**HAP 780-001**

**MIMIC III DATASET EXPLORATION IN SQL AND WEKA**

Abhishek Shambhu

**Prediction of Mortality for Patients admitted in first 24 hours to hospital based on top 25 Lab Events**

**Abstract:**

We all know that the first thing which the doctors do when we are admitted to hospital is checkup where they do all different types of tests and based on the results further medications are suggested. This all tests are done in the initial few hours after the admission. So, it is absolutely necessary to deal with this. The main aim of this project is to predict mortality of patients admitted in hospitals and we focus on first 24 hours after the patient is admitted to hospital and choose just the top 25 most common lab events and to make a final .csv file which will be uploaded to Weka for making predictions.

**Dataset:**

The dataset which I worked on was the MIMIC-III dataset (<https://mimic.physionet.org/>) on which I did preprocessing in SQL using SQL Server 2016 Standard Edition. The MIMIC-III dataset was available after completing the CITI training. The dependent variable is whether the patient died or not when admitted to hospital and independent variables are all Lab events, the patients age, ethnicity and gender.

**Files used:**

1. Admissions Table
2. Patients Table
3. LabEvents Table
4. D\_LabItems Table

**Result:**

After running the final file in Weka and applying different models we can see that

-highest ROC showed up for Naïve Bayes Classifier and highest accuracy was for Decision Table.

**Keywords:**

MIMIC-III, mortality prediction, classification, SQL, Weka

--------------------------------------------------------------------------------------------------------------------

**Introduction:**

Nowadays, prediction of mortality is very crucial for hospitals at a very early stage based on the patient lab results. Moreover, it is very important to do predictions correct which will reduce the cost which will be used on patients which are going to die and can utilize on patients which can actually be survived. In our study, lab events will play a vital role in predictions as those will help us to predict patient mortality within one day of admission. Moreover, taking the 25 most common lab itemids will play a significant part in our study.

**Steps followed:**

1. After finishing the CITI training and getting access to data, the first step was to load all the necessary tables to SQL Server. For this, I had to change the datatypes for few columns, increasing the size so that it can be properly loaded.
2. Once all necessary tables are imported, I went forward with doing basic exploratory data analysis on Admissions and Patients data. Here is where I decided to work on Patient Mortality prediction.
3. First, I did random selection for one admission id for each patient and ignoring all multiple admission id thereby reducing complexity. Further, making Age and Ethnicity into 4 and 5 different categories: For Age – Neonate, Adult, Senior Adult and >89, whereas for Ethnicity – White, Black, Asian, Hispanic/Latino and Others (which do not fall in first 4 categories).
4. Next, I decided to choose LabEvents since it has all the Lab related results for patients which might help us in deciding whether the patient mortality had any effect based on lab item values.
5. Moreover, taking results for only the first 24 hours after the patient is admitted to hospital for predictions.
6. Next, taking only the top 25 most common Lab ItemId results from the first 24 hours after the patient is admitted. Labels associated with ItemId was obtained from D\_LabItems Table.
7. Taking the preprocessed LabEvents file and combining it with Combined Admissions Patients file and exporting it for Weka.
8. Finally, doing feature selection as well as splitting the data in Weka and applying different classification models to get the model summary. Finally, comparing the different models based on Accuracy, Precision, Recall and ROC Curve to see which model worked better.

**Final Variables used:**

* **From Admissions and Patients Table**

Gender, Age\_Category (After creating category based on Age column derived from DOB and AdmitTime), Ethnicity\_Category (using Like operator to get common categories and remaining all were made to Others category).

* **From Lab Events Table**

Hematocrit, Platelet\_Count, White\_Blood\_Cells, Hemoglobin, Mchc, Red\_Blood\_Cells,

Mch, Mcv, Rdw, Ph, Po2, Calculated\_Total\_Co2, Pco2, Potassium, Creatinine, Urea\_Nitrogen, Chloride, Neutrophils, Monocytes, Basophils, Eosinophils, Lymphocytes, Bicarbonate, Sodium, Anion\_Gap

**Data Preprocessing:**

1. Admissions Table

Preprocessing:

* Choosing for each Patient (SUBJECT\_ID), one Admission Id (HADM\_ID) randomly. Random numbers are generated based on Admission Id.

--- Preprocessing the Admissions table

--- so as to have only one random HADM\_ID for one SUBJECT\_ID.

--- Creating random number based on HADM\_ID. Since HADM\_ID is unique which will create unique random numbers.

SELECT [SUBJECT\_ID], [HADM\_ID], [ADMITTIME], [ETHNICITY], [DIAGNOSIS], [HOSPITAL\_EXPIRE\_FLAG], RAND([HADM\_ID]) AS RANDOM\_NO

INTO [mimic].[dbo].ADMISSIONS\_TEMP1

FROM [mimic].[dbo].[ADMISSIONS];

--- 58976 rows affected

* Assigning minimum of the random number value obtained to each subject id

--- Assigning Min random number to each subject\_id

SELECT [SUBJECT\_ID], MIN(RANDOM\_NO) AS MIN\_RANDOM\_NO

INTO [mimic].[dbo].ADMISSIONS\_TEMP2

FROM [mimic].[dbo].ADMISSIONS\_TEMP1

GROUP BY [SUBJECT\_ID];

--- 46520 rows affected

* Combining so as to get the final table. Also, doing preprocessing such as removing all records from diagnosis table which are null and only having outcome as 0 & 1 in Hospital Expire Flag Table.

