CSE 6339 SPECIAL TOPICS IN ADVANCED DATABASE SYSTEMS

PROJECT 1

GROUP 107

|  |  |
| --- | --- |
| DHRUV PRAJAPATI | 1001051824 |
| GURKAMAL DEEP SINGH RAKHRA | 1001049557 |
| NAMRATHA SURYANARAYANA IYER | 1001112730 |
| PUNEETH UMESH BHARADWAJ | 1001106478 |

[Task 1. 3](#_Toc237744600)

*[1.](#_Toc104770248)* [The most common disease for each age group. (Code task11.py) 3](#_Toc104770248)

*[2.](#_Toc1932737419)* [In Hospital Mortality 20](#_Toc1932737419)

*[3.](#_Toc1048509170)* [Demographics 21](#_Toc1048509170)

*[4.](#_Toc296783645)* [Effect of Length of Stay 25](#_Toc296783645)

[Task 2. 29](#_Toc2043307682)

*[1.](#_Toc722035855)* [Relational Schema 29](#_Toc722035855)

*[2.](#_Toc1364804375)* [Import Data 32](#_Toc1364804375)

*[3.](#_Toc922781088)* [SQL queries 34](#_Toc922781088)

[a. Query to get back the original csv file from the DB 34](#_Toc1527735411)

[b. Coverage Ratio 35](#_Toc977621648)

[c. Variation and Length of Stay 36](#_Toc1653983475)

[d. DRG\_PRICE vs TOTAL\_CHARGES 38](#_Toc889915329)

[Task 3. 39](#_Toc696812166)

*[A.](#_Toc1037841278)* [Appropriate Number of Clusters 39](#_Toc1037841278)

*[B.](#_Toc1504966059)* [Calculate Number of Clusters 40](#_Toc1504966059)

*[C.](#_Toc200369354)* [Interesting Profiles 41](#_Toc200369354)

*[D.](#_Toc1892372674)* [Supervised vs Unsupervised 41](#_Toc1892372674)

[Task 4. 42](#_Toc951648190)

*[a.](#_Toc1811019426)* [Method to find DRG\_PRICE\_BINARY 42](#_Toc1811019426)

*[b.](#_Toc103954346)* [Clinicians/Administrators know the following details 42](#_Toc103954346)

*[c.](#_Toc741055048)* [Scenarios 43](#_Toc741055048)

[1. Exclude 7, 10 and 11. 43](#_Toc493868046)

[2. CfsSubsetEval 43](#_Toc1944700404)

[3. Classifiers 45](#_Toc1726514803)

[4. Accuracy 49](#_Toc1811397341)

[a. Overall Accuracy 49](#_Toc1061892312)

[b. Greater than $80000 49](#_Toc1110532421)

[c. Less than $80000 50](#_Toc948384044)

# Task 1.

## The most common disease for each age group. (Code task11.py)

SQL Query

(SELECT

AGE, DIAGNOSIS\_CODE, COUNT(DIAGNOSIS\_CODE) AS T

FROM

`disease`

WHERE

AGE = '1' AND DIAGNOSIS\_CODE != 'NULL'

GROUP BY DIAGNOSIS\_CODE

ORDER BY T DESC

LIMIT 1) UNION (SELECT

AGE, DIAGNOSIS\_CODE, COUNT(DIAGNOSIS\_CODE) AS T

FROM

`disease`

WHERE

AGE = '2' AND DIAGNOSIS\_CODE != 'NULL'

GROUP BY DIAGNOSIS\_CODE

ORDER BY T DESC

LIMIT 1) UNION (SELECT

AGE, DIAGNOSIS\_CODE, COUNT(DIAGNOSIS\_CODE) AS T

FROM

`disease`

WHERE

AGE = '3' AND DIAGNOSIS\_CODE != 'NULL'

GROUP BY DIAGNOSIS\_CODE

ORDER BY T DESC

LIMIT 1) UNION (SELECT

AGE, DIAGNOSIS\_CODE, COUNT(DIAGNOSIS\_CODE) AS T

FROM

`disease`

WHERE

AGE = '4' AND DIAGNOSIS\_CODE != 'NULL'

GROUP BY DIAGNOSIS\_CODE

ORDER BY T DESC

LIMIT 1) UNION (SELECT

AGE, DIAGNOSIS\_CODE, COUNT(DIAGNOSIS\_CODE) AS T

FROM

`disease`

WHERE

AGE = '5' AND DIAGNOSIS\_CODE != 'NULL'

GROUP BY DIAGNOSIS\_CODE

ORDER BY T DESC

LIMIT 1) UNION (SELECT

AGE, DIAGNOSIS\_CODE, COUNT(DIAGNOSIS\_CODE) AS T

FROM

`disease`

WHERE

AGE = '6' AND DIAGNOSIS\_CODE != 'NULL'

GROUP BY DIAGNOSIS\_CODE

ORDER BY T DESC

LIMIT 1) UNION (SELECT

AGE, DIAGNOSIS\_CODE, COUNT(DIAGNOSIS\_CODE) AS T

FROM

`disease`

WHERE

AGE = '7' AND DIAGNOSIS\_CODE != 'NULL'

GROUP BY DIAGNOSIS\_CODE

ORDER BY T DESC

LIMIT 1) UNION (SELECT

AGE, DIAGNOSIS\_CODE, COUNT(DIAGNOSIS\_CODE) AS T

FROM

`disease`

WHERE

AGE = '8' AND DIAGNOSIS\_CODE != 'NULL'

GROUP BY DIAGNOSIS\_CODE

ORDER BY T DESC

LIMIT 1) UNION (SELECT

AGE, DIAGNOSIS\_CODE, COUNT(DIAGNOSIS\_CODE) AS T

FROM

`disease`

WHERE

AGE = '9' AND DIAGNOSIS\_CODE != 'NULL'

GROUP BY DIAGNOSIS\_CODE

ORDER BY T DESC

LIMIT 1)

|  |  |
| --- | --- |
| Age | Diagnosis |
| 1 | 2153 |
| 2 | V6284 |
| 3 | 486 |
| 4 | 5856 |
| 5 | 486 |
| 6 | 486 |
| 7 | 486 |
| 8 | 5990 |
| 9 | 5849 |

Prevelance

SQL Query

SELECT

TOTAL.AGE,

TOTAL.T,

T1.DIAGNOSIS\_CODE,

T1.T,

(T1.T / TOTAL.T) AS FIRST\_COUNT,

T2.DIAGNOSIS\_CODE,

T2.T,

(T2.T / TOTAL.T) AS SECOND\_COUNT,

T3.DIAGNOSIS\_CODE,

T3.T,

(T3.T / TOTAL.T) AS THIRD\_COUNT

FROM

(SELECT

AGE, COUNT(DIAGNOSIS\_CODE) AS T

FROM

`DISEASE`

WHERE

AGE = '1') AS TOTAL,

(SELECT

DIAGNOSIS\_CODE, COUNT(DIAGNOSIS\_CODE) AS T

FROM

`disease`

WHERE

AGE = '1' AND DIAGNOSIS\_CODE != 'NULL'

