is rather the reason for hospital admission requiring at least one overnight stay (inpatient).

For the purpose of our study, we will focus on principal diagnosis relating to neurological, respiratory, and digestive diseases (ICD-9-CM 320-389, 504-519, 520-538). We have chosen to focus on these principal diagnoses because they appear to have a stronger correlation with a mortality rate of elders than other diagnosis.

According to the World Health Organization (WHO), neurological diseases constitute around 12% of deaths globally [3].

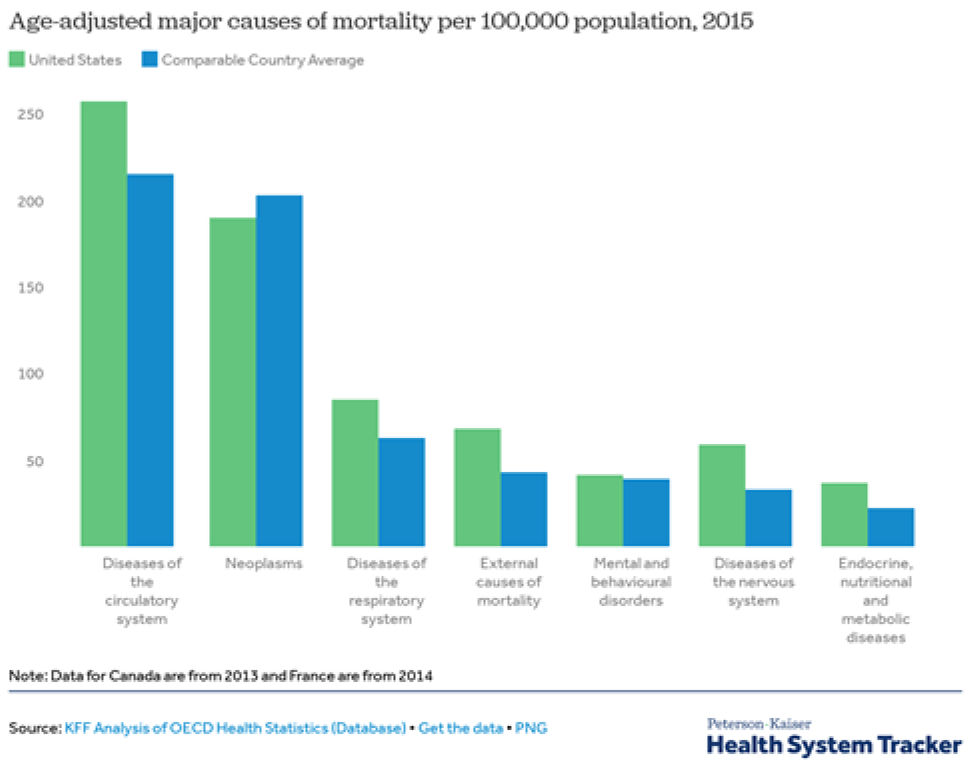
Mortality Due to Nervous System [4]



In the U.S., the mortality rate due to nervous system diseases has a high rate of almost 50%, ranking above other countries. Most of these deaths can be attributed to Alzheimer's disease,

which primarily affects the elderly population [2]. Note, the data above has been age-standardized, as nervous system diseases tend to mostly affect the elderly population.

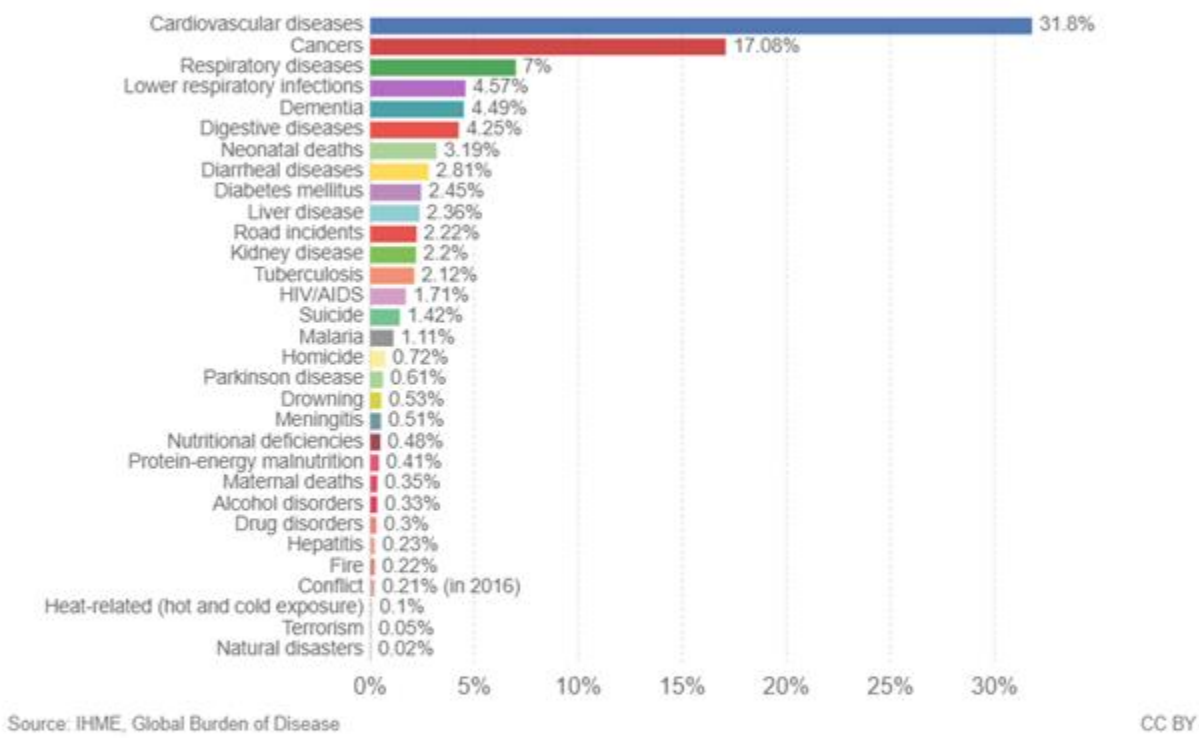
Major Causes of Mortality in 2015 [8]



Respiratory system diseases can also lead to deaths in the aging population. As we can see in the graph above, diseases of the respiratory system were the third leading cause of mortality in the U.S. during 2015. The elderly population has a higher risk of developing pneumonia, which is one of the largest causes of respiratory system-related deaths. According to WHO, pneumonia is also one of the most frequent reasons for hospitalization [4].

According to the Canadian Journal of Gastroenterology and Hepatology, as people age their gastrointestinal system (GIS) changes. The aging population might see effects in motility, enzyme and hormone secretion, digestion, and absorption [5]. Additionally, ulcers are on average the leading cause of mortality associated with digestive system diseases. Ulcers are usually manifested as people age, therefore making them a risk factor for the aging population.

Major Causes of Death 2017 [7]



The rate of digestive disease-related mortality has been increasing over time. As we can see in the graph above, digestive diseases accounted for around 4% of deaths globally in 2017.

Overall, we can see there exists a strong association between the chosen principal diagnosis and the risk of mortality, specifically for the aging population. Therefore, it is appropriate to use such variables when conducting our study.

Problem Formulation

Data Mining Plan

This plan includes the step by step procedure to follow in selecting a final data mining model:

1. Understand the purpose of solving business problem and convert it into a data mining problem
2. Obtain the relevant data
3. Explore and clean the data

4. Implement multiple data mining tasks
5. Evaluate and compare the models
6. Select the final model
7. Measure the performance of the final model

Data Preprocessing

Data

The data involved in the analysis is from the Texas Department of State Health Services Center for Health Statistics; Public Use Data File (PUDF). The data required to address the business data mining problem contains 38018 rows, 4 predictor variables, 1 response variable, and a unique identifier, Record_ID.

Variables related to the business data mining problem

The classification problem would be analyzed based on Age, principal diagnosis, type of admission and source of admission

Predictor Variables: Pat_Age, Princ_Diag_Code, Type_of_Admission, Source_of_Admission

Response Variable: Risk_Mortality

Files included:

1. Base Data #1 File - This file contains the data elements like patient information and details including diagnosis information, surgery information and so on. The files contain Record ID as a primary key.

2. Base Data #2 File - This file provides information about some calculated fields and charges related to situational data.

3. Charges File - The Charges File includes a description of the process such as lab work and the associated charges with that procedure.

4. Facility Type Data - This file gives information about the different facilities (hospitals) that are included in this data, and what amenities are there in each facility.

For our business problem & data mining problem, the analysis is entirely based on the information in the Base data #1 file. Since our problems are related to health care in the elderly, we will use data that is relevant to our age groups, that is above 65 years of age. Further, we are considering a specific range of ICD-9-CM codes from 320-389 which fall under the category of neurological and sense organ diseases. The codes can be accessed *here*

These steps would include the following: (will be explained in the upcoming topics)

1. Data Extraction - Data Extraction involves getting the appropriate data from the source data.
2. Data Cleaning - The extracted data from step 1 needs to be cleaned, such as removing null values, replacing garbage values, etc. This step encompasses these processes.
3. Data Transformation - Finally, the cleaned and relevant data needs to be formatted in a way such that meaningful information and some trends and insights can be derived from it.

Implementation

We have created three models: Naive Bayes, Decision Trees (with different tree depths), and Logistic Regression

Initial Steps

Before implementing the model, we performed all the steps which would be essential for all the model. Those are as follows:

1. Create a balanced sample

We are only dealing with the data that is relevant to our question from the THCIC data. Using SQLServer, we filtered the data for age groups 15 to 21 and 24-26 (PAT_AGE) and the principal diagnosis range (ICD-9-CM 320-389, 504-519, 520-538,038) which encompass bacterial diseases, neurological diseases and diseases of the circulatory, respiratory and digestive system, which gave us about ~31,000 rows to work with. Further, a diverse range of diagnosis codes provided us a balanced sample of Risk Mortality. We were able to get a good mix of all the different values for Risk Mortality, as shown below.

SELECT count(*) as count_of_records, risk_mortality

FROM [ISTM650_IAM].[dbo].['Sheet1 (2)$']

GROUP BY risk_mortality;

	count_of_records	risk_mortality
1	12729	3
2	10149	4
3	4201	1
4	8571	2

2. Partition the sample into training, test, and validation

Partitioning the data into Training, Validation and Test data is an intrinsic step while using Supervised algorithms in Data Mining and a continuation of the previous step of creating a balanced sample.

Training Partition: The data for this partition should be the largest since the models are learning using this data partition. The models use this data to learn how the data is being classified.

Validation Partition: This data partition allows the model to check and assess how it is performing. We use it to compare the performance of different models and then pick the one that works best for our data. After the model is trained with the training partition data, it is important to see how the model performs with known data. This partition enables us to check that.

Test Partition: When we need to finally assess the performance of the model that we have chosen, we use this data partition. It is unseen data and really puts our model to the test. It also helps to avoid and overcome the problem of overfitting.

Within our data, we used a 60-30-10 partition for the training-validation-test. Below, we have included a more detailed summary of our partitioned data. It provides the number of observations in each partition.