--- Combining the two tables based on Min Random Number and performing cleaning of [HOSPITAL\_EXPIRE\_FLAG] and [DIAGNOSIS]

SELECT T1.[SUBJECT\_ID], [HADM\_ID], [ADMITTIME], [ETHNICITY], [DIAGNOSIS], [HOSPITAL\_EXPIRE\_FLAG]

INTO [mimic].[dbo].ADMISSIONS\_PROCESSED

FROM [mimic].[dbo].ADMISSIONS\_TEMP1 T1

INNER JOIN

[mimic].[dbo].ADMISSIONS\_TEMP2 T2

ON RANDOM\_NO=MIN\_RANDOM\_NO

WHERE [HOSPITAL\_EXPIRE\_FLAG] = '0' OR [HOSPITAL\_EXPIRE\_FLAG] = '1' AND [DIAGNOSIS] <> ''; --- 45472 rows

---

* Finally having a look at the Mortality count

--- Getting count of Mortality from Admissions Table

SELECT COUNT(\*)

FROM [mimic].[dbo].ADMISSIONS\_PROCESSED

WHERE [HOSPITAL\_EXPIRE\_FLAG] = 1 --- 4815

1. Patients Table

* Select only the necessary attributes from Patients table. No preprocessing was needed here.

---2. Patients table

SELECT [SUBJECT\_ID], [GENDER], [DOB], [EXPIRE\_FLAG]

FROM [mimic].[dbo].PATIENTS; --- 46520 rows

1. Admissions + Patients Table

* Combining two tables based on Left Join since Admissions Processed table is cleaned and has fewer number of rows.

---3. Joining Processed Admissions Table with the Patients Table

SELECT ap.[SUBJECT\_ID], [HADM\_ID], [ADMITTIME], [ETHNICITY], [DIAGNOSIS], [HOSPITAL\_EXPIRE\_FLAG], [GENDER], [DOB], [EXPIRE\_FLAG]

INTO [mimic].[dbo].COMB\_ADM\_PAT\_RAW1

FROM [mimic].[dbo].[ADMISSIONS\_PROCESSED] ap

LEFT JOIN

[mimic].[dbo].PATIENTS p

ON ap.SUBJECT\_ID = p.SUBJECT\_ID; --- 45472 rows

* Going forward to do a little more preprocessing, I had ADMITTIME and DOB in varchar. So, used CAST function so as to make it to datetime.

---4. Since we have DOB and ADMITTIME, we calculate the Age

--- Converting String to Datetime type for dates

SELECT [SUBJECT\_ID], [HADM\_ID], ADMITTIME, CAST(ADMITTIME AS datetime) AS ADMITTIME\_DATE, [ETHNICITY], [DIAGNOSIS],

[HOSPITAL\_EXPIRE\_FLAG], [GENDER], DOB, CAST(DOB AS datetime) AS DOB\_DATE, [EXPIRE\_FLAG]

INTO [mimic].[dbo].COMB\_ADM\_PAT\_RAW2

FROM [mimic].[dbo].COMB\_ADM\_PAT\_RAW1; --- 45472 rows

* The reason for doing the above step is to calculate Age of the patient at the time of admission based on difference between ADMITTIME and DOB.

--- Taking difference for calculating age --- Here I am avoiding mm and dd complexity for age and just using year

SELECT [SUBJECT\_ID], [HADM\_ID], ADMITTIME\_DATE, [ETHNICITY], [DIAGNOSIS],

[HOSPITAL\_EXPIRE\_FLAG], [GENDER], DOB, [EXPIRE\_FLAG], (DATEPART(year, ADMITTIME\_DATE) - DATEPART(year, DOB\_DATE)) as AGE

INTO [mimic].[dbo].COMB\_ADM\_PAT

FROM [mimic].[dbo].COMB\_ADM\_PAT\_RAW2; --- 45472 rows

* The Age column here needed some preprocessing as there were few patient records having age in 300’s. Also, all the Age’s were made into category of 4: Neonate, Adult, Senior Adult and >89.

--- Now since we have Age we go forward to assign classes to Age (Age category)

--- 0-1 --> Neonate 15-59 --> Adult 60-89 --> Senior Adult 89+ --> >89

SELECT [SUBJECT\_ID], [HADM\_ID], ADMITTIME\_DATE, [ETHNICITY], [DIAGNOSIS], [HOSPITAL\_EXPIRE\_FLAG], [GENDER], DOB, [EXPIRE\_FLAG], [AGE]

, CASE

-- all ages > 89 in the database were replaced with 300

WHEN [AGE] <= 1

THEN 'NEONATE' --- 7874 rows

WHEN [AGE] > 14 AND [AGE] <=59

THEN 'ADULT' --- 14017 rows

WHEN [AGE] > 59 AND [AGE] <=89

THEN 'SENIOR ADULT' --- 21645 rows

ELSE '>89' --- 1936 rows

END AS AGE\_CATEGORY

INTO [mimic].[dbo].COMB\_ADM\_PAT\_AGECAT

FROM [mimic].[dbo].COMB\_ADM\_PAT

ORDER BY [SUBJECT\_ID]; --- 45472 rows

* Similar to the Age column, Ethnicity column also needed some preprocessing. It had around 41 different ethnicity categories out of which few are of same category but the mentioning way was different. So, I used the LIKE function to group those ethnicity classes and the one which did not lay in all of this ethnicity group were kept into the OTHERS group.

--- Reducing 41 Ethnicity classes to top 4 classes

SELECT [SUBJECT\_ID], [HADM\_ID], [GENDER], [DOB], [ADMITTIME\_DATE], [AGE], [AGE\_CATEGORY],

CASE WHEN lower(ethnicity) like '%white%'

THEN 'WHITE'

WHEN lower(ethnicity) like '%black%'

THEN 'BLACK'

WHEN lower(ethnicity) like '%asian%'

THEN 'ASIAN'

WHEN lower(ethnicity) like '%hispanic%'

THEN 'HISPANIC/LATINO'

ELSE 'OTHERS'

END AS ETHNICITY\_CATEGORY,

[HOSPITAL\_EXPIRE\_FLAG], [EXPIRE\_FLAG]

INTO [mimic].[dbo].COMB\_ADM\_PAT\_ETHCAT

FROM [mimic].[dbo].COMB\_ADM\_PAT\_AGECAT; --- 45472 rows affected

Even went forward by creating dummy variables for all gender, age and ethnicity category but since not much change in final models so dropped.