GROUP BY DIAGNOSIS\_CODE

ORDER BY T DESC

LIMIT 1) AS T1,

(SELECT

DIAGNOSIS\_CODE, COUNT(DIAGNOSIS\_CODE) AS T

FROM

`disease`

WHERE

AGE = '1' AND DIAGNOSIS\_CODE != 'NULL'

GROUP BY DIAGNOSIS\_CODE

ORDER BY T DESC

LIMIT 1 OFFSET 1) AS T2,

(SELECT

DIAGNOSIS\_CODE, COUNT(DIAGNOSIS\_CODE) AS T

FROM

`disease`

WHERE

AGE = '1' AND DIAGNOSIS\_CODE != 'NULL'

GROUP BY DIAGNOSIS\_CODE

ORDER BY T DESC

LIMIT 1 OFFSET 2) AS T3

UNION SELECT

TOTAL.AGE,

TOTAL.T,

T1.DIAGNOSIS\_CODE,

T1.T,

(T1.T / TOTAL.T) AS FIRST\_COUNT,

T2.DIAGNOSIS\_CODE,

T2.T,

(T2.T / TOTAL.T) AS SECOND\_COUNT,

T3.DIAGNOSIS\_CODE,

T3.T,

(T3.T / TOTAL.T) AS THIRD\_COUNT

FROM

(SELECT

AGE, COUNT(DIAGNOSIS\_CODE) AS T

FROM

`DISEASE`

WHERE

AGE = '2') AS TOTAL,

(SELECT

DIAGNOSIS\_CODE, COUNT(DIAGNOSIS\_CODE) AS T

FROM

`disease`

WHERE

AGE = '2' AND DIAGNOSIS\_CODE != 'NULL'

GROUP BY DIAGNOSIS\_CODE

ORDER BY T DESC

LIMIT 1) AS T1,

(SELECT

DIAGNOSIS\_CODE, COUNT(DIAGNOSIS\_CODE) AS T

FROM

`disease`

WHERE

AGE = '2' AND DIAGNOSIS\_CODE != 'NULL'

GROUP BY DIAGNOSIS\_CODE

ORDER BY T DESC

LIMIT 1 OFFSET 1) AS T2,

(SELECT

DIAGNOSIS\_CODE, COUNT(DIAGNOSIS\_CODE) AS T

FROM

`disease`

WHERE

AGE = '2' AND DIAGNOSIS\_CODE != 'NULL'

GROUP BY DIAGNOSIS\_CODE

ORDER BY T DESC

LIMIT 1 OFFSET 2) AS T3

UNION SELECT

TOTAL.AGE,

TOTAL.T,

T1.DIAGNOSIS\_CODE,

T1.T,

(T1.T / TOTAL.T) AS FIRST\_COUNT,

T2.DIAGNOSIS\_CODE,

T2.T,

(T2.T / TOTAL.T) AS SECOND\_COUNT,

T3.DIAGNOSIS\_CODE,

T3.T,

(T3.T / TOTAL.T) AS THIRD\_COUNT

FROM

(SELECT

AGE, COUNT(DIAGNOSIS\_CODE) AS T

FROM

`DISEASE`

WHERE

AGE = '3') AS TOTAL,

(SELECT

DIAGNOSIS\_CODE, COUNT(DIAGNOSIS\_CODE) AS T

FROM

`disease`

WHERE

AGE = '3' AND DIAGNOSIS\_CODE != 'NULL'

GROUP BY DIAGNOSIS\_CODE

ORDER BY T DESC

LIMIT 1) AS T1,

(SELECT

DIAGNOSIS\_CODE, COUNT(DIAGNOSIS\_CODE) AS T

FROM

`disease`

WHERE

AGE = '3' AND DIAGNOSIS\_CODE != 'NULL'

GROUP BY DIAGNOSIS\_CODE

ORDER BY T DESC

LIMIT 1 OFFSET 1) AS T2,

(SELECT

DIAGNOSIS\_CODE, COUNT(DIAGNOSIS\_CODE) AS T

FROM

`disease`

WHERE

AGE = '3' AND DIAGNOSIS\_CODE != 'NULL'

GROUP BY DIAGNOSIS\_CODE

ORDER BY T DESC

LIMIT 1 OFFSET 2) AS T3

UNION SELECT

TOTAL.AGE,

TOTAL.T,

T1.DIAGNOSIS\_CODE,

T1.T,

(T1.T / TOTAL.T) AS FIRST\_COUNT,

T2.DIAGNOSIS\_CODE,

T2.T,

(T2.T / TOTAL.T) AS SECOND\_COUNT,

T3.DIAGNOSIS\_CODE,

T3.T,

(T3.T / TOTAL.T) AS THIRD\_COUNT

FROM

(SELECT

AGE, COUNT(DIAGNOSIS\_CODE) AS T

FROM

`DISEASE`

WHERE

AGE = '4') AS TOTAL,

(SELECT

DIAGNOSIS\_CODE, COUNT(DIAGNOSIS\_CODE) AS T

FROM

`disease`

WHERE

AGE = '4' AND DIAGNOSIS\_CODE != 'NULL'

GROUP BY DIAGNOSIS\_CODE

ORDER BY T DESC

LIMIT 1) AS T1,

(SELECT

DIAGNOSIS\_CODE, COUNT(DIAGNOSIS\_CODE) AS T

FROM

`disease`

WHERE

AGE = '4' AND DIAGNOSIS\_CODE != 'NULL'

GROUP BY DIAGNOSIS\_CODE

ORDER BY T DESC

LIMIT 1 OFFSET 1) AS T2,

(SELECT

DIAGNOSIS\_CODE, COUNT(DIAGNOSIS\_CODE) AS T

FROM

`disease`

WHERE

AGE = '4' AND DIAGNOSIS\_CODE != 'NULL'

GROUP BY DIAGNOSIS\_CODE

ORDER BY T DESC

LIMIT 1 OFFSET 2) AS T3

UNION SELECT

TOTAL.AGE,

TOTAL.T,

T1.DIAGNOSIS\_CODE,

T1.T,

(T1.T / TOTAL.T) AS FIRST\_COUNT,

T2.DIAGNOSIS\_CODE,

T2.T,

(T2.T / TOTAL.T) AS SECOND\_COUNT,

T3.DIAGNOSIS\_CODE,

T3.T,

(T3.T / TOTAL.T) AS THIRD\_COUNT

FROM

(SELECT

AGE, COUNT(DIAGNOSIS\_CODE) AS T

FROM

`DISEASE`

WHERE

AGE = '5') AS TOTAL,

(SELECT

DIAGNOSIS\_CODE, COUNT(DIAGNOSIS\_CODE) AS T

FROM

`disease`

WHERE

AGE = '5' AND DIAGNOSIS\_CODE != 'NULL'

GROUP BY DIAGNOSIS\_CODE

ORDER BY T DESC

LIMIT 1) AS T1,

(SELECT

DIAGNOSIS\_CODE, COUNT(DIAGNOSIS\_CODE) AS T

FROM

`disease`

WHERE

AGE = '5' AND DIAGNOSIS\_CODE != 'NULL'