Data Set Allocations	
Training	60.0
Validation	30.0
Test	10.0

The summary of the Variables is given below. This summary of data was obtained from SAS Enterprise Miner

Variable Summary

Role	Measurement Level	Frequency Count
ID	NOMINAL	1
INPUT	BINARY	1
INPUT	INTERVAL	1
INPUT	NOMINAL	5
TARGET	NOMINAL	1

Partition Summary

Type	Data Set	Number of Observations
DATA	EMWS1.FIMPORT_train	35650
TRAIN	EMWS1.Part_TRAIN	21389
VALIDATE	EMWS1.Part_VALIDATE	10692
TEST	EMWS1.Part_TEST	3569

Summary Statistics for Class Targets

Data=DATA

Variable	Numeric Value	Formatted Value	Frequency Count	Percent	Label
RISK_MORTALITY	1	1	4201	11.7840	RISK_MORTALITY
RISK_MORTALITY	2	2	8571	24.0421	RISK_MORTALITY
RISK_MORTALITY	3	3	12729	35.7055	RISK_MORTALITY
RISK_MORTALITY	4	4	10149	28.4684	RISK_MORTALITY

Data=TEST

Variable	Numeric Value	Formatted Value	Frequency Count	Percent	Label
RISK_MORTALITY	1	1	421	11.7960	RISK_MORTALITY
RISK_MORTALITY	2	2	858	24.0403	RISK_MORTALITY
RISK_MORTALITY	3	3	1274	35.6963	RISK_MORTALITY
RISK_MORTALITY	4	4	1016	28.4674	RISK_MORTALITY

Data=TRAIN

Variable	Numeric Value	Formatted Value	Frequency Count	Percent	Label
RISK_MORTALITY	1	1	2521	11.7864	RISK_MORTALITY
RISK_MORTALITY	2	2	5143	24.0451	RISK_MORTALITY
RISK_MORTALITY	3	3	7636	35.7006	RISK_MORTALITY
RISK_MORTALITY	4	4	6089	28.4679	RISK_MORTALITY

Data=VALIDATE

Variable	Numeric Value	Formatted Value	Frequency Count	Percent	Label
RISK_MORTALITY	1	1	1259	11.7752	RISK_MORTALITY
RISK_MORTALITY	2	2	2570	24.0367	RISK_MORTALITY
RISK_MORTALITY	3	3	3819	35.7183	RISK_MORTALITY
RISK_MORTALITY	4	4	3044	28.4699	RISK_MORTALITY

3. Data Preprocessing

Data Extraction

Selecting only the relevant columns from the data was done in SQL Server Management Studio.

The where clause in the SQL statement included the range of diagnosis codes and the age group we are focusing on in this project. The SQL statement is given below

SELECT [RECORD_ID],[TYPE_OF_ADMISSION],[SOURCE_OF_ADMISSION],[PAT_AGE],[ADMITTING_DIAGNOSIS],[PRINC_DIAG_CODE] ,[OTH_DIAG_CODE_1] ,[OTH_DIAG_CODE_2] ,[OTH_DIAG_CODE_3], [OTH_DIAG_CODE_4] ,[OTH_DIAG_CODE_5] ,[OTH_DIAG_CODE_6] ,[OTH_DIAG_CODE_7] , [OTH_DIAG_CODE_8] ,[OTH_DIAG_CODE_9] ,[OTH_DIAG_CODE_10] ,[OTH_DIAG_CODE_11] ,[OTH_DIAG_CODE_12] ,[OTH_DIAG_CODE_13],[OTH_DIAG_CODE_14] ,[OTH_DIAG_CODE_15] ,[OTH_DIAG_CODE_16] ,[OTH_DIAG_CODE_17] ,[OTH_DIAG_CODE_18] ,[OTH_DIAG_CODE_19] ,[OTH_DIAG_CODE_20] ,[OTH_DIAG_CODE_21] ,[OTH_DIAG_CODE_22] ,[OTH_DIAG_CODE_23] ,[OTH_DIAG_CODE_24] ,[RISK_MORTALITY]

FROM [ISTM650_IAM].[dbo].[base1]

WHERE (Pat_age='15' or PAT_AGE='16' or PAT_AGE='17' or PAT_AGE='18' or PAT_AGE='19' or PAT_AGE='20' or PAT_AGE='21' or pat_age = '24' or pat_age = '25' or pat_age='26') and (PRINC_DIAG_CODE LIKE '32%' OR PRINC_DIAG_CODE LIKE '33%' OR PRINC_DIAG_CODE LIKE '34%' OR PRINC_DIAG_CODE LIKE '35%' OR PRINC_DIAG_CODE LIKE '36%' OR PRINC_DIAG_CODE LIKE '37%' OR PRINC_DIAG_CODE LIKE '38%' OR PRINC_DIAG_CODE LIKE '50%' OR PRINC_DIAG_CODE LIKE '51%' OR PRINC_DIAG_CODE LIKE '52%' OR PRINC_DIAG_CODE LIKE '53%' or PRINC_DIAG_CODE like '03%');

Data Cleaning

The categories of diagnosis include diseases related to Neurological, Respiratory and Digestive System. Considering our focus variables, we have done the following operations in order to have the data in the desired format.

1. Source of Admission

    a. Remove Null/Missing Values - deleted the rows which contained blanks using MS Excel

    b. Converting character variable to an integer value - the rows containing 1 was converted into 10. This transformation was done to ensure that the categories are consistent. Moreover, D was converted to category 11. These operations were done by using the replace function in MS Excel

2. Type of Admission

    a. Remove Null/Missing values - deleted the data which contained blanks using MS Excel

Data Variables

All the data variables in the data are categorical. There was no need for any transformation since Naive Bayes works on categorical variables. The categorical variables are nominal in our data which means that they are just categories and do not have any order. On the other hand, the target variable i.e. risk mortality variable is ordinal because the variable is ordered. The values that are present in risk mortality are 1,2,3,4 with 1 representing minor risk and 4 representing extreme risk of mortality. We did not convert the age which is a categorical variable to a numerical variable because Naive Bayes supports categorical data and build models based on that

Data Transformation

In this step we focused on finding more relevant data from the existing data which was achieved by creating derived variables. These derived variables were useful in predicting the class of a patient as well

Derived Variables

We created the following derived variables:

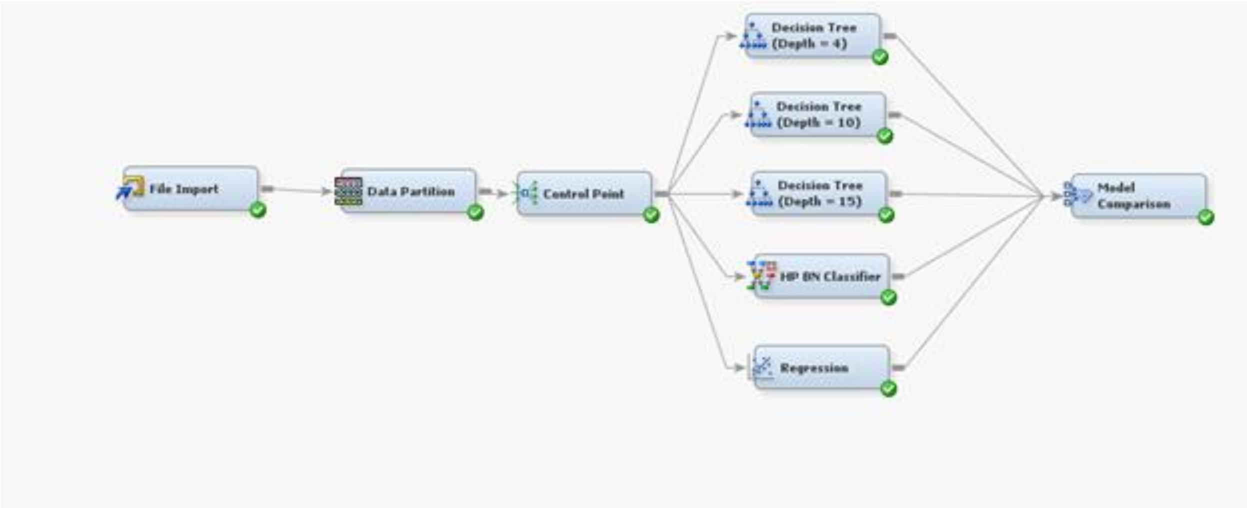
1. NUMBER_OF_DIAG - This variable counts the number of diagnosis codes the patient falls under. This variable is continuous since it is a count

2. ADMITTING_PRINC_DIAG - The data under this variable is categorical. This variable checks if the admitting diagnosis and the principal diagnosis code are the same. Admitting diagnosis code represents the diagnosis the patient was admitted under and principal diagnosis code is the diagnosis code the patient is getting treated for. The values under this column are either a 1 or a 0. 1 stands for "Yes" i.e. both the codes are same and 0 stands for "No" i.e. the codes are not equal

Data After cleaning

RECORD_ID	TYPE_OF_ADMISSION	SOURCE_OF_ADMISSION	PAT_AGE	ADMITTING_DIAGNOSIS	PRINC_DIAG_CODE	NUMBER_OF_DIAG	ADMITTING_PRINC_DIAG	RISK_MORTALITY
120133705286	3	1	17	53641	53641	13	1	3
120133705376	3	4	15	51881	389	13	0	4
120133705388	3	1	15	486	3812	13	0	3
120133705394	3	1	15	389	389	13	1	3
120133705395	3	4	15	389	389	13	1	3
120133705396	3	1	17	8770	5198	13	0	3
120133705399	3	1	24	496	51884	13	0	4
120133705403	3	1	18	486	3812	13	0	3
120133705404	3	1	16	389	3842	13	0	2
120133705405	9	4	19	51881	51881	13	1	4
120133705406	3	1	16	486	5070	10	0	3
120133705407	3	4	17	4019	51881	13	0	4
120133705409	3	1	17	51881	51884	13	0	3
120133705410	3	1	21	389	389	12	1	3
120133705412	3	4	16	51881	51881	13	1	4
120133705414	3	1	15	51881	3812	13	0	3
120133705416	3	4	18	496	51884	9	0	2
120133705481	3	2	15	4210	3812	13	0	3
120133705490	3	4	18	5990	3842	13	0	3
120133705623	3	1	17	4280	380	13	0	3
120133705625	3	4	16	436	389	13	0	4

4. Model Creation and Comparison



In SAS Enterprise Miner, we built the models as details in the following steps:

1. Added the ‘File Import’ node

2. Imported our Excel data file by providing the path to this node

3. Configured the variables under ‘Edit Variables’ & then ran this node

4. Next, we added a ‘Data Partition’ node, with the output of the ‘File Import’ node as the input for this one.

Name	Role	Level	Report	Order	Drop	Lower Limit	Upper Limit
ADMITTING_DIA	Input	Nominal	No		No	.	.
ADMITTING_PRI	Input	Binary	No		No	.	.
NUMBER_OF_DI	Input	Interval	No		No	.	.
PAT_AGE	Input	Nominal	No		No	.	.
PRINC_DIAG_C	Input	Nominal	No		No	.	.
RECORD_ID	ID	Nominal	No		No	.	.
RISK_MORTALIT	Target	Nominal	No		No	.	.
SOURCE_OF_AI	Input	Nominal	No		No	.	.
TYPE_OF_ADMI	Input	Nominal	No		No	.	.

5. Within this node, we set our partitions to 60-30-10, and then ran this node.

Data Set Allocations	
Training	60.0
Validation	30.0
Test	10.0

6. Created a control point node which makes sure that same partitioned data is passed to all the models

7. Finally, we added the Decision Trees, Logistic Regression and HP BN node (Naïve Bayes model). We added the decision tree with different depths to have a fair comparison between different models.

Naive Bayes

The predictor variables for our data have 6 categorical variables (TYPE_OF_ADMISSION, SOURCE_OF_ADMISSION, PAT_AGE, ADMITTING_DIAGNOSIS, PRINC_DIAG_CODE, ADMITTING_PRINC_DIAG)

and 1 numerical variable (NUMBER_OF_DIAG). The target variable (RISK_MORTALITY)

is also a categorical variable.

Considering this type of data, we have chosen to work with the Naive-Bayes model since it performs really well with categorical data and has a fairly simple computation. It is based on Bayes Theorem of conditional probability (from where it gets its name), but the assumptions in Naive-Bayes ignore the fact that there can be dependencies with the data and hence it is said to be 'naive'

We add one step in the above model creation step while implementing Naive Bayes. We change the 'Network Model' to 'Naive-Bayes' and set the 'Automatic Model Selection' to 'No', and then run this node as shown below

Variables	...
Network Model	Naive Bayes
Automatic Model Selection	No
Prescreen Variables	Yes
Variable Selection	No
Independence Test Statisti	G-Square
Significance Level	0.2
Missing Interval Variable	None
Missing Class Variable	None
Number of Bins	10
Maximum Parents	5
Network Structure	Parent-Child
Parenting Method	Set of Parents
⊟Validation with Train Data	

Assess Naive Bayes

Assessment of the model is an important step to analyze how well our chosen model performs. It gives us the accuracy of the model and how the model will classify new data. After running the HP BN node in the previous step, we got the following results which we have evaluated below as an assessment for our model. From a high-level analysis of our results, our Naive Bayes model seems to be pretty stable in predicting Risk Mortality.

Fit Statistics: The statistics below indicate a 'goodness of fit'

<u>Average Squared Error (ASE):</u> The smaller this value is, the closer we are to finding the line of best fit. The ASE value for our model is approximately 0.15 and it is stable across the different partitions of data, which indicates that our model is quite close to the best fit.