1. LabEvents Table

--- Taking Lab events of patients

SELECT \* FROM [mimic].[dbo].[LABEVENTS] --- 27854055 rows affected

Even went forward with using one percent of LabEvents table by using **newid**() function and **tablesample** but later dropped.

Preprocessing:

* First, since my LabEvents data had double quotes and since the file was big enough to not to get opened completely in notepad. I had to load to SQL Server with the double quotes. So, used the below queries to remove quotes.

--- Preprocessing LabEvents data

--- To remove quotes

SET QUOTED\_IDENTIFIER OFF

UPDATE [mimic].[dbo].[LABEVENTS]

SET [VALUEUOM] = REPLACE([VALUEUOM],"""","")

SET QUOTED\_IDENTIFIER ON --- 27854055 rows affected

SET QUOTED\_IDENTIFIER OFF

UPDATE [mimic].[dbo].[LABEVENTS]

SET VALUENUM = REPLACE(VALUENUM,"'","")

SET QUOTED\_IDENTIFIER ON --- 27854055 rows affected

* Now, since we are considering only patients which are in hospital (in patient data) we removed patients from LabEvents whose HADMID was null, also removing negative and null rows from Valuenum and ValueUOM columns so as to make the data as clean as we can do. We can even do imputation here, but since we already have a large number of records, we exclude them.

--- Considering only in patient data(Removing all nulls from HADM\_ID & nulls from Valuenum column)

---Removing all negative valuenum and preparing final Labevents table

select [SUBJECT\_ID], [HADM\_ID], [ITEMID], CAST([CHARTTIME] AS datetime) AS CHARTTIME\_DATE, [VALUENUM], [VALUEUOM]

INTO [mimic].[dbo].LABEVENTS\_PROCESSED

FROM [mimic].[dbo].[LABEVENTS]

WHERE HADM\_ID <> '' AND VALUENUM not like '%-%' AND [VALUENUM] <> '' AND [VALUEUOM] <> ''; --- 19041736 rows

1. LabEvents within first 24 hours of admission

* Taking LabEvents only for first 24 hours of admission. i.e. labevents recorded within 24 hours from the time of admission.

select a.[SUBJECT\_ID], a.[HADM\_ID], [GENDER], [DOB], [ADMITTIME\_DATE], [AGE], [AGE\_CATEGORY], [ETHNICITY\_CATEGORY],

[DIAGNOSIS], [HOSPITAL\_EXPIRE\_FLAG], [EXPIRE\_FLAG], ITEMID, CHARTTIME\_DATE, VALUENUM, VALUEUOM

into [mimic].[dbo].LAB\_RESULT\_WITHIN\_24H

from [mimic].[dbo].COMB\_ADM\_PAT\_ETHCAT a RIGHT JOIN [mimic].[dbo].LABEVENTS\_PROCESSED l on l.hadm\_id = a.hadm\_id

where (DATEPART (day,CHARTTIME\_DATE-ADMITTIME\_DATE)\*24 + DATEPART (hour,CHARTTIME\_DATE-ADMITTIME\_DATE))<='24'

and (DATEPART (day,CHARTTIME\_DATE-ADMITTIME\_DATE)\*24 + DATEPART (hour,CHARTTIME\_DATE-ADMITTIME\_DATE)) >='0'; ---- 24 hr --- 194583 rows

* After getting the necessary lab events, I went forward to look for total unique itemids.

---- Distinct ITEM\_ID

select [ITEMID], count([ITEMID]) from [mimic].[dbo].LAB\_RESULT\_WITHIN\_24H group by [ITEMID] ORDER BY count([ITEMID]) desc ---- 289 rows

* Here, since we have 289 unique itemids we would have 289 unique columns for different Lab Events. This would make our Weka file to have nearly 300 independent attributes. So, I went forward to select the top 25 most common ItemIds.

---- Distinct ITEM\_ID

select TOP 25 [ITEMID], count([ITEMID]) as count INTO [mimic].[dbo].temp1 from [mimic].[dbo].LAB\_RESULT\_WITHIN\_24H group by [ITEMID] ORDER BY count([ITEMID]) desc ---- 25 rows

--- Combining with D\_LabItems Table to get Labtest labels.

Select d.ITEMID, t.count, d.[LABEL]

from [mimic].[dbo].D\_LABITEMS d

INNER JOIN

[mimic].[dbo].temp1 t

ON t.ITEMID = d.ITEMID

ORDER BY t.count desc; --- 25 rows

* Below is the list of top 25 most common tests conducted in the first 24 hours of admission.

Hematocrit, Platelet\_Count, White\_Blood\_Cells, Hemoglobin, Mchc, Red\_Blood\_Cells, Mch, Mcv, Rdw, Ph, Po2, Calculated\_Total\_Co2, Pco2, Potassium, Creatinine, Urea\_Nitrogen, Chloride, Neutrophils, Monocytes, Basophils, Eosinophils, Lymphocytes, Bicarbonate, Sodium, Anion\_Gap

* Total records based on top 25 Lab events

--- Number of records for top 25 itemids in Labresults table

Select \* INTO [mimic].[dbo].SELECT\_ITEMID\_24H

from [mimic].[dbo].LAB\_RESULT\_WITHIN\_24H where ITEMID IN (51221,51265,51301,51222,51249,51279,51248,51250,51277,50820,

50821,50804,50818,50971,50912,51006,50902,51256,51254,51146,51200,51244,50882,50983,50868); ---114230 rows