GROUP BY DIAGNOSIS\_CODE

ORDER BY T DESC

LIMIT 1 OFFSET 1) AS T2,

(SELECT

DIAGNOSIS\_CODE, COUNT(DIAGNOSIS\_CODE) AS T

FROM

`disease`

WHERE

AGE = '5' AND DIAGNOSIS\_CODE != 'NULL'

GROUP BY DIAGNOSIS\_CODE

ORDER BY T DESC

LIMIT 1 OFFSET 2) AS T3

UNION SELECT

TOTAL.AGE,

TOTAL.T,

T1.DIAGNOSIS\_CODE,

T1.T,

(T1.T / TOTAL.T) AS FIRST\_COUNT,

T2.DIAGNOSIS\_CODE,

T2.T,

(T2.T / TOTAL.T) AS SECOND\_COUNT,

T3.DIAGNOSIS\_CODE,

T3.T,

(T3.T / TOTAL.T) AS THIRD\_COUNT

FROM

(SELECT

AGE, COUNT(DIAGNOSIS\_CODE) AS T

FROM

`DISEASE`

WHERE

AGE = '6') AS TOTAL,

(SELECT

DIAGNOSIS\_CODE, COUNT(DIAGNOSIS\_CODE) AS T

FROM

`disease`

WHERE

AGE = '6' AND DIAGNOSIS\_CODE != 'NULL'

GROUP BY DIAGNOSIS\_CODE

ORDER BY T DESC

LIMIT 1) AS T1,

(SELECT

DIAGNOSIS\_CODE, COUNT(DIAGNOSIS\_CODE) AS T

FROM

`disease`

WHERE

AGE = '6' AND DIAGNOSIS\_CODE != 'NULL'

GROUP BY DIAGNOSIS\_CODE

ORDER BY T DESC

LIMIT 1 OFFSET 1) AS T2,

(SELECT

DIAGNOSIS\_CODE, COUNT(DIAGNOSIS\_CODE) AS T

FROM

`disease`

WHERE

AGE = '6' AND DIAGNOSIS\_CODE != 'NULL'

GROUP BY DIAGNOSIS\_CODE

ORDER BY T DESC

LIMIT 1 OFFSET 2) AS T3

UNION SELECT

TOTAL.AGE,

TOTAL.T,

T1.DIAGNOSIS\_CODE,

T1.T,

(T1.T / TOTAL.T) AS FIRST\_COUNT,

T2.DIAGNOSIS\_CODE,

T2.T,

(T2.T / TOTAL.T) AS SECOND\_COUNT,

T3.DIAGNOSIS\_CODE,

T3.T,

(T3.T / TOTAL.T) AS THIRD\_COUNT

FROM

(SELECT

AGE, COUNT(DIAGNOSIS\_CODE) AS T

FROM

`DISEASE`

WHERE

AGE = '7') AS TOTAL,

(SELECT

DIAGNOSIS\_CODE, COUNT(DIAGNOSIS\_CODE) AS T

FROM

`disease`

WHERE

AGE = '7' AND DIAGNOSIS\_CODE != 'NULL'

GROUP BY DIAGNOSIS\_CODE

ORDER BY T DESC

LIMIT 1) AS T1,

(SELECT

DIAGNOSIS\_CODE, COUNT(DIAGNOSIS\_CODE) AS T

FROM

`disease`

WHERE

AGE = '7' AND DIAGNOSIS\_CODE != 'NULL'

GROUP BY DIAGNOSIS\_CODE

ORDER BY T DESC

LIMIT 1 OFFSET 1) AS T2,

(SELECT

DIAGNOSIS\_CODE, COUNT(DIAGNOSIS\_CODE) AS T

FROM

`disease`

WHERE

AGE = '7' AND DIAGNOSIS\_CODE != 'NULL'

GROUP BY DIAGNOSIS\_CODE

ORDER BY T DESC

LIMIT 1 OFFSET 2) AS T3

UNION SELECT

TOTAL.AGE,

TOTAL.T,

T1.DIAGNOSIS\_CODE,

T1.T,

(T1.T / TOTAL.T) AS FIRST\_COUNT,

T2.DIAGNOSIS\_CODE,

T2.T,

(T2.T / TOTAL.T) AS SECOND\_COUNT,

T3.DIAGNOSIS\_CODE,

T3.T,

(T3.T / TOTAL.T) AS THIRD\_COUNT

FROM

(SELECT

AGE, COUNT(DIAGNOSIS\_CODE) AS T

FROM

`DISEASE`

WHERE

AGE = '8') AS TOTAL,

(SELECT

DIAGNOSIS\_CODE, COUNT(DIAGNOSIS\_CODE) AS T

FROM

`disease`

WHERE

AGE = '8' AND DIAGNOSIS\_CODE != 'NULL'

GROUP BY DIAGNOSIS\_CODE

ORDER BY T DESC

LIMIT 1) AS T1,

(SELECT

DIAGNOSIS\_CODE, COUNT(DIAGNOSIS\_CODE) AS T

FROM

`disease`

WHERE

AGE = '8' AND DIAGNOSIS\_CODE != 'NULL'

GROUP BY DIAGNOSIS\_CODE

ORDER BY T DESC

LIMIT 1 OFFSET 1) AS T2,

(SELECT

DIAGNOSIS\_CODE, COUNT(DIAGNOSIS\_CODE) AS T

FROM

`disease`

WHERE

AGE = '8' AND DIAGNOSIS\_CODE != 'NULL'

GROUP BY DIAGNOSIS\_CODE

ORDER BY T DESC

LIMIT 1 OFFSET 2) AS T3

UNION SELECT

TOTAL.AGE,

TOTAL.T,

T1.DIAGNOSIS\_CODE,

T1.T,

(T1.T / TOTAL.T) AS FIRST\_COUNT,

T2.DIAGNOSIS\_CODE,

T2.T,

(T2.T / TOTAL.T) AS SECOND\_COUNT,

T3.DIAGNOSIS\_CODE,

T3.T,

(T3.T / TOTAL.T) AS THIRD\_COUNT

FROM

(SELECT

AGE, COUNT(DIAGNOSIS\_CODE) AS T

FROM

`DISEASE`

WHERE

AGE = '9') AS TOTAL,

(SELECT

DIAGNOSIS\_CODE, COUNT(DIAGNOSIS\_CODE) AS T

FROM

`disease`

WHERE

AGE = '9' AND DIAGNOSIS\_CODE != 'NULL'

GROUP BY DIAGNOSIS\_CODE

ORDER BY T DESC

LIMIT 1) AS T1,

(SELECT

DIAGNOSIS\_CODE, COUNT(DIAGNOSIS\_CODE) AS T

FROM

`disease`

WHERE

AGE = '9' AND DIAGNOSIS\_CODE != 'NULL'