<u>Root Average Squared Error (RASE):</u> This is another statistic that tells us how concentrated our data is around the line of best fit. A smaller value of RASE is desirable and indicates that our model is providing a good fit for our data. Similar to ASE, this value for our model is pretty small (~0.37) and consistent across the three data partitions. This further solidifies the goodness of our data model.

<u>Misclassification Rate (MISC):</u> This rate is a model characteristic that is used to determine how accurate a network model is. Accuracy is determined by = (1-Misclassification Rate). This gives the actual accuracy of our model, which in our case is ~54%. We calculate this from our

misclassification rate which is roughly 46%. This value is pretty consistent across all the data partitions, which indicates that our model is actually doing well with the training, validation and test data.

Fit Statistics

Target=RISK_MORTALITY Target Label=RISK_MORTALITY

Fit Statistics	Statistics Label	Train	Validation	Test
ASE	Average Squared Error	0.14	0.15	0.15
DIV	Divisor for ASE	85556.00	42768.00	14276.00
MAX	Maximum Absolute Error	1.00	1.00	1.00
NOBS	Sum of Frequencies	21389.00	10692.00	3569.00
RASE	Root Average Squared Error	0.37	0.38	0.39
SSE	Sum of Squared Errors	11972.16	6324.15	2163.57
DISF	Frequency of Classified Cases	21389.00	10692.00	3569.00
MISC	Misclassification Rate	0.44	0.47	0.49
WRONG	Number of Wrong Classifications	9480.00	5041.00	1759.00

Classification Matrix: Also known as the confusion matrix, this is the table that gives us the misclassification rate and thus the accuracy of our model.

False Negative: Records that were classified incorrectly as negative

True Negative: Records that were correctly classified as negative

False Positive: Records that were incorrectly classified as positive

True Positive: Records that were correctly classified as positive

Event Classification Table

Data Role=TRAIN Target=RISK_MORTALITY Target Label=RISK_MORTALITY

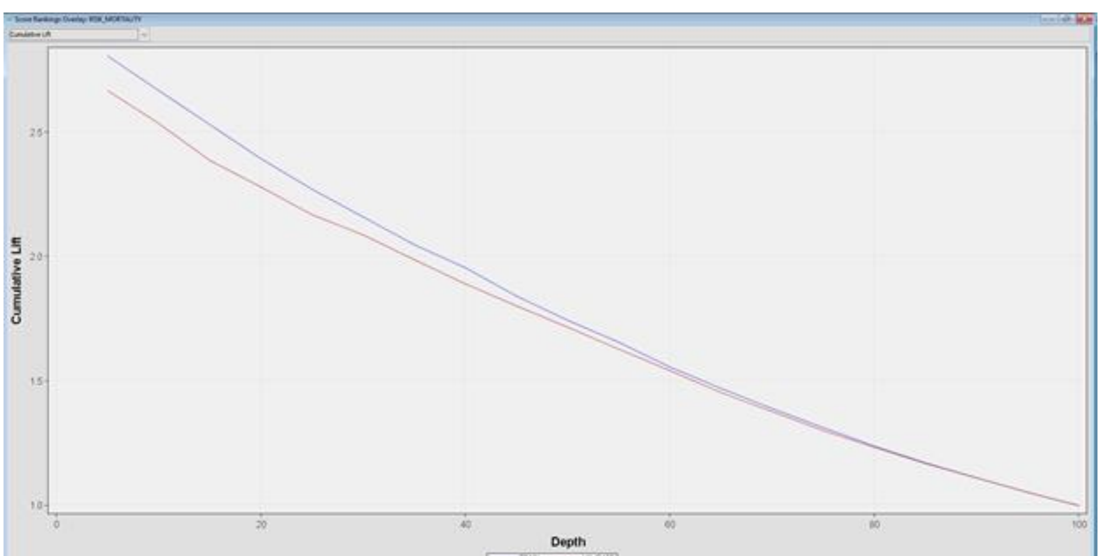
False Negative	True Negative	False Positive	True Positive
1734	12170	3130	4355

Data Role=VALIDATE Target=RISK_MORTALITY Target Label=RISK_MORTALITY

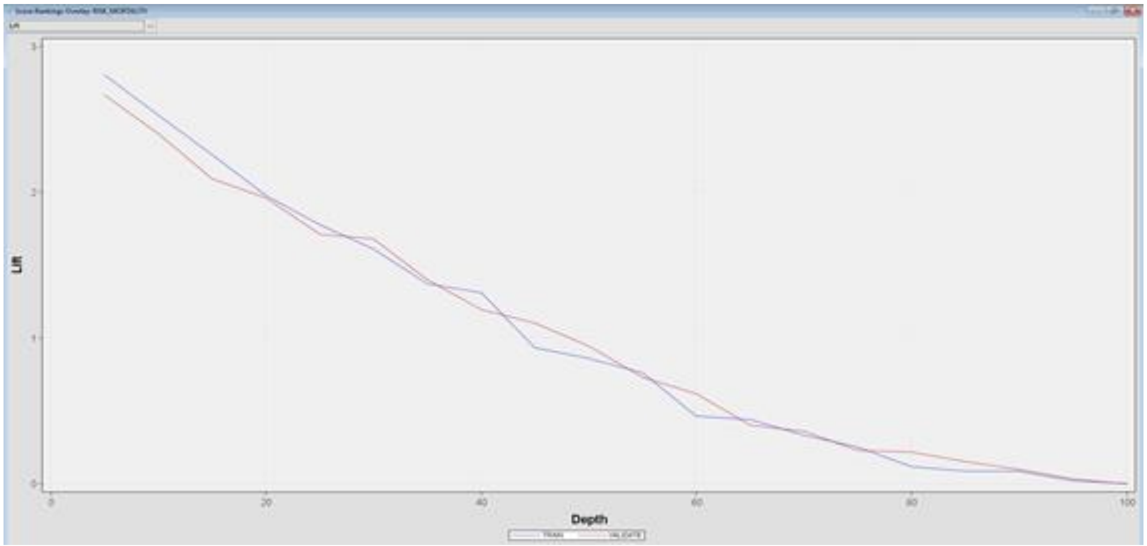
False Negative	True Negative	False Positive	True Positive
923	5942	1706	2121

Analysis of the output charts

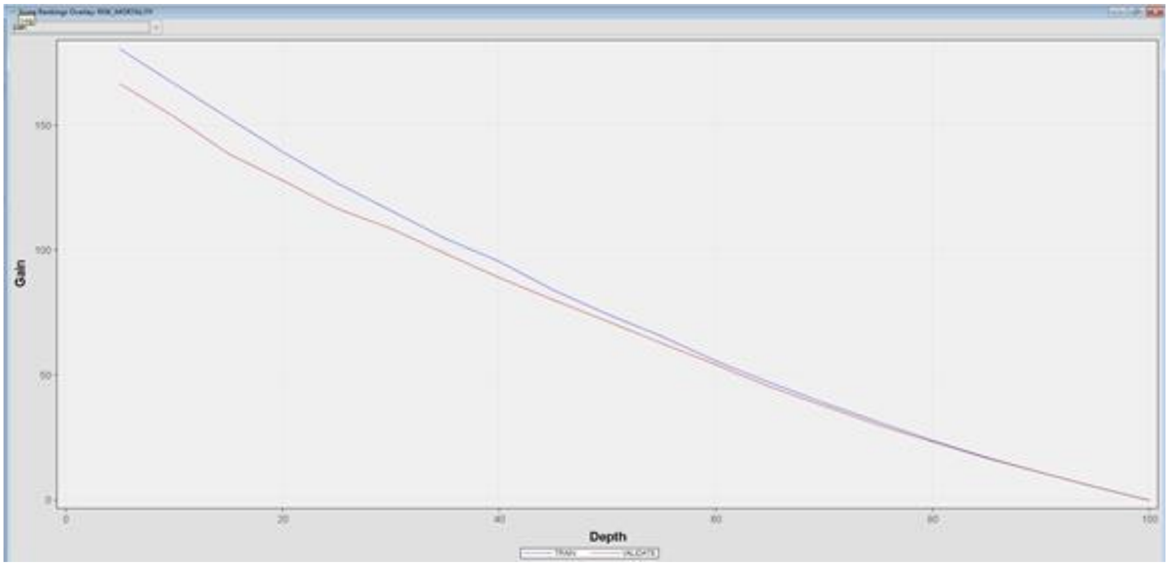
Cumulative Lift Chart: The Cumulative Lift Chart shows the relationship between the percent of depth (x-axis) and the percent of responses we will get (y-axis). What the below chart is saying is that if we go up to 35% of the depth of the network at random, we can capture roughly 50% of the data. This chart helps to decide how much of the data depth do we really want to explore to get meaningful information.



Lift Chart: Similar to the Cumulative Lift Chart, but it gives the actual lift. Without using a model, we would get no data at 8% depth of the network, but with this model, we're reaching almost 80% of the responses at 8% depth at random. This chart can be useful in determining after which point it becomes less effective and therefore more expensive to keep running.

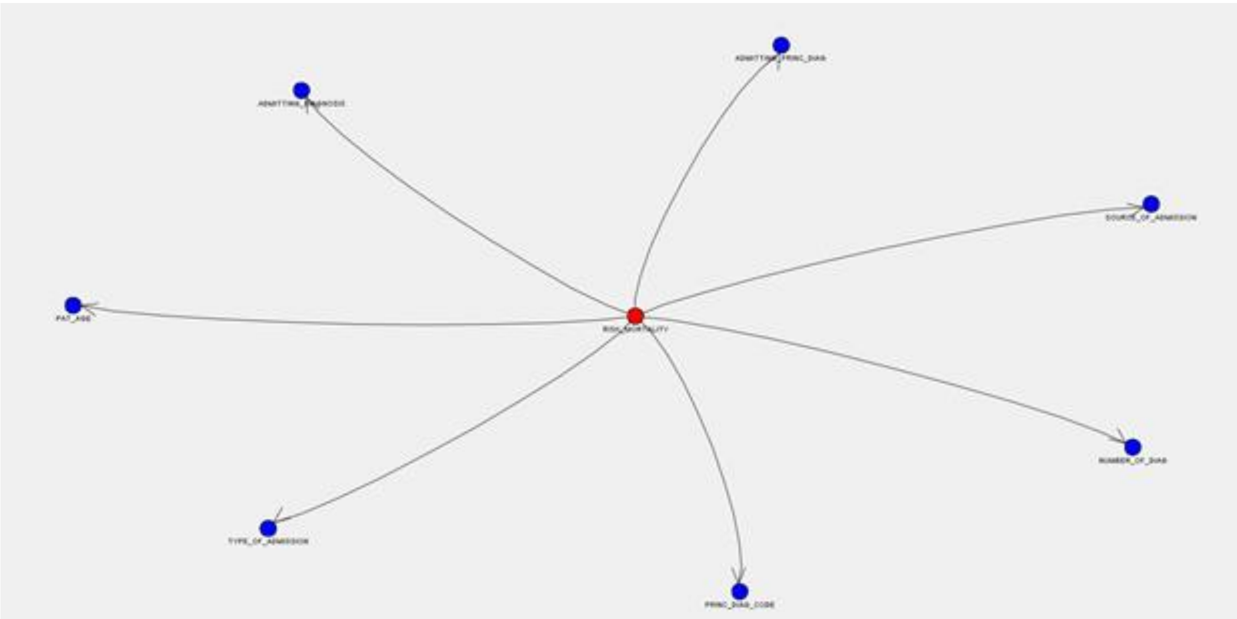


Gain Chart: The gain chart gives the ratio of the expected response using the model / expected response using a random sample. In other words, it measures the ratio between the training and validation data.



Bayesian Network

This network visually represents the target variable and predictor variables that it depends on. It is helpful in trying to understand the network at a glance



Decision Trees

In SAS Enterprise Miner, we built the model as details in the following steps (in addition to the above steps):

1. Setting the properties

    There are a couple of properties we can set for the Decision Tree such as the Splitting Rule, Node and Split Search among others.