* Total records based on top 25 Lab events

--- Select top 25 most common item ids, getting label from D\_LABITEMS and generating columns based on LabItemId table

select [HADM\_ID],

avg(CASE WHEN [ITEMID]=51221 THEN CAST(VALUENUM as DECIMAL(9,2)) ELSE null END) as HEMATOCRIT,

avg(CASE WHEN [ITEMID]=51265 THEN CAST(VALUENUM as DECIMAL(9,2)) ELSE null END) as PLATELET\_COUNT,

avg(CASE WHEN [ITEMID]=51301 THEN CAST(VALUENUM as DECIMAL(9,2)) ELSE null END) as WHITE\_BLOOD\_CELLS,

avg(CASE WHEN [ITEMID]=51222 THEN CAST(VALUENUM as DECIMAL(9,2)) ELSE null END) as HEMOGLOBIN,

avg(CASE WHEN [ITEMID]=51249 THEN CAST(VALUENUM as DECIMAL(9,2)) ELSE null END) as MCHC,

avg(CASE WHEN [ITEMID]=51279 THEN CAST(VALUENUM as DECIMAL(9,2)) ELSE null END) as RED\_BLOOD\_CELLS,

avg(CASE WHEN [ITEMID]=51248 THEN CAST(VALUENUM as DECIMAL(9,2)) ELSE null END) as MCH,

avg(CASE WHEN [ITEMID]=51250 THEN CAST(VALUENUM as DECIMAL(9,2)) ELSE null END) as MCV,

avg(CASE WHEN [ITEMID]=51277 THEN CAST(VALUENUM as DECIMAL(9,2)) ELSE null END) as RDW,

avg(CASE WHEN [ITEMID]=50820 THEN CAST(VALUENUM as DECIMAL(9,2)) ELSE null END) as pH,

avg(CASE WHEN [ITEMID]=50821 THEN CAST(VALUENUM as DECIMAL(9,2)) ELSE null END) as pO2,

avg(CASE WHEN [ITEMID]=50804 THEN CAST(VALUENUM as DECIMAL(9,2)) ELSE null END) as CALCULATED\_TOTAL\_CO2,

avg(CASE WHEN [ITEMID]=50818 THEN CAST(VALUENUM as DECIMAL(9,2)) ELSE null END) as pCO2,

avg(CASE WHEN [ITEMID]=50971 THEN CAST(VALUENUM as DECIMAL(9,2)) ELSE null END) as POTASSIUM,

avg(CASE WHEN [ITEMID]=50912 THEN CAST(VALUENUM as DECIMAL(9,2)) ELSE null END) as CREATININE,

avg(CASE WHEN [ITEMID]=51006 THEN CAST(VALUENUM as DECIMAL(9,2)) ELSE null END) as UREA\_NITROGEN,

avg(CASE WHEN [ITEMID]=50902 THEN CAST(VALUENUM as DECIMAL(9,2)) ELSE null END) as CHLORIDE,

avg(CASE WHEN [ITEMID]=51256 THEN CAST(VALUENUM as DECIMAL(9,2)) ELSE null END) as NEUTROPHILS,

avg(CASE WHEN [ITEMID]=51254 THEN CAST(VALUENUM as DECIMAL(9,2)) ELSE null END) as MONOCYTES,

avg(CASE WHEN [ITEMID]=51146 THEN CAST(VALUENUM as DECIMAL(9,2)) ELSE null END) as BASOPHILS,

avg(CASE WHEN [ITEMID]=51200 THEN CAST(VALUENUM as DECIMAL(9,2)) ELSE null END) as EOSINOPHILS,

avg(CASE WHEN [ITEMID]=51244 THEN CAST(VALUENUM as DECIMAL(9,2)) ELSE null END) as LYMPHOCYTES,

avg(CASE WHEN [ITEMID]=50882 THEN CAST(VALUENUM as DECIMAL(9,2)) ELSE null END) as BICARBONATE,

avg(CASE WHEN [ITEMID]=50983 THEN CAST(VALUENUM as DECIMAL(9,2)) ELSE null END) as SODIUM,

avg(CASE WHEN [ITEMID]=50868 THEN CAST(VALUENUM as DECIMAL(9,2)) ELSE null END) as ANION\_GAP

into [mimic].[dbo].LAB\_RESULT\_WITHIN\_24H\_PIVOT

from [mimic].[dbo].SELECT\_ITEMID\_24H group by [HADM\_ID] ---- 9677 rows

* Count of records from two files that needs to be combined to make a final Weka file

Select \* from [mimic].[dbo].COMB\_ADM\_PAT\_ETHCAT; ----- 45472 rows

SELECT \* FROM [mimic].[dbo].LAB\_RESULT\_WITHIN\_24H\_PIVOT; ---- 9677 rows

* Count of records from two files that needs to be combined to make a final Weka file

---- Performing join on two files to make a single final csv file that can be dumped in Weka

select c.\* , HEMATOCRIT, PLATELET\_COUNT, WHITE\_BLOOD\_CELLS, HEMOGLOBIN, MCHC, RED\_BLOOD\_CELLS,

MCH, MCV, RDW, pH, pO2, CALCULATED\_TOTAL\_CO2, pCO2, POTASSIUM, CREATININE, UREA\_NITROGEN, CHLORIDE,

NEUTROPHILS, MONOCYTES, BASOPHILS, EOSINOPHILS, LYMPHOCYTES, BICARBONATE, SODIUM, ANION\_GAP

---INTO [mimic].[dbo].FINAL\_TO\_WEKA

from [mimic].[dbo].COMB\_ADM\_PAT\_ETHCAT c

INNER JOIN

[mimic].[dbo].LAB\_RESULT\_WITHIN\_24H\_PIVOT l on c.hadm\_id=l.hadm\_id; ---- 9677 rows