GROUP BY DIAGNOSIS\_CODE

ORDER BY T DESC

LIMIT 1 OFFSET 1) AS T2,

(SELECT

DIAGNOSIS\_CODE, COUNT(DIAGNOSIS\_CODE) AS T

FROM

`disease`

WHERE

AGE = '9' AND DIAGNOSIS\_CODE != 'NULL'

GROUP BY DIAGNOSIS\_CODE

ORDER BY T DESC

LIMIT 1 OFFSET 2) AS T3

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Age | Total Diagnosis Count per Age (Total) | Diagnosis Code | Count | Prevalence = Count/Total x 100% |
| 1 | 8 | 2153 | 1 | 12.5 |
| 2767 | 1 | 12.5 |
| 4168 | 1 | 12.5 |
| 2 | 136 | V6284 | 5 | 3.67 |
| 34590 | 4 | 2.94 |
| 486 | 3 | 2.20 |
| 3 | 456 | 486 | 15 | 3.28 |
| 5849 | 13 | 2.85 |
| 5856 | 13 | 2.85 |
| 4 | 440 | 5856 | 25 | 5.68 |
| 486 | 14 | 3.18 |
| 4019 | 13 | 2.95 |
| 5 | 432 | 486 | 14 | 3.24 |
| 389 | 13 | 3.01 |
| 51881 | 12 | 2.77 |
| 6 | 424 | 486 | 22 | 5.18 |
| 5849 | 12 | 2.83 |
| 5990 | 12 | 2.83 |
| 7 | 440 | 486 | 22 | 5 |
| 5849 | 14 | 3.19 |
| 389 | 13 | 2.96 |
| 8 | 376 | 5990 | 20 | 5.31 |
| 5849 | 14 | 3.72 |
| 486 | 13 | 3.46 |
| 9 | 242 | 5849 | 11 | 4.54 |
| 486 | 8 | 3.31 |
| 5990 | 8 | 3.31 |

## In Hospital Mortality

In hospital excel formula =IF(OR(AND(C2,A2="N"),AND(C2,A2=0),AND(D2,B2="N"),AND(D2,B2=0)),"Y","N")

C2 = DC1

A2 = POA\_1

D2 = DC2

B2 = POA\_2

Y = PRESENT

N = NOT PRESENT DURING HOSPITAL, IT WAS AT THE TIME OF ADMISSION

In hospital mortality formula

IN HOSITAL MORTALITY: =IF(AND(E2="Y",G2="B"),"Y","N")

Y = DEAD

N = ALIVE / DEAD BUT NOT BECAUSE OF THE DISEASE

E2 = RESULT OF FIRST FORMULA

G2 = DISCHARGE\_STATUS

1. Total In Hospital Mortality from above formulas = 14

Men = 4/14 = 28.57%

Women = 10/14 = 71.42%

1. In hospital mortality of top 3 diseases

The top 3 diseases are 486, 5849 and 389. As per our analysis there were no in hospital deaths as for diseases are as follows.

|  |  |  |
| --- | --- | --- |
| Disease Code | Count of Death | Sex |
| 389 | 2 | Females |
| 5849 | 1 | Male |
| 486 | 0 | NA |

## Demographics

1. We can see from the below tables that across the patient demographics, they have stayed for a shorter time in the hospital.

Following are the queries.

SELECT

SHORT\_STAY.AGE, SHORT\_STAY.S, LONG\_STAY.L

FROM

(SELECT

AGE, COUNT(STAY\_INDICATOR) AS S

FROM

`HOSPITAL`

WHERE

STAY\_INDICATOR = 'S'

GROUP BY AGE) AS SHORT\_STAY

LEFT JOIN

(SELECT

AGE, COUNT(STAY\_INDICATOR) AS L

FROM

`HOSPITAL`

WHERE

STAY\_INDICATOR = 'L'

GROUP BY AGE) AS LONG\_STAY ON SHORT\_STAY.AGE = LONG\_STAY.AGE

|  |  |  |
| --- | --- | --- |
| AGE | S | L |
| 1 | 4 | NULL |
| 2 | 62 | 6 |
| 3 | 211 | 17 |
| 4 | 211 | 9 |
| 5 | 199 | 17 |
| 6 | 195 | 17 |
| 7 | 206 | 14 |
| 8 | 174 | 14 |
| 9 | 110 | 11 |

SELECT

SHORT\_STAY.SEX, SHORT\_STAY.S, LONG\_STAY.L

FROM

(SELECT

SEX, COUNT(STAY\_INDICATOR) AS S

FROM

`HOSPITAL`

WHERE

STAY\_INDICATOR = 'S'

GROUP BY SEX) AS SHORT\_STAY

LEFT JOIN

(SELECT

SEX, COUNT(STAY\_INDICATOR) AS L

FROM

`HOSPITAL`

WHERE

STAY\_INDICATOR = 'L'

GROUP BY SEX) AS LONG\_STAY ON SHORT\_STAY.SEX = LONG\_STAY.SEX

|  |  |  |
| --- | --- | --- |
| SEX | S | L |
| 1 | 594 | 50 |
| 2 | 778 | 55 |

SELECT

SHORT\_STAY.RACE, SHORT\_STAY.S, LONG\_STAY.L

FROM

(SELECT

RACE, COUNT(STAY\_INDICATOR) AS S

FROM

`HOSPITAL`

WHERE

STAY\_INDICATOR = 'S'

GROUP BY RACE) AS SHORT\_STAY

LEFT JOIN

(SELECT

RACE, COUNT(STAY\_INDICATOR) AS L

FROM

`HOSPITAL`

WHERE

STAY\_INDICATOR = 'L'

GROUP BY RACE) AS LONG\_STAY ON SHORT\_STAY.RACE = LONG\_STAY.RACE

|  |  |  |
| --- | --- | --- |
| RACE | S | L |
| 0 | 5 | NULL |
| 1 | 1103 | 88 |
| 2 | 187 | 14 |
| 3 | 21 | NULL |
| 4 | 13 | NULL |
| 5 | 36 | 3 |
| 6 | 7 | NULL |

Most common Long Stay

SELECT

DIAGNOSIS\_CODE\_1, COUNT(DIAGNOSIS\_CODE\_1) AS TOTAL

FROM

`hospital`

WHERE

STAY\_INDICATOR = 'L'

GROUP BY DIAGNOSIS\_CODE\_1

ORDER BY TOTAL DESC

LIMIT 1;

|  |  |
| --- | --- |
| DIAGNOSIS CODE | COUNT |
| V5789 | 18 |

Most Common Short Stay

SELECT

DIAGNOSIS\_CODE\_1, COUNT(DIAGNOSIS\_CODE\_1) AS TOTAL

FROM

`hospital`

WHERE

STAY\_INDICATOR = 'S'