    The Interval Target Criterion, Nominal Target Criterion and Ordinal Target Criterion specify the methods to evaluate candidate splitting rules for interval, nominal and ordinal variables, respectively and then choose the best one. Maximum Branch refers to the maximum number of branches in the decision tree, while Maximum Depth specifies the maximum number of generations of nodes allowed in our tree. [8]

Among the Node properties, Leaf Size indicates the minimum number of training observations in the lead node. Number of Rules refers to the splitting rules for each node and Split Size specifies the smallest number of training observations a node should have before it is split. [8]

Splitting Rule	
Interval Target Criterion	ProbF
Nominal Target Criterion	ProbChisq
Ordinal Target Criterion	Entropy
Significance Level	0.2
Missing Values	Use in search
Use Input Once	No
Maximum Branch	2
Maximum Depth	4
Minimum Categorical Size	5
Node	
Leaf Size	4
Number of Rules	5
Number of Surrogate Ru	0
Split Size	.
Split Search	
Use Decisions	No
Use Priors	No
Exhaustive	5000

2. Run model

   Once we set the properties for the different models and depths, we ran the Decision Tree node in SAS.

Assess Decision Trees

Fit Statistics: The statistics below indicate a 'goodness of fit'

<u>Average Squared Error (ASE):</u> The smaller this value is, the closer we are to finding the line of best fit. The ASE value for our model is approximately 0.154 across the training, validation and test partitions. It is fairly stable.

Root Average Squared Error (RASE): This is another statistic that tells us how concentrated our data is around the line of best fit. A smaller value of RASE is desirable and indicates that our model is providing a good fit for our data. The RASE value for this decision tree is about 0.39.

Misclassification Rate (MISC): This rate is a model characteristic that is used to determine how accurate a network model is. Accuracy is determined by = (1-Misclassification Rate). Using this formula, we get an accuracy of roughly 50% for this decision tree model.

Fit Statistics

Target	Target Label	Fit Statistics	Statistics Label	Train	Validation	Test
RISK_MORTALITY	RISK_MORTALITY	_NOBS_	Sum of Frequencies	21388	10693	3569
RISK_MORTALITY	RISK_MORTALITY	_MISC_	Misclassification Rate	0.519263	0.532591	0.503783
RISK_MORTALITY	RISK_MORTALITY	_MAX_	Maximum Absolute Err...	0.997934	1	1
RISK_MORTALITY	RISK_MORTALITY	_SSE_	Sum of Squared Errors	13204.42	6740.647	2192.453
RISK_MORTALITY	RISK_MORTALITY	_ASE_	Average Squared Error	0.154344	0.157595	0.153576
RISK_MORTALITY	RISK_MORTALITY	_RASE_	Root Average Squared...	0.392866	0.396982	0.391888
RISK_MORTALITY	RISK_MORTALITY	_DIV_	Divisor for ASE	85552	42772	14276
RISK_MORTALITY	RISK_MORTALITY	_DFT_	Total Degrees of Free...	64164	.	.

Classification Matrix: Also known as the confusion matrix, this is the table that gives us the misclassification rate and thus the accuracy of our model.

Event Classification Table

Data Role=TRAIN Target=RISK_MORTALITY Target Label=RISK_MORTALITY

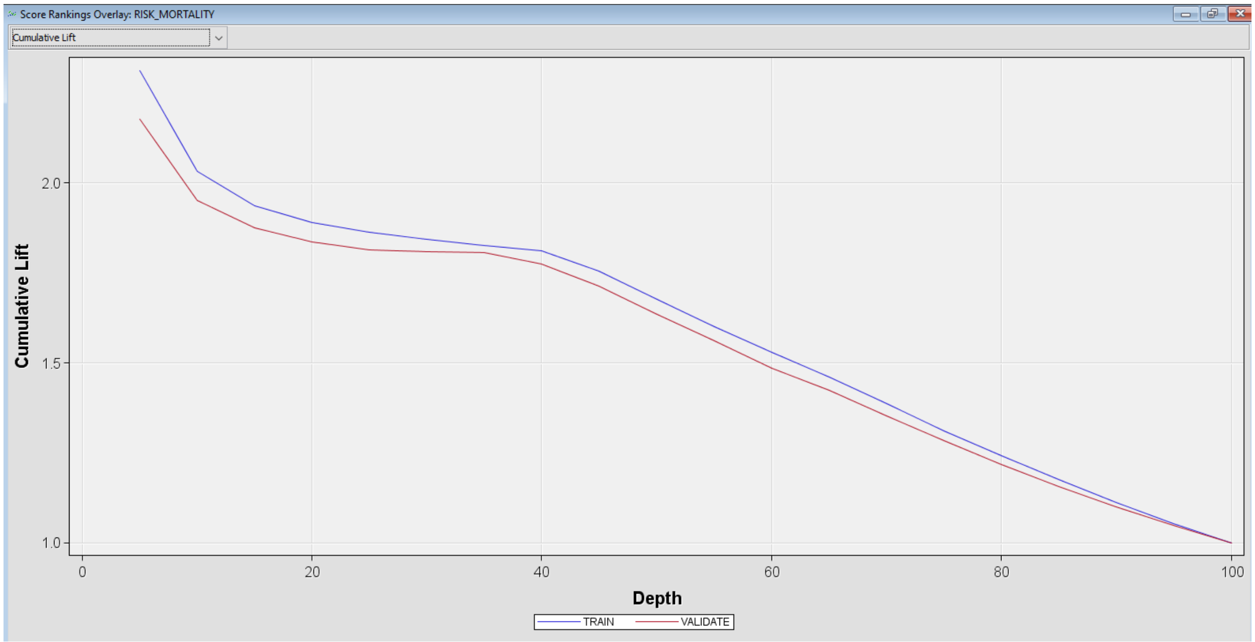
False Negative	True Negative	False Positive	True Positive
2363	11688	3612	3726

Data Role=VALIDATE Target=RISK_MORTALITY Target Label=RISK_MORTALITY

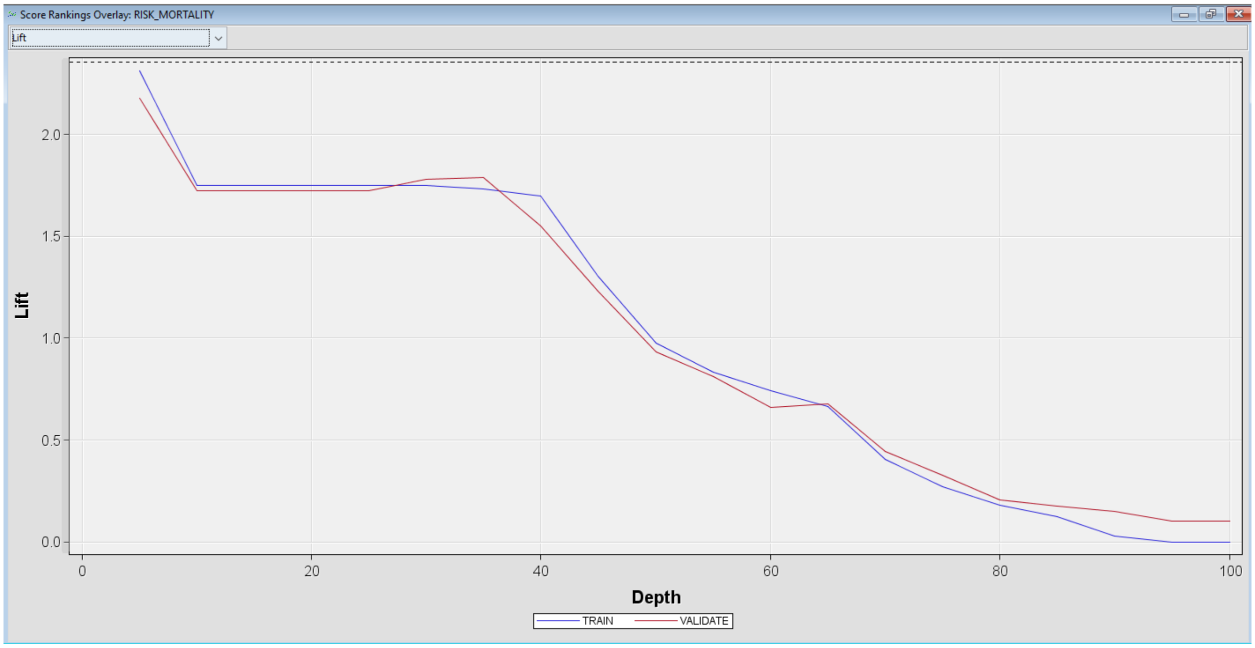
False Negative	True Negative	False Positive	True Positive
1169	5816	1832	1875

Analysis of the output charts

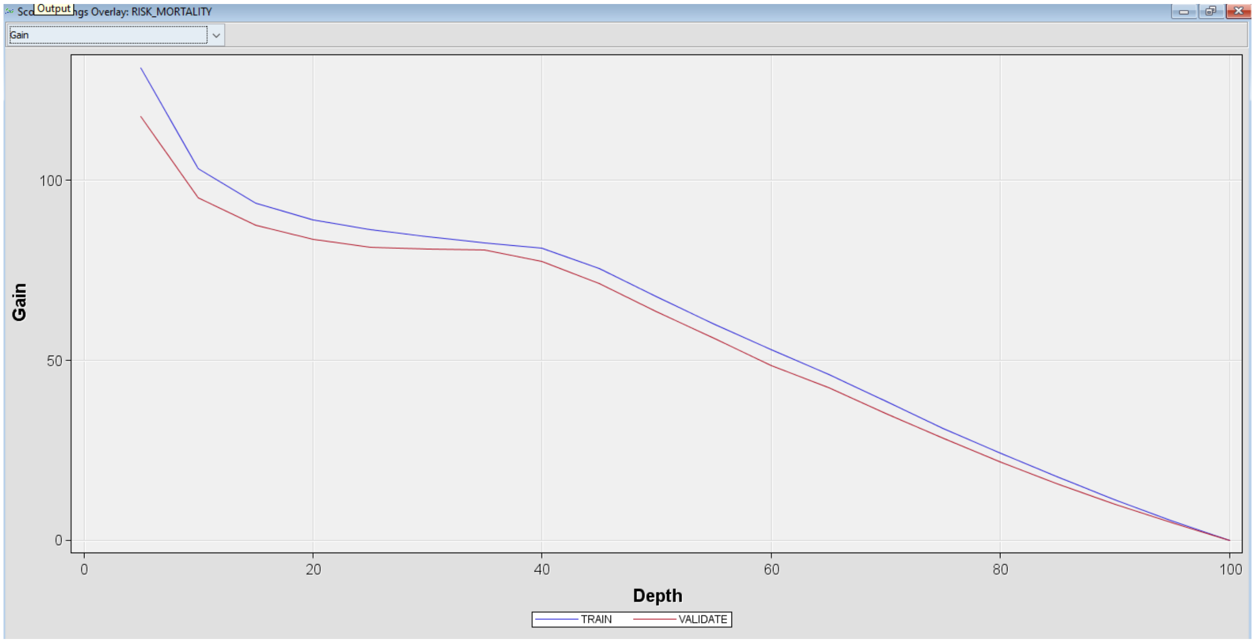
Cumulative Lift Chart: The Cumulative Lift Chart shows the relationship between the percent of depth (x-axis) and the percent of responses we will get (y-axis). What the below chart is saying is that if we go up to 20% of the depth of the network at random, we can capture more than 50% of the data. This chart helps to decide how much of the data depth do we really want to explore to get meaningful information.



Lift Chart: Similar to the Cumulative Lift Chart, but it gives the actual lift. Without using a model we would get no data at 10% depth of the network, but with this model, we're reaching almost all of the responses at 10% depth at random. This chart can be useful in determining after which point it becomes less effective and therefore more expensive to keep running.