---- View of the final file that would be loaded to Weka

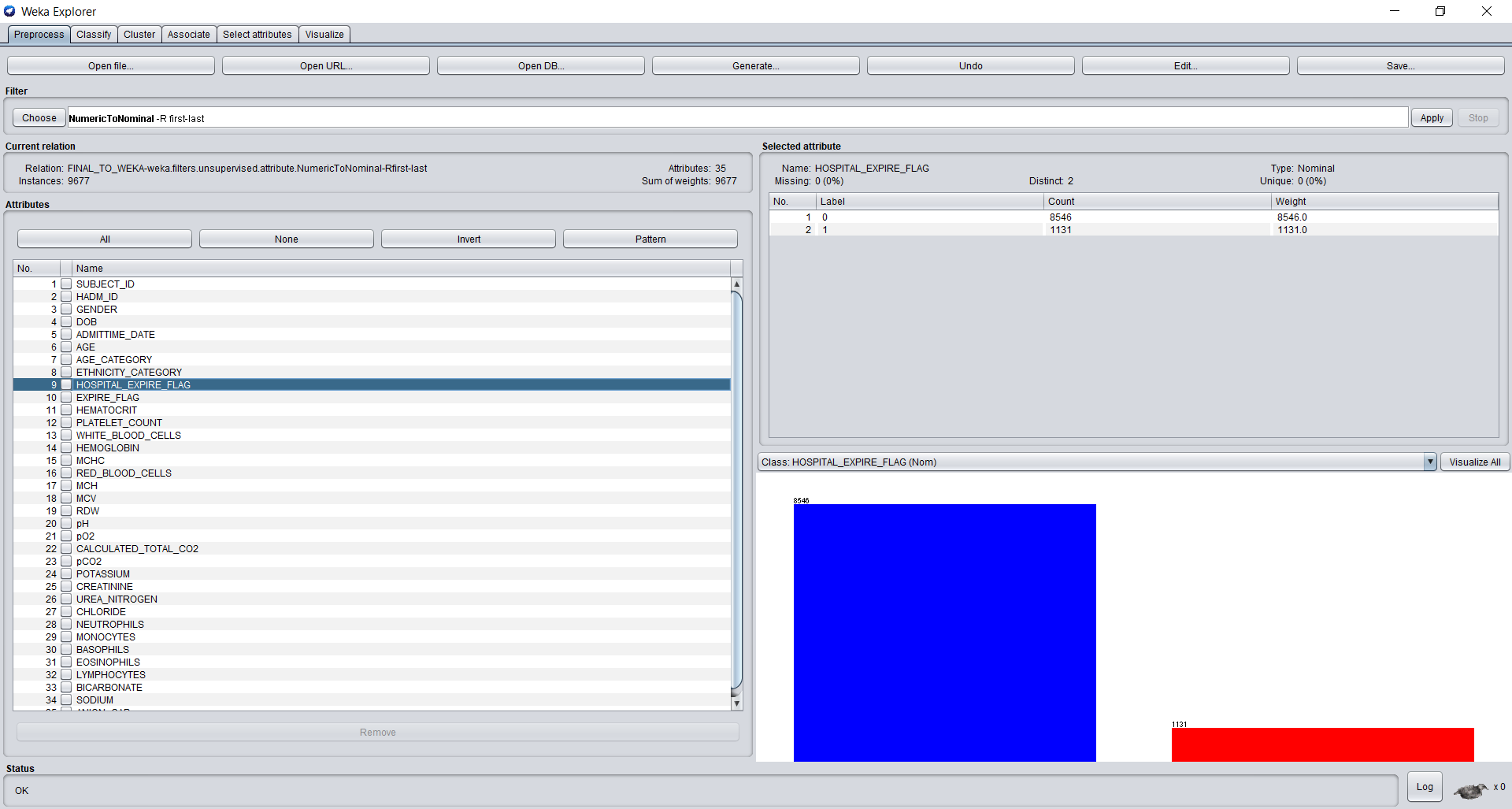
Select \* from [mimic].[dbo].FINAL\_TO\_WEKA; ---- 9677 rows

**Weka:**

1. **Feature Selection**

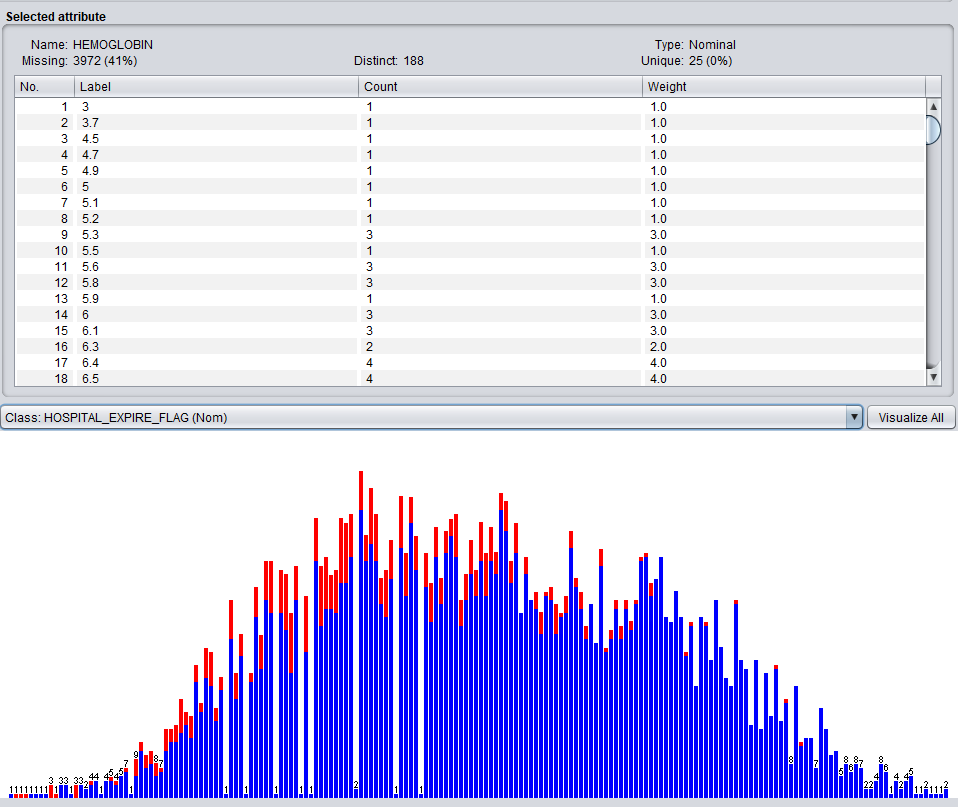
Feature selection plays a vital role in selecting only the necessary attributes which will have impact on the outcome variable. In our case, it’s the prediction of mortality in hospitals based on Lab Events and other attributes such as Age, Ethnicity and Gender.