GROUP BY DIAGNOSIS\_CODE\_1

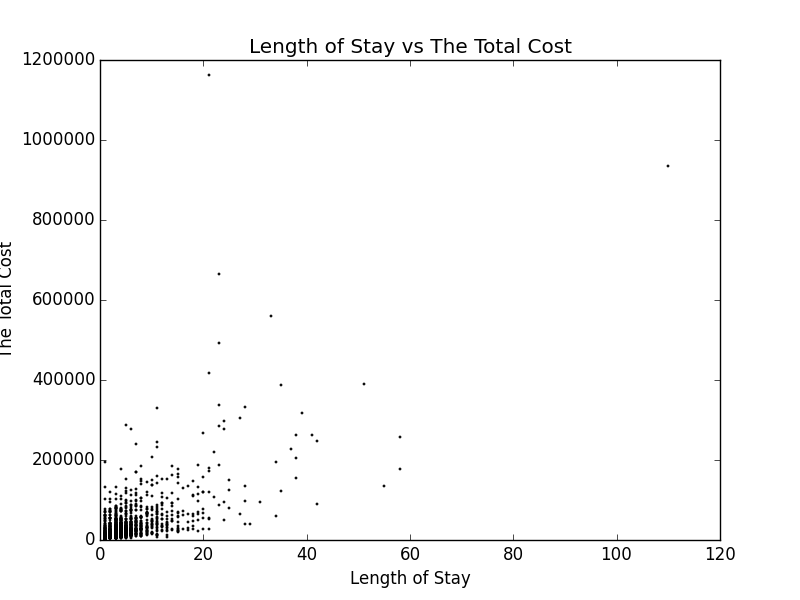
ORDER BY TOTAL DESC

LIMIT 1

|  |  |
| --- | --- |
| DIAGNOSIS CODE | COUNT |
| 389 | 65 |

## Effect of Length of Stay

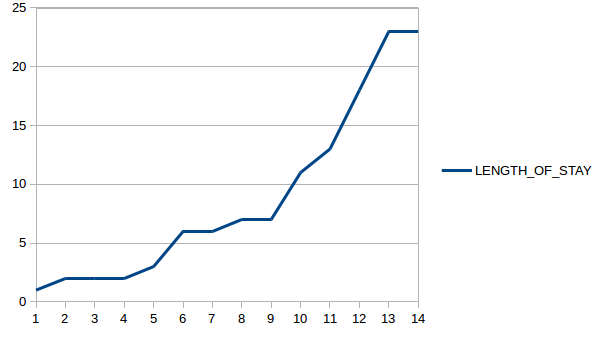
1. As per the following graph of Length of Stay vs Total Cost, we can infer that just a longer stay does not mean a higher cost after discharge. It also depends on the patient’s condition on admission, recovery rate, charges during the treatment etc. (Code task141.py)



1. As per the graph of Length of Stay vs In Hospital Mortality (IHM), the IHM depends on whether the disease was contracted after the patient was admitted and subsequently died because of it.

The maximum length of stay for patients was 110. Of those 14 had died with in hospital mortality. Majority of the in hospital deaths were for patients who stayed for shorter period of time.

|  |  |
| --- | --- |
| LENGTH\_OF\_STAY | IN\_HOSPITAL\_MORTALITY |
| 1 | 1 |
| 2 | 1 |
| 2 | 1 |
| 2 | 1 |
| 3 | 1 |
| 6 | 1 |
| 6 | 1 |
| 7 | 1 |
| 7 | 1 |
| 11 | 1 |
| 13 | 1 |
| 18 | 1 |
| 23 | 1 |
| 23 | 1 |



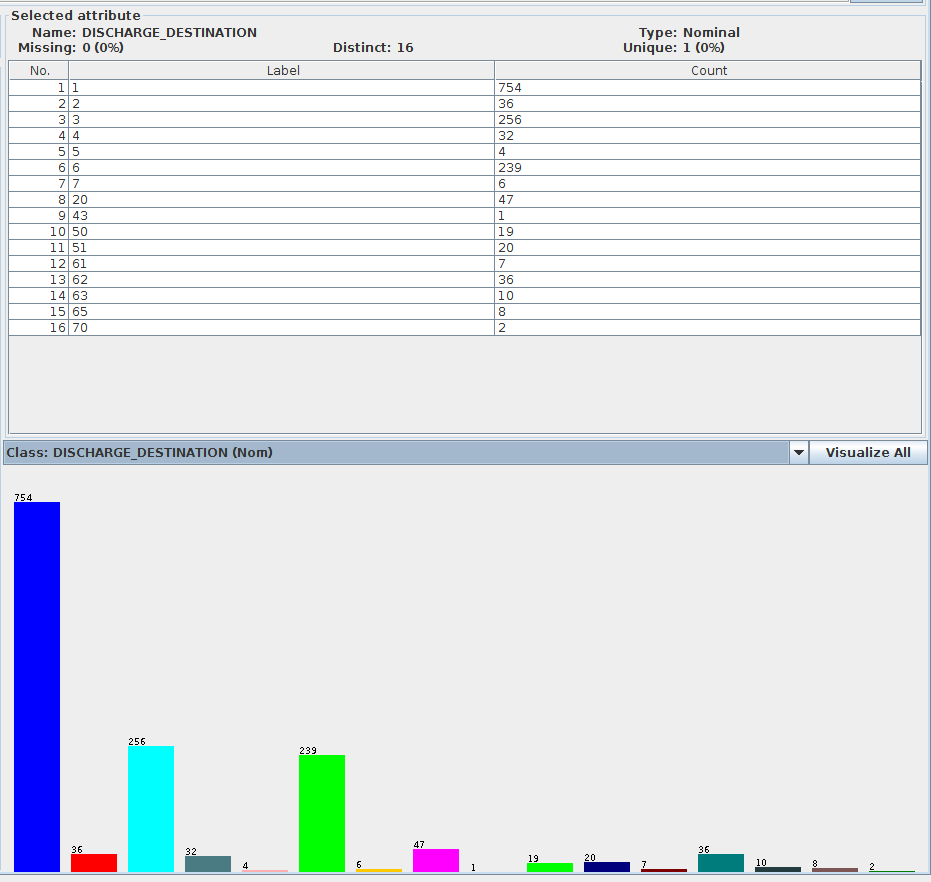
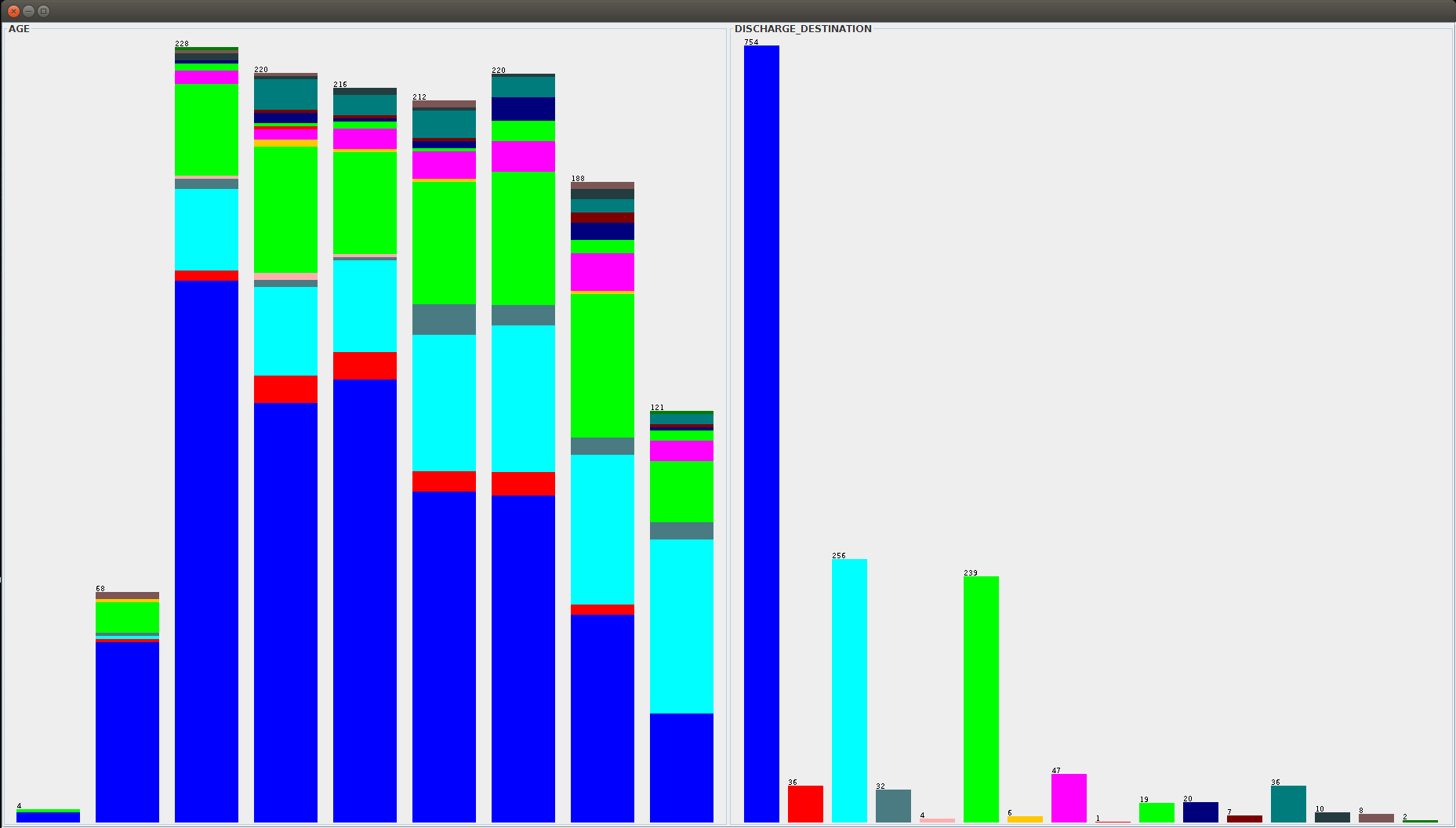
1. Relationship between the Discharge Destination and the Age Group