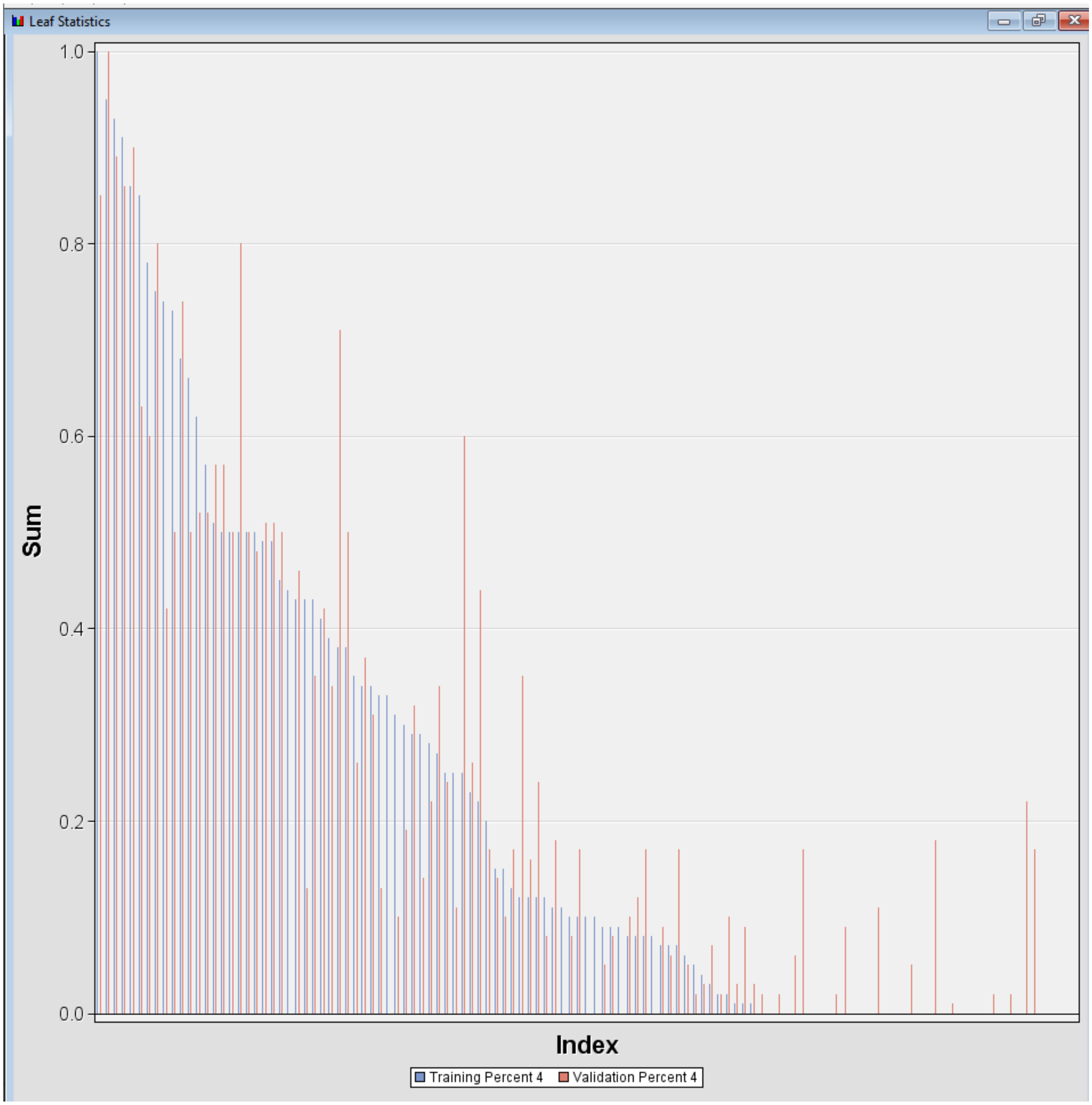


Gain Chart: The gain chart gives the ratio of the expected response using the model / expected response using a random sample. In other words, it measures the ratio between the training and validation data.

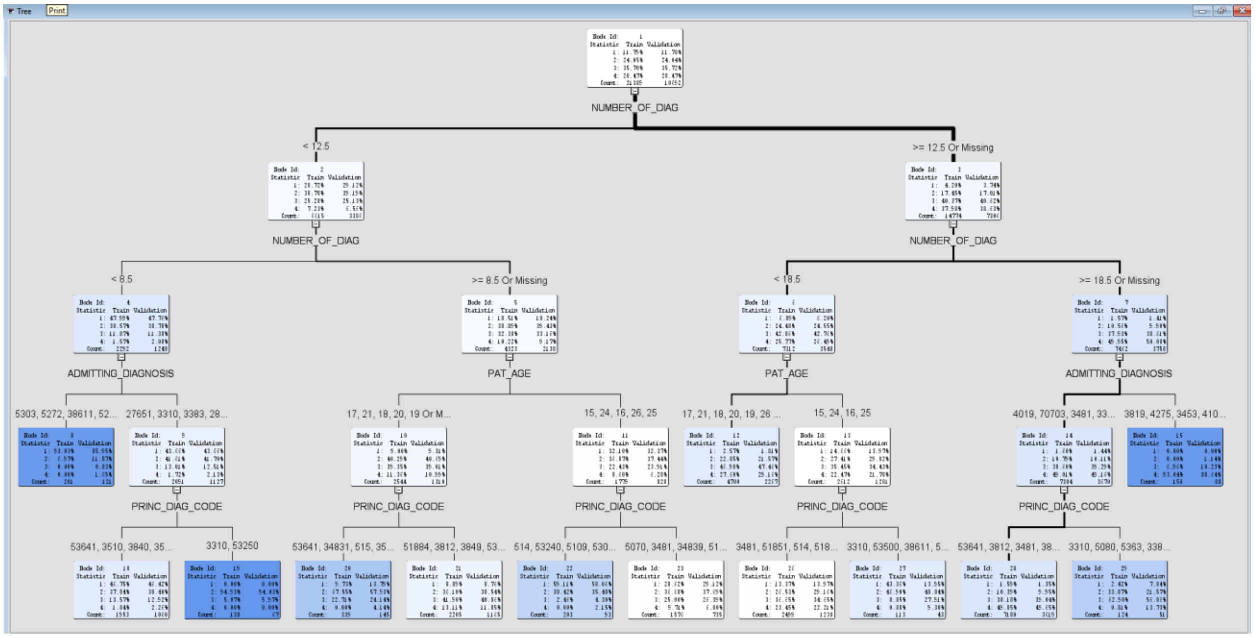


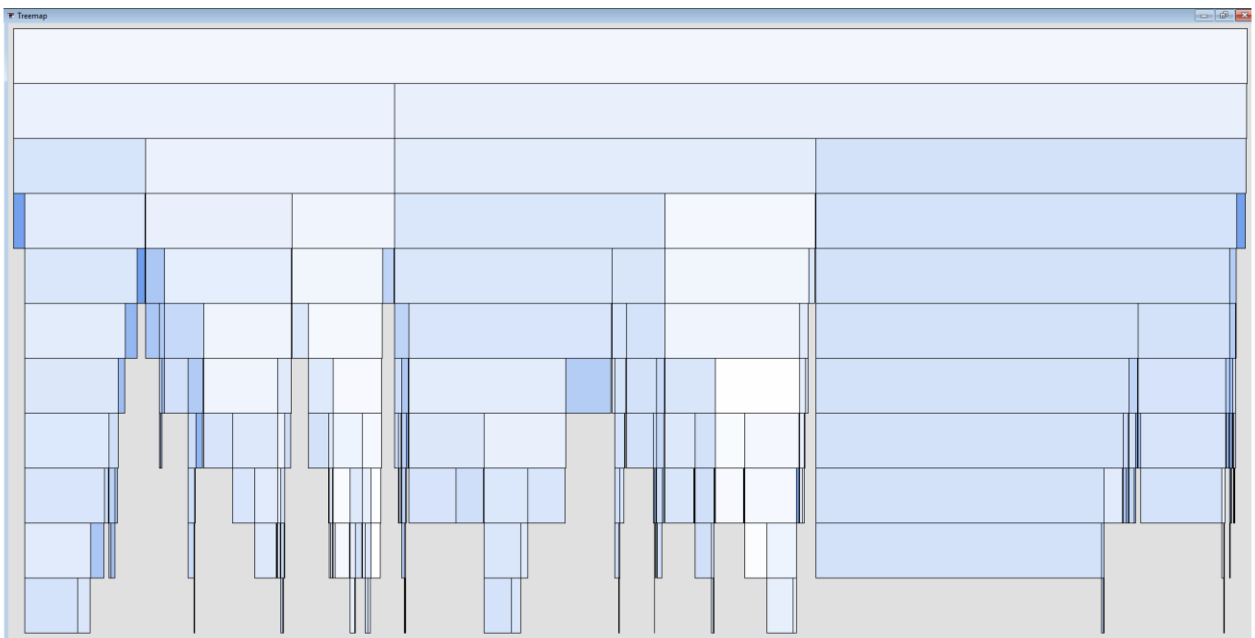
Leaf Statistics

The graph below shows the percentage of observations for each leaf node in the decision tree. It gives us a sort of frequency distribution amongst the leaf nodes in our tree.



Visual Representation of the decision tree with depth 4





Node Rules:

These were the rules that SAS E Miner formulated for our decision tree. Below are some of the rules, wherein the value of Admitting Diagnosis, Number of Diagnosis and Patient Age is checked to classify the value of Risk Mortality. These node rules are simply the decision statement i.e. IF-THEN statements. Following are the node rules when depth = 4

--

Node = 8

--

if NUMBER_OF_DIAG < 8.5

AND ADMITTING_DIAGNOSIS IS ONE OF: 5303, 5272, 38611, 5225, 7820, 3331, 3453, 53081, 5300, 3320, 3313, 3314, 3315, 35781, 3682, 3501, 78659, 7813, 340, 6820, 33818, 53909

then

Tree Node Identifier = 8

Number of Observations = 201

Predicted: RISK_MORTALITY=4 = 0.00

Predicted: RISK_MORTALITY=3 = 0.00

Predicted: RISK_MORTALITY=2 = 0.07

Predicted: RISK_MORTALITY=1 = 0.93

--

Node = 12

--

if PAT_AGE IS ONE OF: 17, 21, 18, 20, 19, 26 or MISSING

AND NUMBER_OF_DIAG < 18.5 AND NUMBER_OF_DIAG >= 12.5

then

Tree Node Identifier = 12

Number of Observations = 4700

Predicted: RISK_MORTALITY=4 = 0.28

Predicted: RISK_MORTALITY=3 = 0.47

Predicted: RISK_MORTALITY=2 = 0.23

Predicted: RISK_MORTALITY=1 = 0.03

To choose the best Decision Trees, we have performed comparison among multiple decision trees with varying depths. Hence, we started with performing decision trees beginning from depth 4 to depth of 20. The comparison amongst these models in done in the model comparison section.

Logistic Regression

Logistic Regression is a variation of Multiple Linear Regression, which is used when the outcome variable is a categorical variable (as opposed to a numerical variable in MLR). It uses a logarithmic function (logit) to predict/classify the outcome.

Stepwise Regression is a method of building a model by adding or removing predictor variables.

Assess Logistic Regression Model

Fit Statistics: The statistics below indicate a ‘goodness of fit’

Average Squared Error (ASE): The smaller this value is, the closer we are to finding the line of best fit. The ASE value for our model is approximately 0.14, and is fairly consistent across the different partitions as shown below.

Root Average Squared Error (RASE): This is another statistic that tells us how concentrated our data is around the line of best fit. A smaller value of RASE is desirable and indicates that our model is providing a good fit for our data. In the screenshot below, this value is indicated by RMSE, which is ~0.38 for this logistic regression model.