First, we will be loading the exported final file to Weka with 35 attributes. Next, we will do Numeric to Nominal so as to get in the below format:

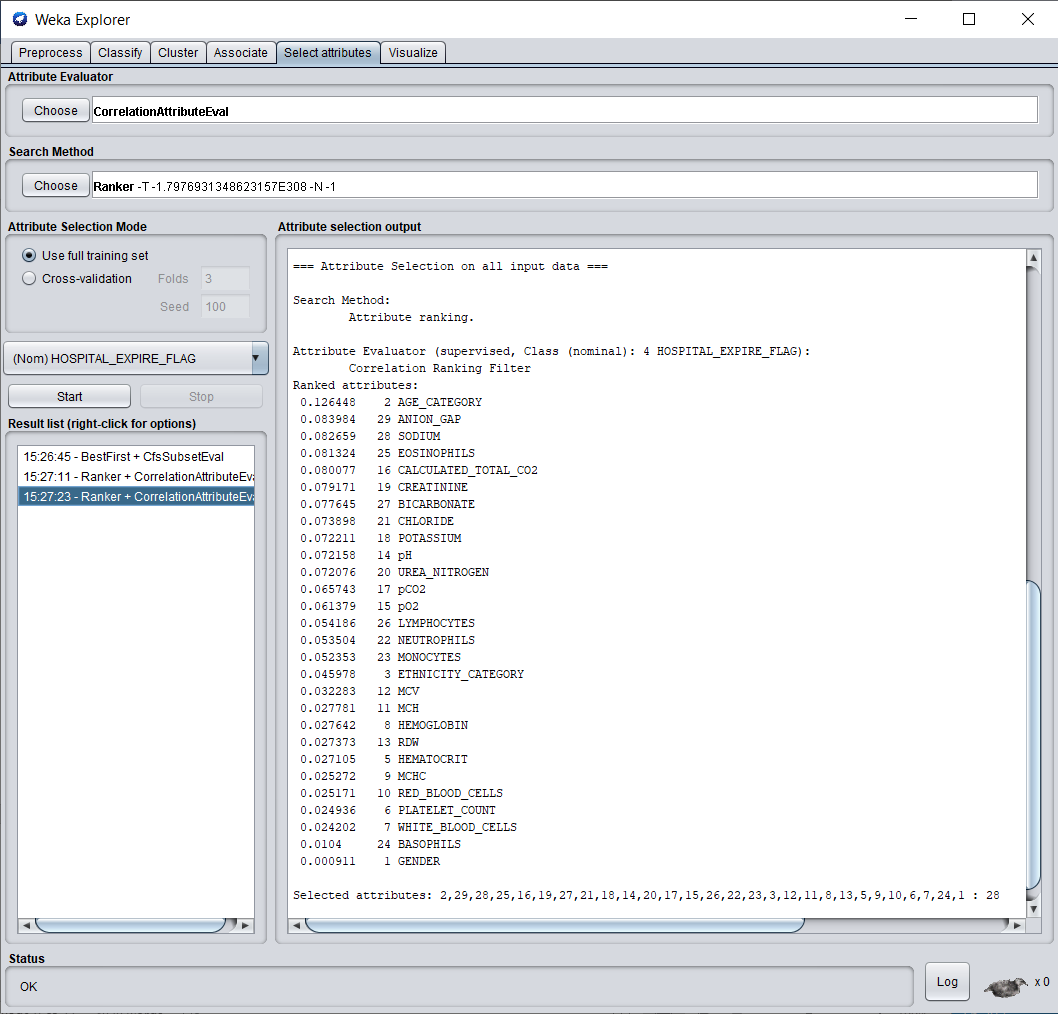


Now, we will remove all attributes which are not important in predicting mortality of patients in hospitals such as SUBJECT\_ID, HADM\_ID, DOB, ADMITTIME\_DATE, AGE and EXPIRE\_FLAG. So, we are left with 29 attributes now.

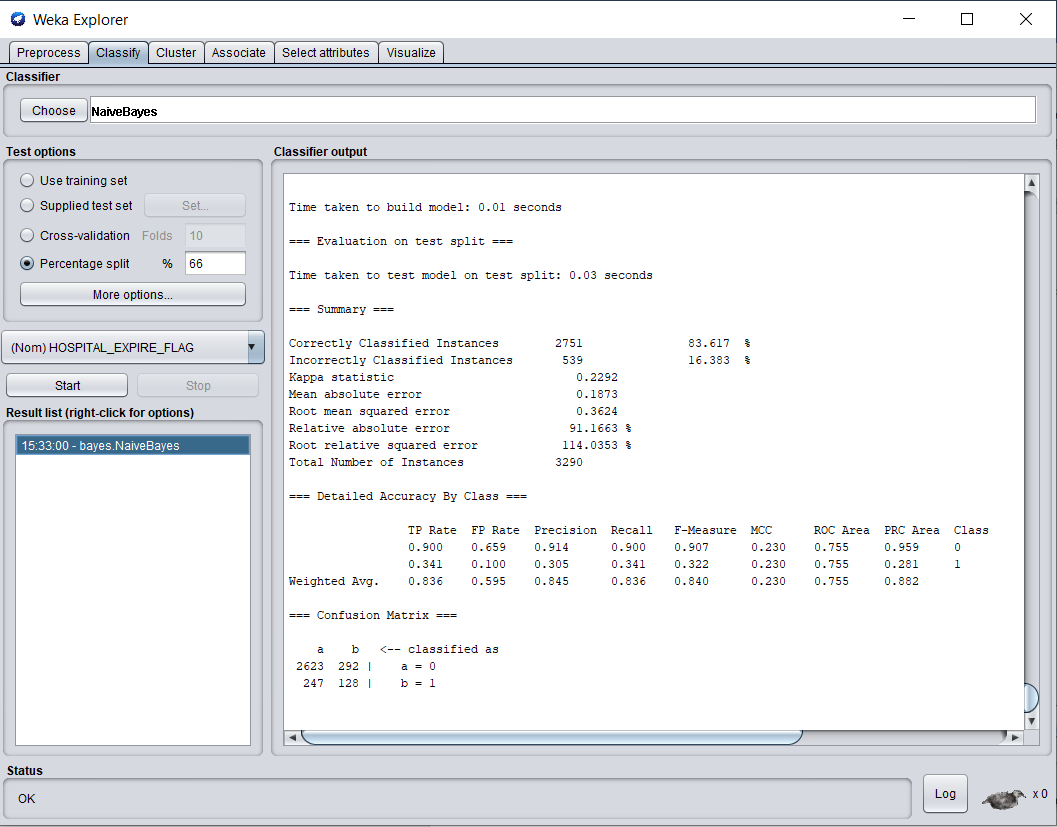
One of the attributes which I came across was Hemoglobin whose graph is as follows. Here, we can see that low Hemoglobin has higher chances of deaths. The value for Haemoglobin ranges from 3 to 23.8. Similarly, we can have a look for effect of all the other independent variables on the outcome variable.