Following are the steps to get the graphs as shown below.

1. Open weka tool
2. Choose weka explorer
3. Choose ‘open file’ and import the dataset
4. Choose AGE and DISCHARGE\_DESTINATION
5. Click invert
6. Click remove
7. Choose all the attributes
8. From filter choose NumericToNominal
9. Click visualize all to get the following pic 1.

After looking at the graphs, we can infer that, DISCHARGE\_DESTINATION = 1 had the highest number of discharges.

As per the AGE group also, DISCHARGE\_DESTINATION = 1 had the highest among all the DISCHARGE\_DESTINATIONs.



# Task 2.

## Relational Schema

The SQL query to create all the tables for our schema are as below. We have used MySQL as our DB server.

CREATE DATABASE ‘cse6339’;

USE DATABASE ‘cse6339’;

CREATE TABLE IF NOT EXISTS `admission` (

`ADMISSION\_ID` int(5) NOT NULL AUTO\_INCREMENT,

`AGE` int(5) NOT NULL,

`SEX` int(5) NOT NULL,

`RACE` int(5) NOT NULL,

`DAY\_OF\_ADMISSION` int(5) NOT NULL,

`DISCHARGE\_STATUS` varchar(5) NOT NULL,

`STAY\_INDICATOR` varchar(5) NOT NULL,

`DRG\_CODE` int(5) NOT NULL,

`LENGTH\_OF\_STAY` int(5) NOT NULL,

`DRG\_PRICE` int(10) NOT NULL,

`TOTAL\_CHARGES` int(10) NOT NULL,

`COVERED\_CHARGES` int(10) NOT NULL,

`DISCHARGE\_DESTINATION` int(5) NOT NULL,

`SOURCE\_OF\_ADMISSION` int(5) NOT NULL,

`TYPE\_OF\_ADMISSION` int(5) NOT NULL,

`ADMITTING\_DIAGNOSIS\_CODE` varchar(10) NOT NULL,

PRIMARY KEY (`ADMISSION\_ID`)

) ENGINE=InnoDB DEFAULT CHARSET=LATIN1 AUTO\_INCREMENT=1478 ;

CREATE TABLE IF NOT EXISTS `procedure` (

`PROCEDURE\_ID` int(5) NOT NULL AUTO\_INCREMENT,

`PROCEDURE\_CODE` int(10) NOT NULL,

PRIMARY KEY (`PROCEDURE\_ID`)

) ENGINE=InnoDB DEFAULT CHARSET=LATIN1 AUTO\_INCREMENT=385 ;

CREATE TABLE IF NOT EXISTS `diagnosis` (

`DIAGNOSIS\_ID` int(5) NOT NULL AUTO\_INCREMENT,

`DIAGNOSIS\_CODE` varchar(10) DEFAULT NULL,

PRIMARY KEY (`DIAGNOSIS\_ID`)

) ENGINE=InnoDB DEFAULT CHARSET=LATIN1 AUTO\_INCREMENT=808 ;

CREATE TABLE IF NOT EXISTS `admission\_diagnosis` (

`ADMISSION\_ID` int(5) NOT NULL,

`DIAGNOSIS\_ID` int(5) NOT NULL,

`POA\_INDICATOR` varchar(5) NOT NULL,

PRIMARY KEY (`ADMISSION\_ID`,`DIAGNOSIS\_ID`),

KEY `DIAGNOSIS\_ID` (`DIAGNOSIS\_ID`)

) ENGINE=InnoDB DEFAULT CHARSET=LATIN1;

CREATE TABLE IF NOT EXISTS `admission\_procedure` (

`ADMISSION\_ID` INT(5) NOT NULL,

`PROCEDURE\_ID` INT(5) NOT NULL,

PRIMARY KEY (`ADMISSION\_ID` , `PROCEDURE\_ID`),

KEY `PROCEDURE\_ID` (`PROCEDURE\_ID`)

) ENGINE=INNODB DEFAULT CHARSET=LATIN1;