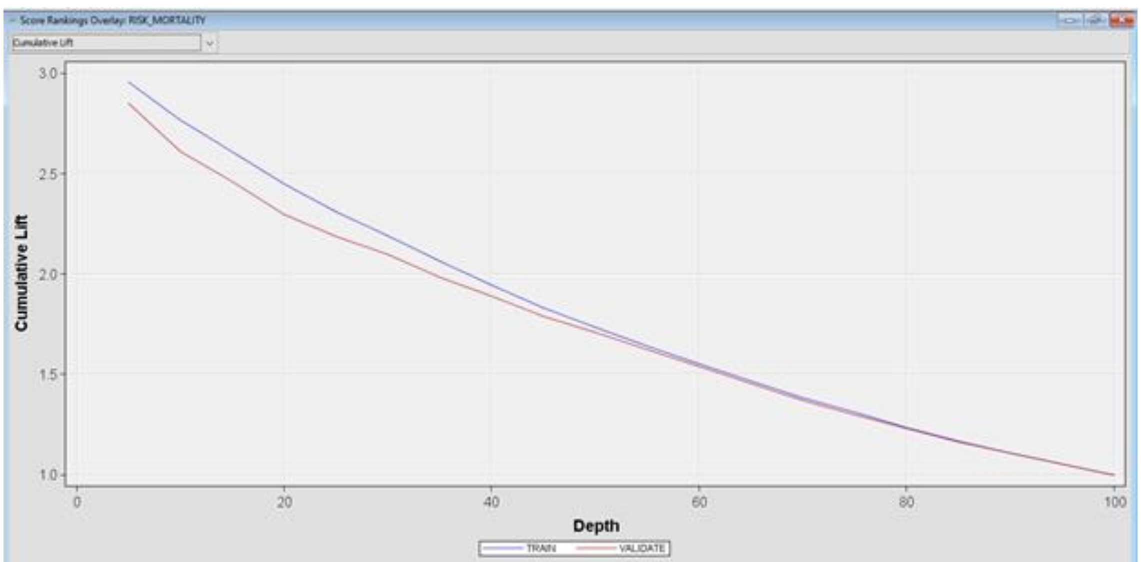
Misclassification Rate (MISC): This rate is a model characteristic that is used to determine how accurate a network model is. Accuracy is determined by = (1-Misclassification Rate). Using this formula, we get an accuracy rate of ~52%, and there is a little bit of a difference between the training partition and the other values.

Fit Statistics

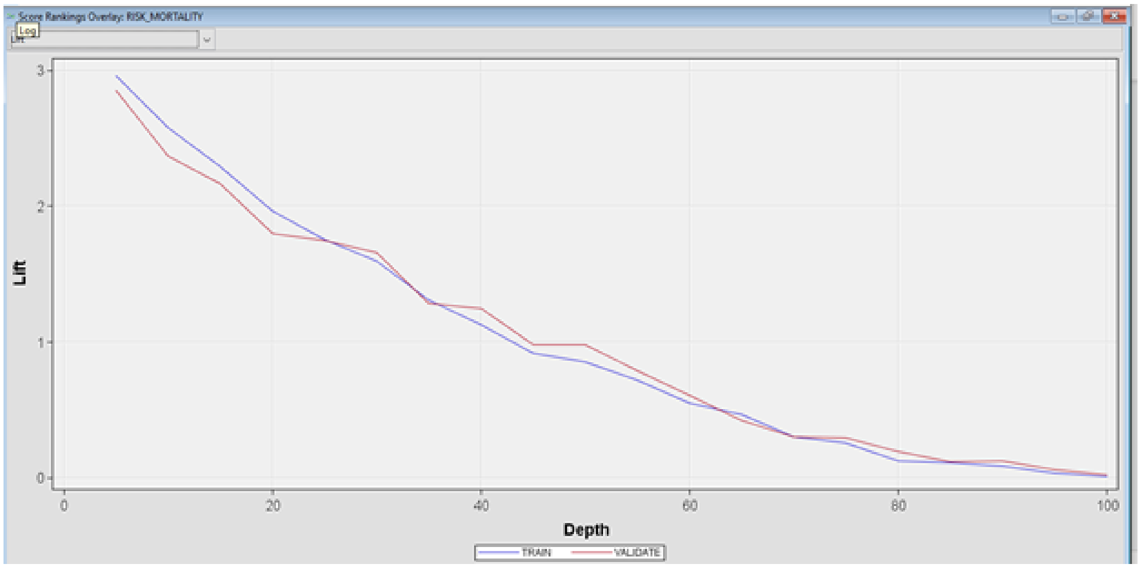
Target	Target Label	Fit Statistics	Statistics Label	Train	Validation	Test
RISK_MORTALITY	RISK_MORTALITY	_AIC_	Akaike's Information Criterion	43381.47		
RISK_MORTALITY	RISK_MORTALITY	_ASE_	Average Squared Error	0.137724	0.145078	0.14
RISK_MORTALITY	RISK_MORTALITY	_AVERR_	Average Error Function	0.476477	0.502865	0.50
RISK_MORTALITY	RISK_MORTALITY	_DFE_	Degrees of Freedom for Error	62859		
RISK_MORTALITY	RISK_MORTALITY	_DFM_	Model Degrees of Freedom	1308		
RISK_MORTALITY	RISK_MORTALITY	_DFT_	Total Degrees of Freedom	64167		
RISK_MORTALITY	RISK_MORTALITY	_DIV_	Divisor for ASE	85556	42768	1
RISK_MORTALITY	RISK_MORTALITY	_ERR_	Error Function	40765.47	21506.54	723
RISK_MORTALITY	RISK_MORTALITY	_FPE_	Final Prediction Error	0.143455		
RISK_MORTALITY	RISK_MORTALITY	_MAX_	Maximum Absolute Error	0.997925	0.997318	0.99
RISK_MORTALITY	RISK_MORTALITY	_MSE_	Mean Square Error	0.14059	0.145078	0.14
RISK_MORTALITY	RISK_MORTALITY	_NOBS_	Sum of Frequencies	21389	10692	
RISK_MORTALITY	RISK_MORTALITY	_NW_	Number of Estimate Weights	1308		
RISK_MORTALITY	RISK_MORTALITY	_RASE_	Root Average Sum of Squares	0.371112	0.380892	0.38
RISK_MORTALITY	RISK_MORTALITY	_RFPE_	Root Final Prediction Error	0.378755		
RISK_MORTALITY	RISK_MORTALITY	_RMSE_	Root Mean Squared Error	0.374953	0.380892	0.38
RISK_MORTALITY	RISK_MORTALITY	_SBC_	Schwarz's Bayesian Criterion	55244.05		
RISK_MORTALITY	RISK_MORTALITY	_SSE_	Sum of Squared Errors	11783.1	6204.714	208
RISK_MORTALITY	RISK_MORTALITY	_SUMW_	Sum of Case Weights Times Freq	85556	42768	1
RISK_MORTALITY	RISK_MORTALITY	_MISC_	Misclassification Rate	0.435177	0.472222	0.47

Analysis of the output charts

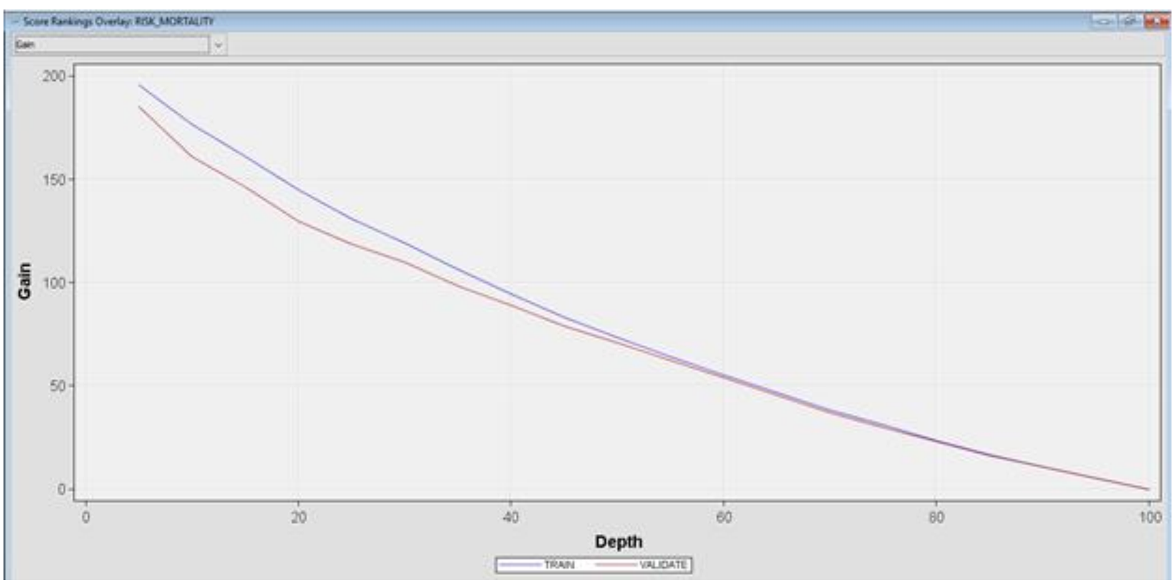
Cumulative Lift Chart: The Cumulative Lift Chart shows the relationship between the depth (x-axis) and the percent of responses we will get (y-axis). What the below chart is saying is that if we go up to 25% of the depth of the network at random, we can capture roughly 65% of the data. This chart helps to decide how much of the data depth do we really want to explore to get meaningful information.



Lift Chart: Similar to the Cumulative Lift Chart, but it gives the actual lift. Without using a model, we would get no data at 7-8% depth of the network, but with this model, we're reaching almost 95% of the responses at 7-8% depth at random. This chart can be useful in determining after which point it becomes less effective and therefore more expensive to keep running.



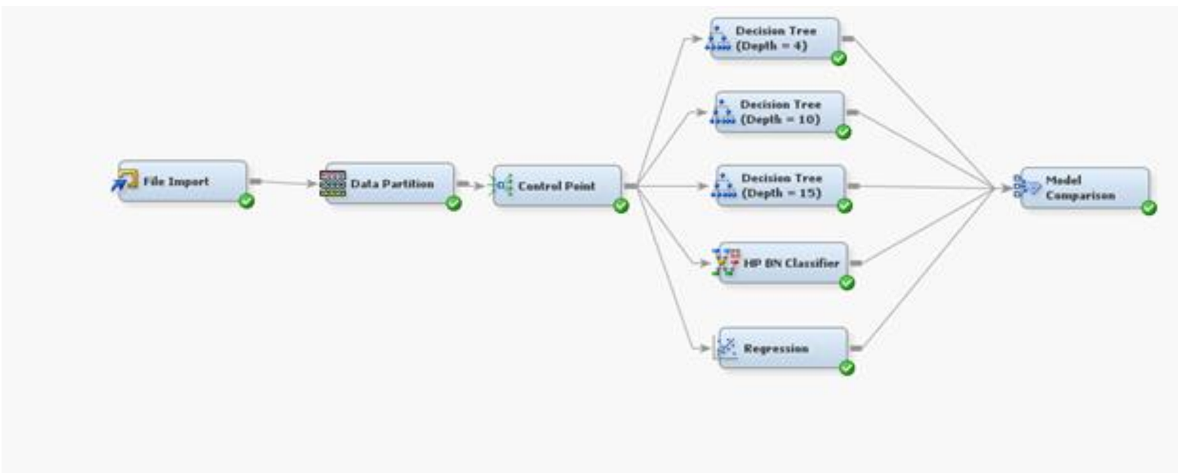
Gain Chart: The gain chart gives the ratio of the expected response using the model / expected response using a random sample. In other words, it measures the ratio between the training and validation data.



Model Comparison

It is important to perform a comparison between different data mining models so that we can assess the best one for our data and purpose. The Model Comparison Node in SAS E Miner makes it easy to do so. It runs and compares all the models, checks for the fit statistics and then gives us the best model for our data.

Using Model Comparison node in SAS Enterprise Miner



Variables

Below is an overview of our variables that are input into each of the different models. It shows the variable type and each of their roles.

Variables - FIMPORT

(none) not Equal to

Columns: Label Mining

Name	Role	Level	Report	Order	Drop	Lower Limit	Upper Limit
ADMITTING_DIA	Input	Nominal	No		No	.	.
ADMITTING_PRI	Input	Binary	No		No	.	.
NUMBER_OF_DI	Input	Interval	No		No	.	.
PAT_AGE	Input	Nominal	No		No	.	.
PRINC_DIAG_C	Input	Nominal	No		No	.	.
RECORD_ID	ID	Nominal	No		No	.	.
RISK_MORTALIT	Target	Ordinal	No		No	.	.
SOURCE_OF_AE	Input	Nominal	No		No	.	.
TYPE_OF_ADMIS	Input	Nominal	No		No	.	.