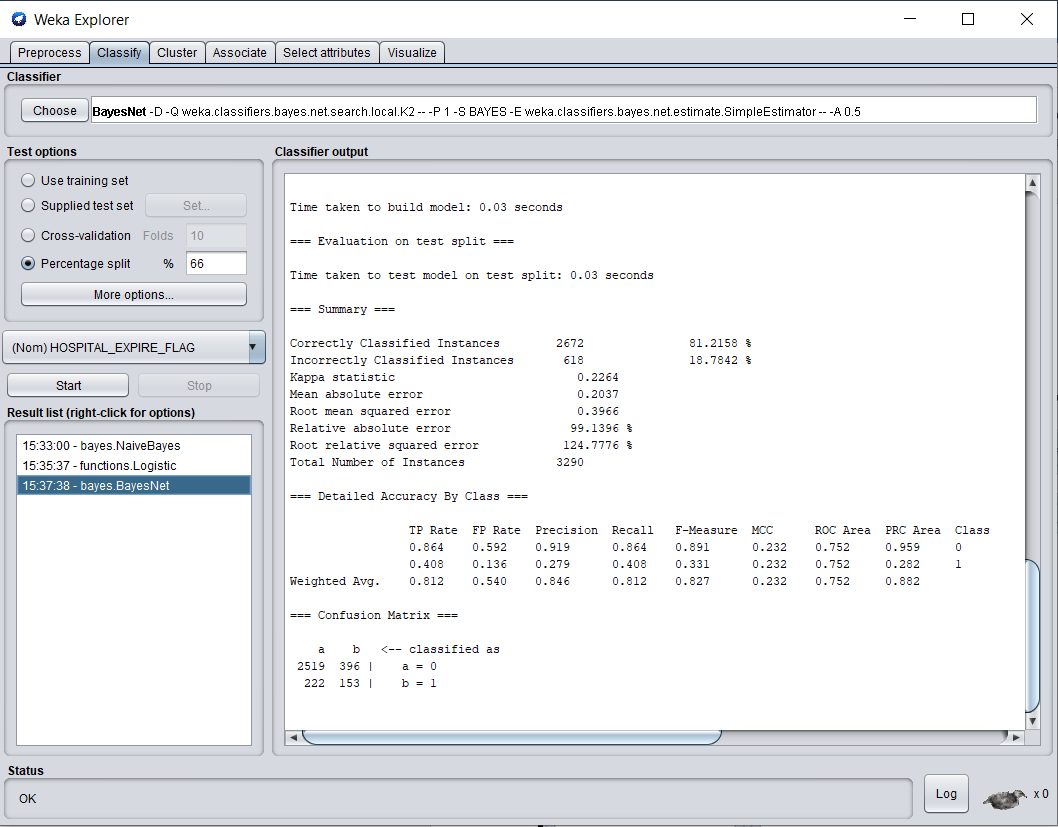
* Using Attribute Evaluator method ‘Correlation Attribute Eval’ and Ranker Search Method on Full Training Set.