ALTER TABLE `admission\_diagnosis`

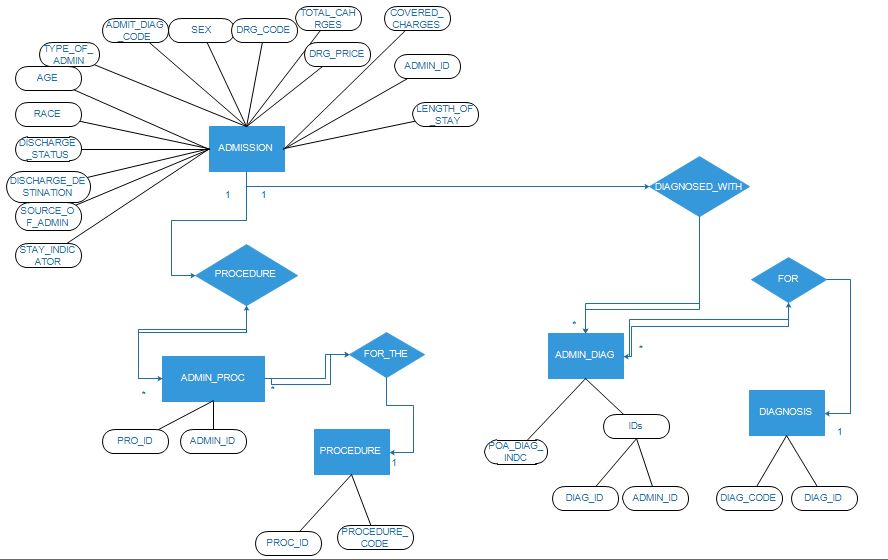
ADD CONSTRAINT `admission\_diagnosis\_ibfk\_2` FOREIGN KEY (`DIAGNOSIS\_ID`) REFERENCES `diagnosis` (`DIAGNOSIS\_ID`),

ADD CONSTRAINT `admission\_diagnosis\_ibfk\_1` FOREIGN KEY (`ADMISSION\_ID`) REFERENCES `admission` (`ADMISSION\_ID`);

ALTER TABLE `admission\_procedure`

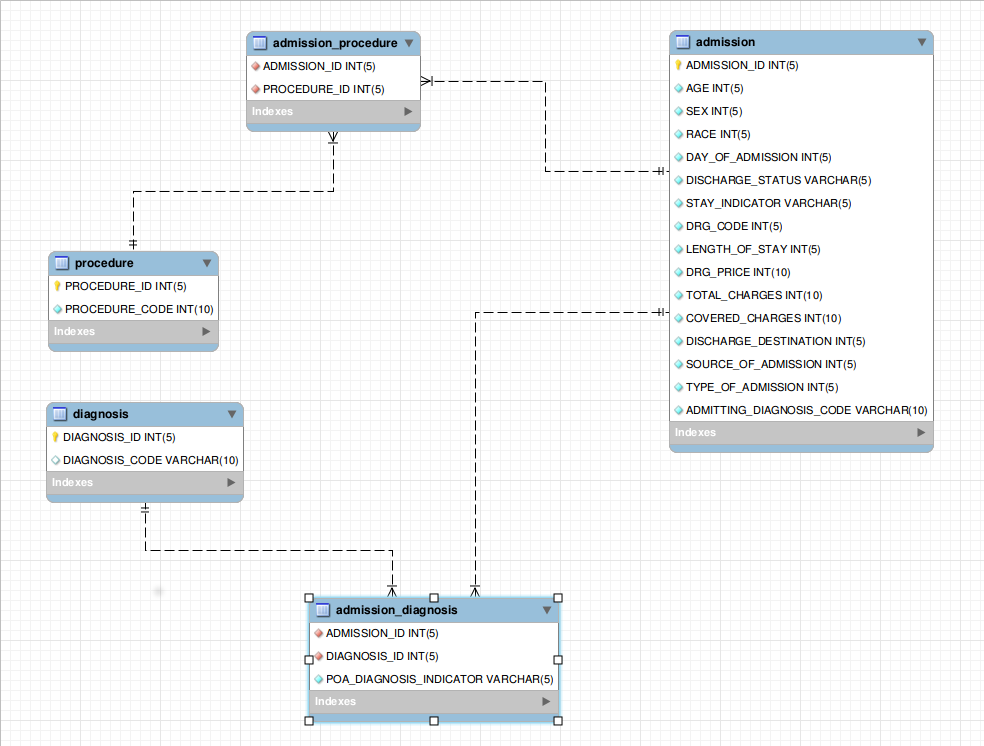
ADD CONSTRAINT `admission\_procedure\_ibfk\_2` FOREIGN KEY (`PROCEDURE\_ID`) REFERENCES `procedure` (`PROCEDURE\_ID`),

ADD CONSTRAINT `admission\_procedure\_ibfk\_1` FOREIGN KEY (`ADMISSION\_ID`) REFERENCES `admission` (`ADMISSION\_ID`);



Double arrow are total participation.

Single arrow is partial participation.



## Import Data

We have used LOAD INFILE command to push data into the tables. We created 5 CSVs, 1 for each table.

LOAD DATA INFILE 'C:/Users/Dhruv/Desktop/ADMISSION.CSV' INTO TABLE `ADMISSION` FIELDS TERMINATED BY ',' LINES TERMINATED BY '\r\n' IGNORE 1 LINES

LOAD DATA INFILE 'C:/Users/Dhruv/Desktop/DIAGNOSIS.CSV' INTO TABLE `DIAGNOSIS` FIELDS TERMINATED BY ',' LINES TERMINATED BY '\r\n' IGNORE 1 LINES

LOAD DATA INFILE 'C:/Users/Dhruv/Desktop/PROCEDURE.CSV' INTO TABLE `PROCEDURE` FIELDS TERMINATED BY ',' LINES TERMINATED BY '\r\n' IGNORE 1 LINES

LOAD DATA INFILE 'C:/Users/Dhruv/Desktop/ADMISSION\_DIAGNOSIS.CSV' INTO TABLE `ADMISSION\_DIAGNOSIS` FIELDS TERMINATED BY ',' LINES TERMINATED BY '\r\n' IGNORE 1 LINES

LOAD DATA INFILE 'C:/Users/Dhruv/Desktop/ADMISSION\_PROCEDURE.CSV' INTO TABLE `ADMISSION\_PROCEDURE` FIELDS TERMINATED BY ',' LINES TERMINATED BY '\r\n' IGNORE 1 LINES

Following is Query used to combine DIAGNOSIS\_CODE\_1 and DIAGNOSIS\_CODE\_2 into a single column.