Data Partitioning

Random Data Partitioning is important to ensure the data used for the different partitions in not sequential. The data for the training partition is usually the largest because this is the partition using which our model actually learns. Similar to the partitions we made before, our data was partitioned in the following way :

Partition Summary

Type	Data Set	Number of Observations
DATA	EMWS1.FIMPORT_train	35650
TRAIN	EMWS1.Part_TRAIN	21389
VALIDATE	EMWS1.Part_VALIDATE	10692
TEST	EMWS1.Part_TEST	3569

The following report displays the summary statistics for the target variable, RISK_MORTALITY in source data, train, test and validate data sets. It shows the number of records belonging to each of the risk mortality categories and proportion of those records.

```
*------------------------------------------------------------*
* Report Output
*------------------------------------------------------------*
```

Summary Statistics for Class Targets

Data=DATA

Variable	Numeric Value	Formatted Value	Frequency Count	Percent	Label
RISK_MORTALITY	1	1	4201	11.7840	RISK_MORTALITY
RISK_MORTALITY	2	2	8571	24.0421	RISK_MORTALITY
RISK_MORTALITY	3	3	12729	35.7055	RISK_MORTALITY
RISK_MORTALITY	4	4	10149	28.4684	RISK_MORTALITY

Data=TEST

Variable	Numeric Value	Formatted Value	Frequency Count	Percent	Label
RISK_MORTALITY	1	1	421	11.7960	RISK_MORTALITY
RISK_MORTALITY	2	2	858	24.0403	RISK_MORTALITY
RISK_MORTALITY	3	3	1274	35.6963	RISK_MORTALITY
RISK_MORTALITY	4	4	1016	28.4674	RISK_MORTALITY

Data=TRAIN

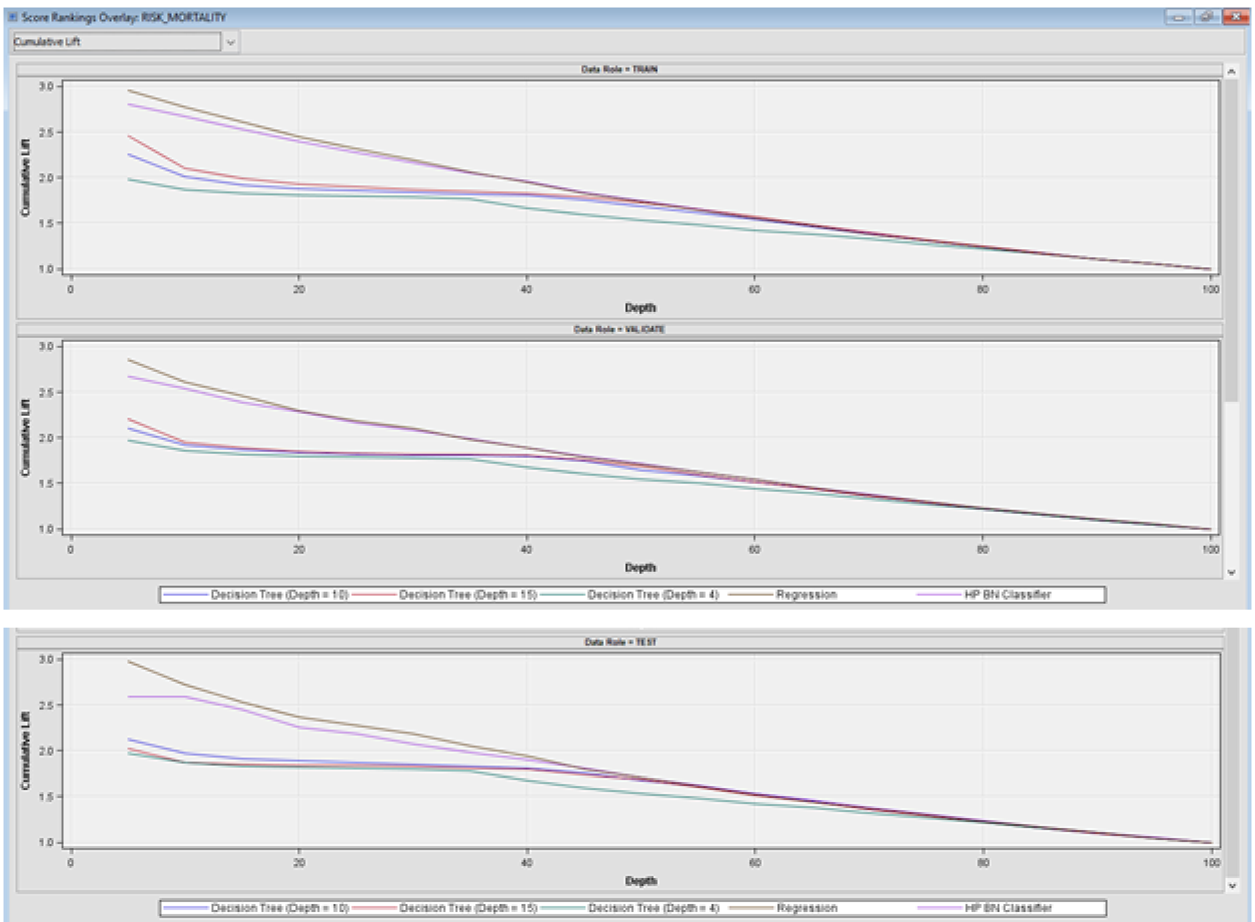
Variable	Numeric Value	Formatted Value	Frequency Count	Percent	Label
RISK_MORTALITY	1	1	2521	11.7864	RISK_MORTALITY
RISK_MORTALITY	2	2	5143	24.0451	RISK_MORTALITY
RISK_MORTALITY	3	3	7636	35.7006	RISK_MORTALITY
RISK_MORTALITY	4	4	6089	28.4679	RISK_MORTALITY

Data=VALIDATE

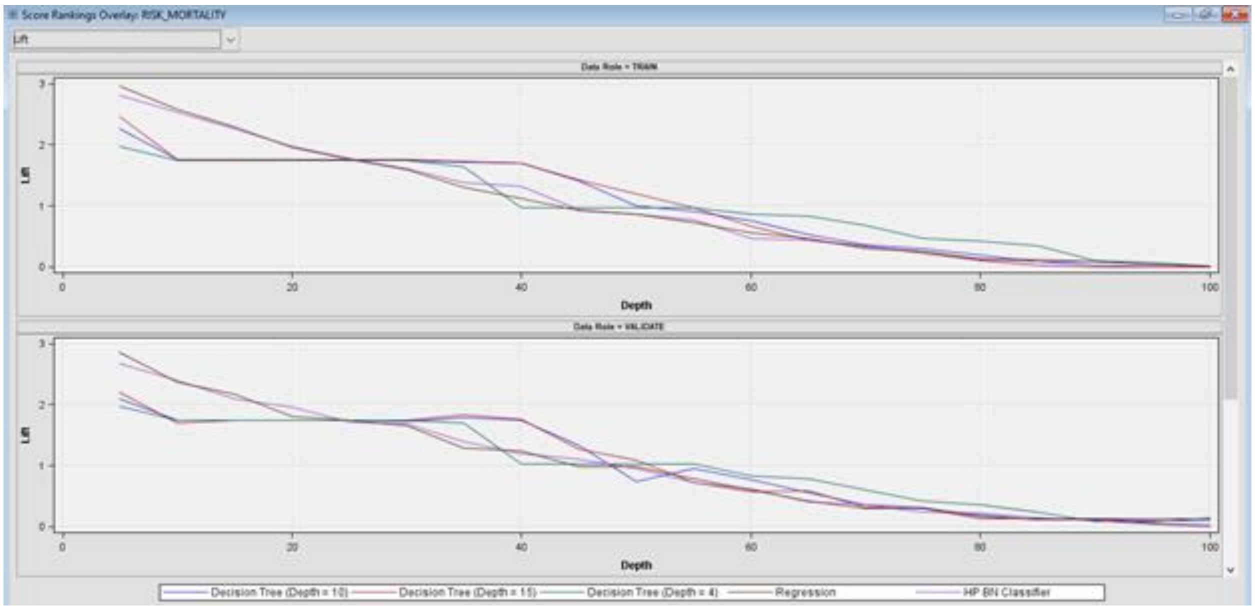
Variable	Numeric Value	Formatted Value	Frequency Count	Percent	Label
RISK_MORTALITY	1	1	1259	11.7752	RISK_MORTALITY
RISK_MORTALITY	2	2	2570	24.0367	RISK_MORTALITY
RISK_MORTALITY	3	3	3819	35.7183	RISK_MORTALITY
RISK_MORTALITY	4	4	3044	28.4699	RISK_MORTALITY

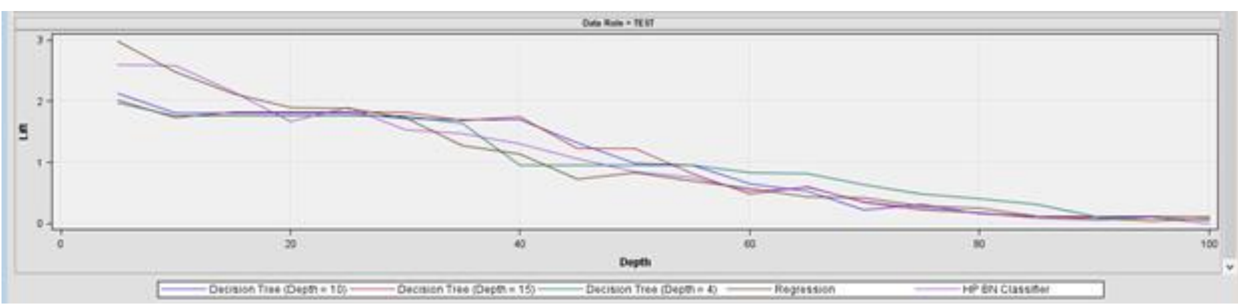
Analysis of the output charts

Cumulative Lift Chart: The Cumulative Lift Chart shows the relationship between the depth of the tree (x-axis) and the lift (y-axis). This chart measures the model performance. The baseline model i.e. no model lies at Lift value = 1. Hence, the greater the area between the curve and the baseline model, the better the model. The charts displays the curves for train, validate and test data sets. We can derive that the Naive Bayes is performing pretty well (not better than logistic regression though). However, since our measure of model performance is misclassification rate, we'll be sticking to the results obtained from the classification matrix. This chart helps to decide how much of the data depth do we really want to explore to get meaningful information.

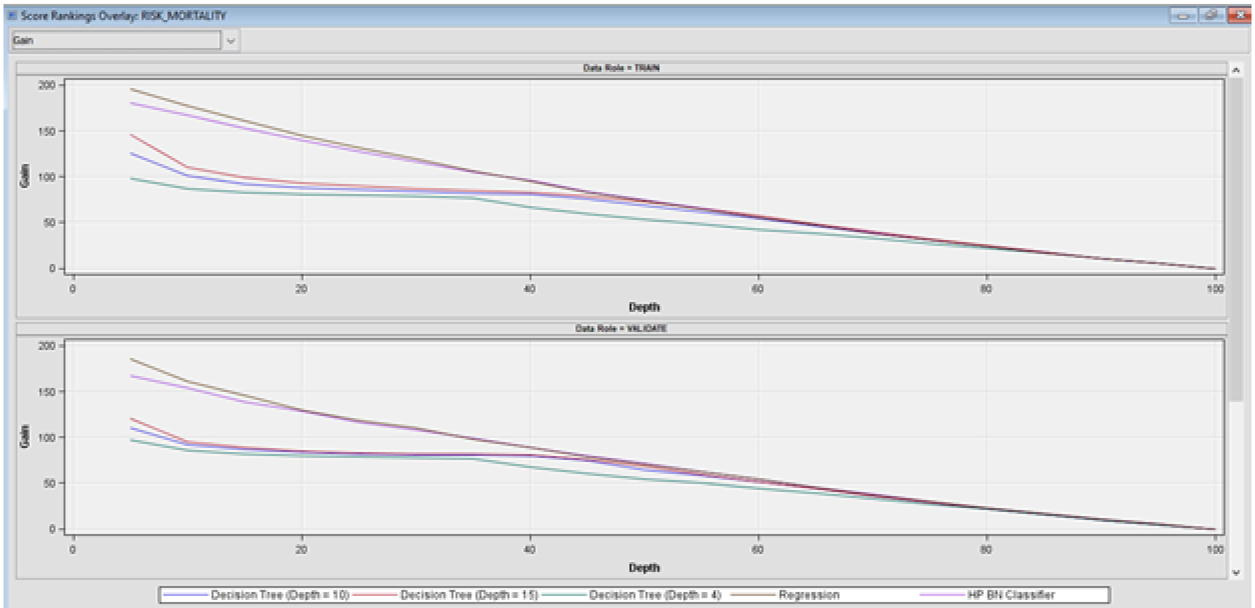


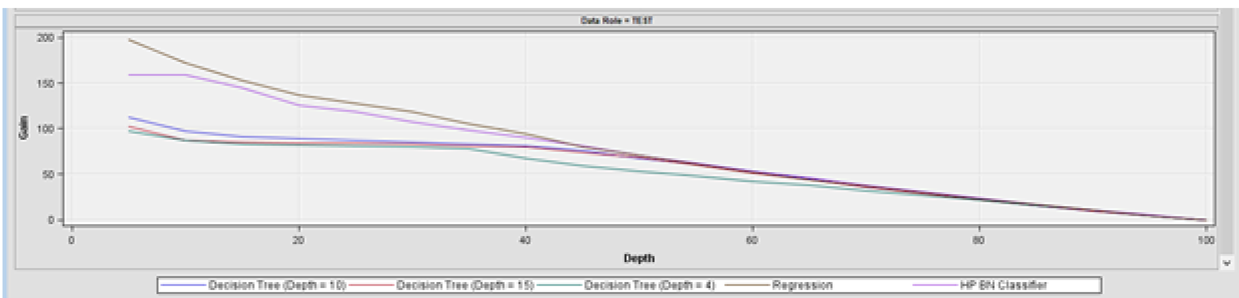
Lift Chart: Instead of focusing on cumulative lift, this chart focuses on lift value. It compares how well the model performed with respect to no model by plotting the results outcome predicted by the model and results of no model. The y-axis represents the lift value and the x-axis represents the depth value. As shown in the charts below, as depth increases, the model performance decreases A lift value greater than 1 depicts that the model performance in good. This chart can be useful in determining after which point it becomes less effective and therefore more expensive to keep running. Moreover, the test data set depicts that the models at depth (20) have the same performance and the performance of the model keeps on reducing