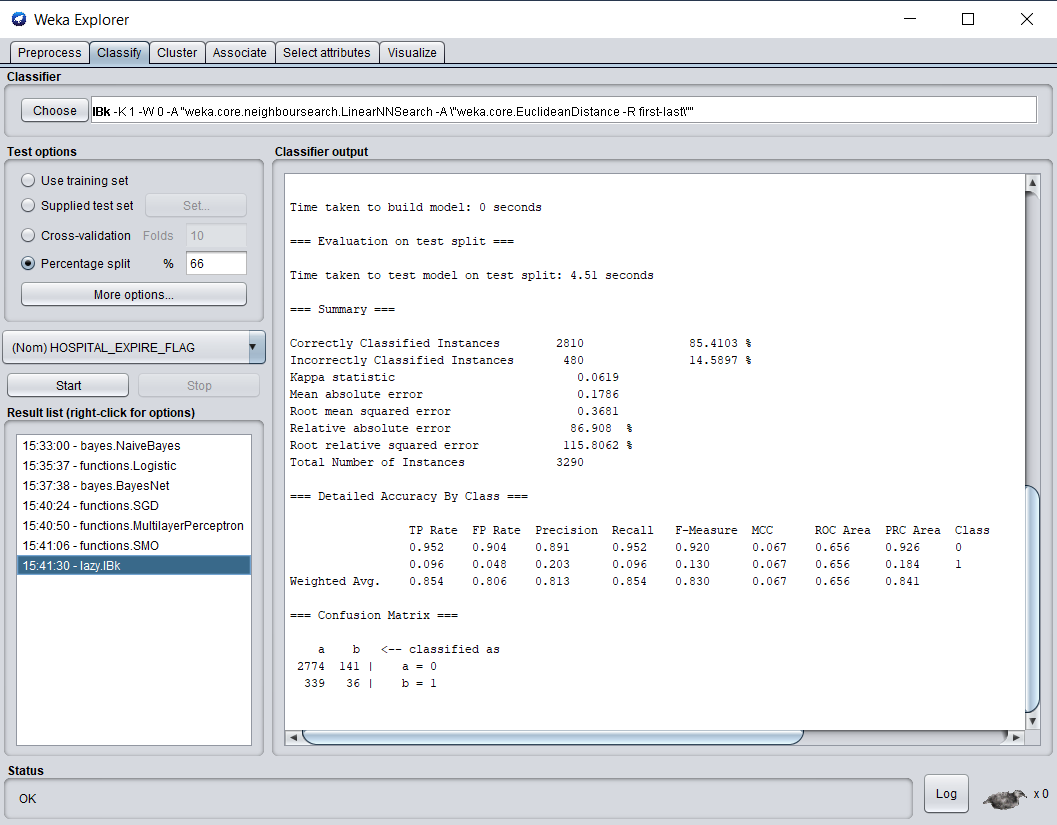
1. **Applying Classification Models to the dataset based on attributes selected –**
2. Naïve Bayes



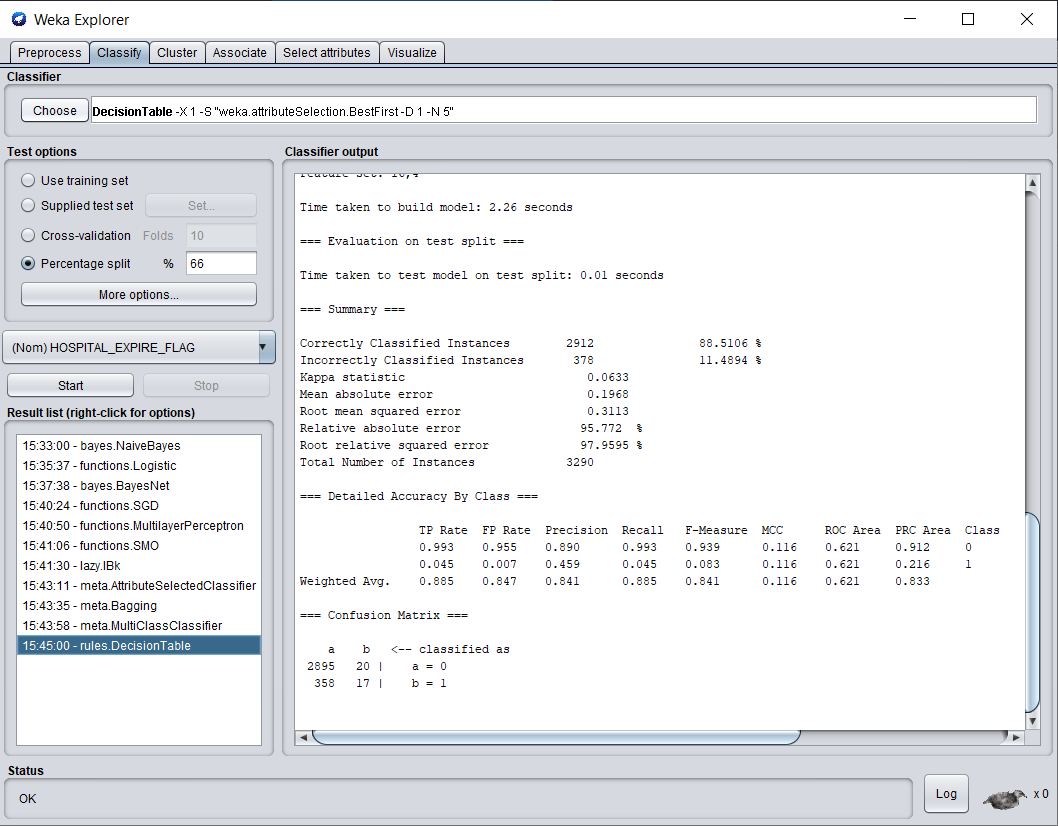
1. Bayes Net



1. K-Nearest Neighbors (lbk)



1. Decision Table



**Summary Comparison of 4 models:**

We used the Weka default splitting of 66% on building the models. We can see that Naïve Bayes model has the highest ROC area of 0.755 whereas Decision Table performed the worst in all our models having ROC area of 0.621. This were the improved results after performing attribute selection method.

|  |  |  |
| --- | --- | --- |
| Models | Accuracy | ROC |
| Naïve Bayes | 83.617 % | 0.755 |
| Bayes Net | 81.2158 % | 0.752 |
| K-nearest neighbors | 85.4103 % | 0.656 |
| Decision Table | 88.5106 % | 0.621 |

**Future Work:**

We can try to take chartevents into consideration and combine it with our labevents so as to make better predictions. Also, considering the length of icu stay for a patient based on admission time and discharge would be useful to make mortality predictions and to see if that can make any significant improvement to our existing models. This results would be useful for medical practitioners, doctors in mortality predictions and for data scientists for the study of healthcare data.

**Conclusion:**

The above study would be beneficial and can be used as a second opinion by both medical practitioners and doctors to accurately predict mortality which will help to save resources as well as efforts spent trying to save mortality patients.

**References:**

[1] The Laboratory for Computational Physiology, MIT. “MIMIC Critical Care Database.” MIMIC Critical Care Database. Accessed September 23, 2019. <https://mimic.physionet.org/>.

[2] Prediction of patients’ mortality during hospitalizations. Accessed from George Mason University Blackboard.