Gain Chart: The gain chart gives the ratio of the expected response using the model / expected response using a random sample. In other words, it measures the ratio between the training and validation data.





Classification Matrix: Also known as the confusion matrix, this is the table that gives us the misclassification rate and thus the accuracy of our model.

<u>False Negative:</u> Records that were classified incorrectly as negative

<u>True Negative:</u> Records that were correctly classified as negative

<u>False Positive:</u> Records that were incorrectly classified as positive

<u>True Positive:</u> Records that were correctly classified as positive

Event Classification Table
Model Selection based on Valid: Misclassification Rate (_VMISC_)

Model Node	Model Description	Data Role	Target	Target Label	False Negative	True Negative	False Positive	True Positive
Reg	Regression	TRAIN	RISK_MORTALITY	RISK_MORTALITY	2432	13305	1995	3657
Reg	Regression	VALIDATE	RISK_MORTALITY	RISK_MORTALITY	1298	6543	1105	1746
HPBNC	HP BN Classifier	TRAIN	RISK_MORTALITY	RISK_MORTALITY	1734	12170	3130	4355
HPBNC	HP BN Classifier	VALIDATE	RISK_MORTALITY	RISK_MORTALITY	923	5942	1706	2121
Tree	Decision Tree	TRAIN	RISK_MORTALITY	RISK_MORTALITY	2363	11688	3612	3726
Tree	Decision Tree	VALIDATE	RISK_MORTALITY	RISK_MORTALITY	1169	5816	1832	1875
Tree4	Decision Tree	TRAIN	RISK_MORTALITY	RISK_MORTALITY	1456	10799	4501	4633
Tree4	Decision Tree	VALIDATE	RISK_MORTALITY	RISK_MORTALITY	742	5410	2238	2302
Tree3	Decision Tree	TRAIN	RISK_MORTALITY	RISK_MORTALITY	1609	11136	4164	4480
Tree3	Decision Tree	VALIDATE	RISK_MORTALITY	RISK_MORTALITY	816	5579	2069	2228

Fit Statistics
Model Selection based on Valid: Misclassification Rate (_VMISC_)

Selected Model	Model Node	Model Description	Valid: Misclassification Rate	Train: Average Squared Error	Train: Misclassification Rate	Valid: Average Squared Error
Y	HPBNC	HP BN Classifier	0.47147	0.13993	0.44322	0.14787
	Reg	Regression	0.47222	0.13772	0.43518	0.14508
	Tree3	Decision Tree	0.47306	0.13951	0.46454	0.14914
	Tree4	Decision Tree	0.49111	0.14519	0.48459	0.15010
	Tree	Decision Tree	0.53423	0.15651	0.52952	0.15672

We have decided on that the misclassification rate as a measure of model performance hence we are going to compare the misclassifications rates of all the model (along with different depths of the decision tree). The above picture depicts that Naive Bayes has the lowest misclassification rates which makes sense it has the number of true positives and true negatives. We have considered performing decision tree with varying depths in order to understand the trend of the misclassification rate for the same

Moreover, it would make more sense to evaluate the rates for validation and test data since it would be a better indication of the model performance. Both Logistic and Naive Bayes are the competitive models, with extremely close misclassification rates. In the validation test data set, the rate is exactly the same. However, if we go see the lift value, Naive Bayes seem to perform better than Logistic Regression. On the other hand, logistic regression performed slightly better than Naive Bayes, since this difference if not very significant, we'll still stick to Naive Bayes being our final model.

Data Role=Valid

Statistics	HPBNC	Reg	Tree3	Tree4	Tree
Valid: Kolmogorov-Smirnov Statistic	0.51	0.50	0.48	0.47	0.38
Valid: Average Squared Error	0.15	0.15	0.15	0.15	0.16
Valid: Roc Index	0.83	0.83	0.78	0.78	0.75
Valid: Average Error Function	.	0.50	.	.	.
Valid: Bin-Based Two-Way Kolmogorov-Smirnov Probability Cutoff	0.35	0.31	0.34	0.38	0.28
Valid: Cumulative Percent Captured Response	25.36	26.12	19.53	19.25	18.58
Valid: Percent Captured Response	12.02	11.85	8.52	8.74	8.73
Valid: Frequency of Classified Cases	10692.00	.	.	.	.
Valid: Divisor for VASE	42768.00	42768.00	42768.00	42768.00	42768.00
Valid: Error Function	.	21506.54	.	.	.
Valid: Gain	153.42	160.97	95.11	92.36	85.67
Valid: Gini Coefficient	0.65	0.66	0.57	0.56	0.51
Valid: Bin-Based Two-Way Kolmogorov-Smirnov Statistic	0.50	0.50	0.48	0.47	0.38
Valid: Kolmogorov-Smirnov Probability Cutoff	0.32	0.23	0.33	0.34	0.24
Valid: Cumulative Lift	2.53	2.61	1.95	1.92	1.86
Valid: Lift	2.40	2.37	1.70	1.75	1.74
Valid: Maximum Absolute Error	1.00	1.00	1.00	1.00	1.00
Valid: Misclassification Rate	0.47	0.47	0.47	0.49	0.53
Valid: Mean Square Error	.	0.15	.	.	.
Valid: Sum of Frequencies	10692.00	10692.00	10692.00	10692.00	10692.00
Valid: Root Average Squared Error	0.38	0.38	0.39	0.39	0.40
Valid: Cumulative Percent Response	72.15	74.30	55.55	54.76	52.86
Valid: Percent Response	68.41	67.41	48.49	49.73	49.65
Valid: Root Mean Square Error	.	0.38	.	.	.

Valid: Cumulative Percent Response	72.15	74.30	55.55	54.76	52.86
Valid: Percent Response	68.41	67.41	48.49	49.73	49.65
Valid: Root Mean Square Error	.	0.38	.	.	.
Valid: Sum of Square Errors	6324.15	6204.71	6378.30	6419.58	6702.57
Valid: Sum of Case Weights Times Freq	.	42768.00	.	.	.
Valid: Number of Wrong Classifications	5041.00	.	.	.	.

Data Role=Test

Statistics	HPBNC	Reg	Tree3	Tree4	Tree
Test: Kolmogorov-Smirnov Statistic	0.51	0.52	0.48	0.48	0.38
Test: Average Squared Error	0.15	0.15	0.15	0.15	0.16
Test: Roc Index	0.83	0.84	0.78	0.79	0.75
Test: Average Error Function	.	0.51	.	.	.
Test: Bin-Based Two-Way Kolmogorov-Smirnov Probability Cutoff	0.34	0.31	0.34	0.39	0.39
Test: Cumulative Percent Captured Response	25.89	27.23	18.70	19.65	18.67
Test: Percent Captured Response	12.89	12.32	8.59	9.01	8.80
Test: Frequency of Classified Cases	3569.00	.	.	.	.
Test: Divisor for TASE	14276.00	14276.00	14276.00	14276.00	14276.00
Test: Error Function	.	7230.24	.	.	.
Test: Gain	158.79	172.23	86.94	96.47	86.65
Test: Gini Coefficient	0.66	0.67	0.56	0.58	0.50
Test: Bin-Based Two-Way Kolmogorov-Smirnov Statistic	0.51	0.52	0.48	0.47	0.38
Test: Kolmogorov-Smirnov Probability Cutoff	0.33	0.33	0.31	0.29	0.28
Test: Cumulative Lift	2.59	2.72	1.87	1.96	1.87
Test: Lift	2.59	2.47	1.72	1.81	1.76
Test: Maximum Absolute Error	1.00	1.00	1.00	1.00	1.00
Test: Misclassification Rate	0.49	0.47	0.52	0.51	0.54
Test: Lower 95% Conf. Limit for TMISC	.	0.46	.	.	.
Test: Upper 95% Conf. Limit for TMISC	.	0.49	.	.	.
Test: Mean Square Error	.	0.15	.	.	.
Test: Sum of Frequencies	3569.00	3569.00	3569.00	3569.00	3569.00
Test: Root Average Squared Error	0.39	0.38	0.39	0.39	0.40
Test: Cumulative Percent Response	73.67	77.50	53.22	55.93	53.14
Test: Percent Response	73.60	70.32	49.04	51.45	50.21

Test: Root Mean Square Error	.	0.38	.	.	.
Test: Sum of Square Errors	2163.57	2080.75	2211.56	2184.96	2260.19
Test: Sum of Case Weights Times Freq	.	14276.00	14276.00	14276.00	14276.00
Test: Number of Wrong Classifications	1759.00	.	.	.	.

Conclusion

From the output of the Model Comparison node, see that the Naive Bayes model is the best one for our data and to answer our data mining problem. Misclassification Rate was one of the measures that we considered to be important while evaluating the models, and Naive Bayes indeed has the least Misclassification Rate, thus the highest Accuracy Rate among the models we chose to compare and use. The screenshot further emphasizes that the Naive Bayes model had the least ASE, and therefore would be the best model for us to choose.

Selected Model	Model Node	Model Description	Valid: Misclassification Rate	Train: Average Squared Error	Train: Misclassification Rate	Valid: Average Squared Error
Y	HPBNC	HP BN Classifier	0.47147	0.13993	0.44322	0.14787
	Reg	Regression	0.47222	0.13772	0.43518	0.14508
	Tree3	Decision Tree	0.47306	0.13951	0.46454	0.14914
	Tree4	Decision Tree	0.49111	0.14519	0.48459	0.15010
	Tree	Decision Tree	0.53423	0.15651	0.52952	0.15672

References

[1] https://www.ncbi.nlm.nih.gov/pmc/articles/PMC6312200/

[2] https://www.conferenceboard.ca/hcp/provincial/health/nervous.aspx?AspxAutoDetectCookie Support=1

[3] https://www.who.int/mental_health/neurology/neurological_disorders_report_web.pdf

[4] https://www.who.int/gard/publications/The_Global_Impact_of_Respiratory_Disease.pdf

[5] https://www.ncbi.nlm.nih.gov/pmc/articles/PMC6354172/

[6] https://www.healthsystemtracker.org/chart-collection/mortality-rates-u-s-compare-countries/

[7] https://ourworldindata.org/burden-of-disease

[8] https://documentation.sas.com/?docsetId=emref&docsetTarget=n0cx4ud03paymdn1kargegadueml.htm&docsetVersion=14.3&locale=en#n1bj7zqor15ayen1rij0nhf0y1jt

[9] http://support.sas.com/documentation/cdl/en/vaug/68027/HTML/default/viewer.htm#n16w481g6eyvp2n1mjqi5f7xmgmi.htm