SSELECT

ADMISSION\_ID, PROCEDURE\_ID

FROM

ADMISSION AS H,

PROCEDURE AS P

WHERE

P.PROCEDURE\_CODE = H.PROCEDURE\_CODE\_1

OR P.PROCEDURE\_CODE = H.PROCEDURE\_CODE\_2

AND PROCEDURE\_CODE != 0;

SELECT

H.ADMISSION\_ID, P.diagnosis\_ID

FROM

ADMISSION AS H,

DIAGNOSIS AS P

WHERE

P.DIAGNOSIS\_CODE = H.DIAGNOSIS\_CODE\_1

OR P.DIAGNOSIS\_CODE = H.DIAGNOSIS\_CODE\_2

AND P.DIAGNOSIS\_CODE != “NULL”;

We have used the following Excel formula to combine POA\_DIAGNOSIS\_INDICATOR\_1 and POA\_DIAGNOSIS\_INDICATOR\_2 into a single column POA\_DIAGNOSIS\_INDICATOR =INDEX($A$2:$B$1478,INT((ROWS(F$2:F3)-1)/2)+1,MOD(ROWS(F$2:F3)-1,2)+1) WHERE A2:B1478 = ARRAY RANGE

## SQL queries

### Query to get back the original csv file from the DB

SELECT

A.ADMISSION\_ID,

A.AGE,

A.SEX,

A.RACE,

A.DAY\_OF\_ADMISSION,

A.DISCHARGE\_STATUS,

A.STAY\_INDICATOR,

A.DRG\_CODE,

A.LENGTH\_OF\_STAY,

A.DRG\_PRICE,

A.TOTAL\_CHARGES,

A.COVERED\_CHARGES,

AD.POA\_INDICATOR,

D.DIAGNOSIS\_CODE,

P.PROCEDURE\_CODE,

A.DISCHARGE\_DESTINATION,

A.SOURCE\_OF\_ADMISSION,

A.TYPE\_OF\_ADMISSION,

A.ADMITTING\_DIAGNOSIS\_CODE

FROM

`PROCEDURE` AS P,

`ADMISSION` AS A,

`ADMISSION\_DIAGNOSIS` AS AD,

`ADMISSION\_PROCEDURE` AS AP,

`DIAGNOSIS` AS D

WHERE

A.ADMISSION\_ID = AD.ADMISSION\_ID

AND A.ADMISSION\_ID = AP.ADMISSION\_ID

AND D.DIAGNOSIS\_ID = AD.DIAGNOSIS\_ID

AND P.PROCEDURE\_ID = AP.PROCEDURE\_ID;

### Coverage Ratio

Coverage ratio for LENGTH\_OF\_STAY > 5 is 97.10%

SELECT

LENGTH\_OF\_STAY,

SUM(COVERED\_CHARGES) / SUM(COVERED\_CHARGES) AS COVERAGE\_RATIO

FROM

`hospital`

WHERE

LENGTH\_OF\_STAY > 5;

Coverage for STAY\_INDICATOR=’L’ is 97.06%.

SELECT

STAY\_INDICATOR,

SUM(COVERED\_CHARGES) / SUM(COVERED\_CHARGES) AS COVERAGE\_RATIO

FROM

`ADMISSION`

WHERE

STAY\_INDICATOR = 'L'

There were some cases where the COVERED\_CHARGES = 0. In comparison, all 97% of the Long Stay patients had coverage ratio of 97%.

### Variation and Length of Stay

We have used the following SQL query for this task.

SELECT

DAY\_OF\_ADMISSION, AVG(LENGTH\_OF\_STAY) AS AVERAGE

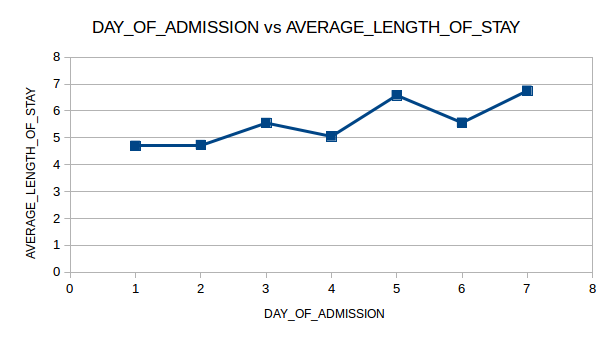
FROM

ADMISSION

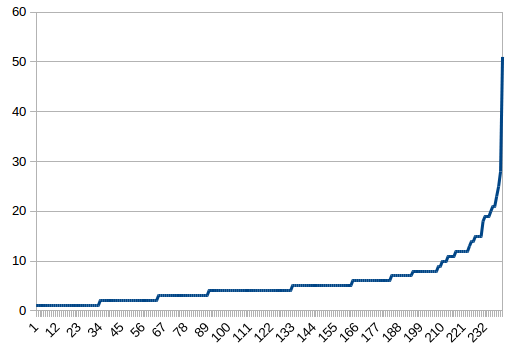
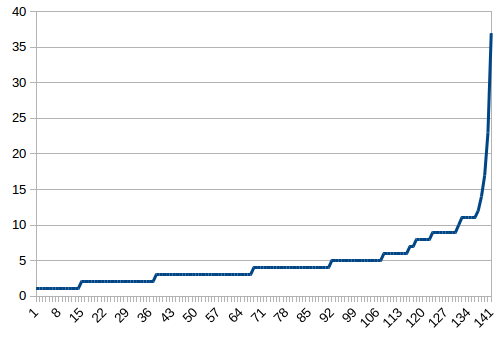
GROUP BY DAY\_OF\_ADMISSION

The patients admitted on day 5 (Thursday) and day 7 (Saturday) have stayed longer in the hospital.

|  |  |  |
| --- | --- | --- |
| DAY | DAY\_OF\_ADMISSION | AVERAGE\_LENGTH\_OF\_STAY |
| SUNDAY | 1 | 4.70 |
| MONDAY | 2 | 4.73 |
| TUESDAY | 3 | 5.55 |
| WEDNESDAY | 4 | 5.05 |
| THURSDAY | 5 | 6.58 |
| FRIDAY | 6 | 5.56 |
| SATURDAY | 7 | 6.75 |

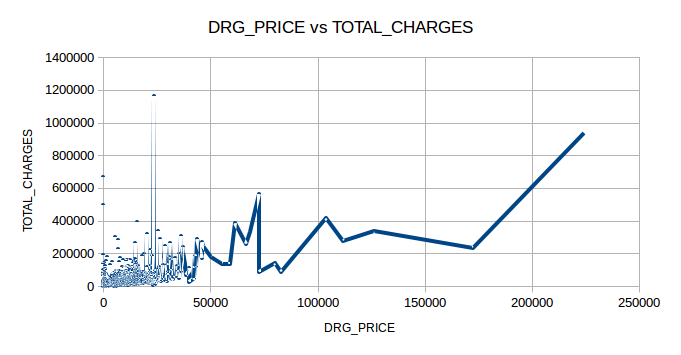


Average patients admitted on Friday have stayed longer than those admitted on Monday. Based on our web search and personal experience, we can say that Friday is more prone towards partying and traveling. This prevalence can give rise to more admissions than on Mondays. As per this web article ['Study: Higher risk of death for patients admitted to hospitals on weekends'](http://www.greenvilleonline.com/story/health/2015/07/09/study-higher-risk-death-patients-admitted-hospitals-weekends/29904991/), there is “Friday effect”, where the patients undergo a planned surgery on Fridays, even though the risk is 33% higher than if they were admitted on Monday.



### DRG\_PRICE vs TOTAL\_CHARGES

There is no linear relationship between DRG\_PRICE and TOTAL\_CHARGES. There were cases where the DRG\_PRICE=0 and the TOTAL\_CHARGES are still posted for the patient. (Code task23d.py)

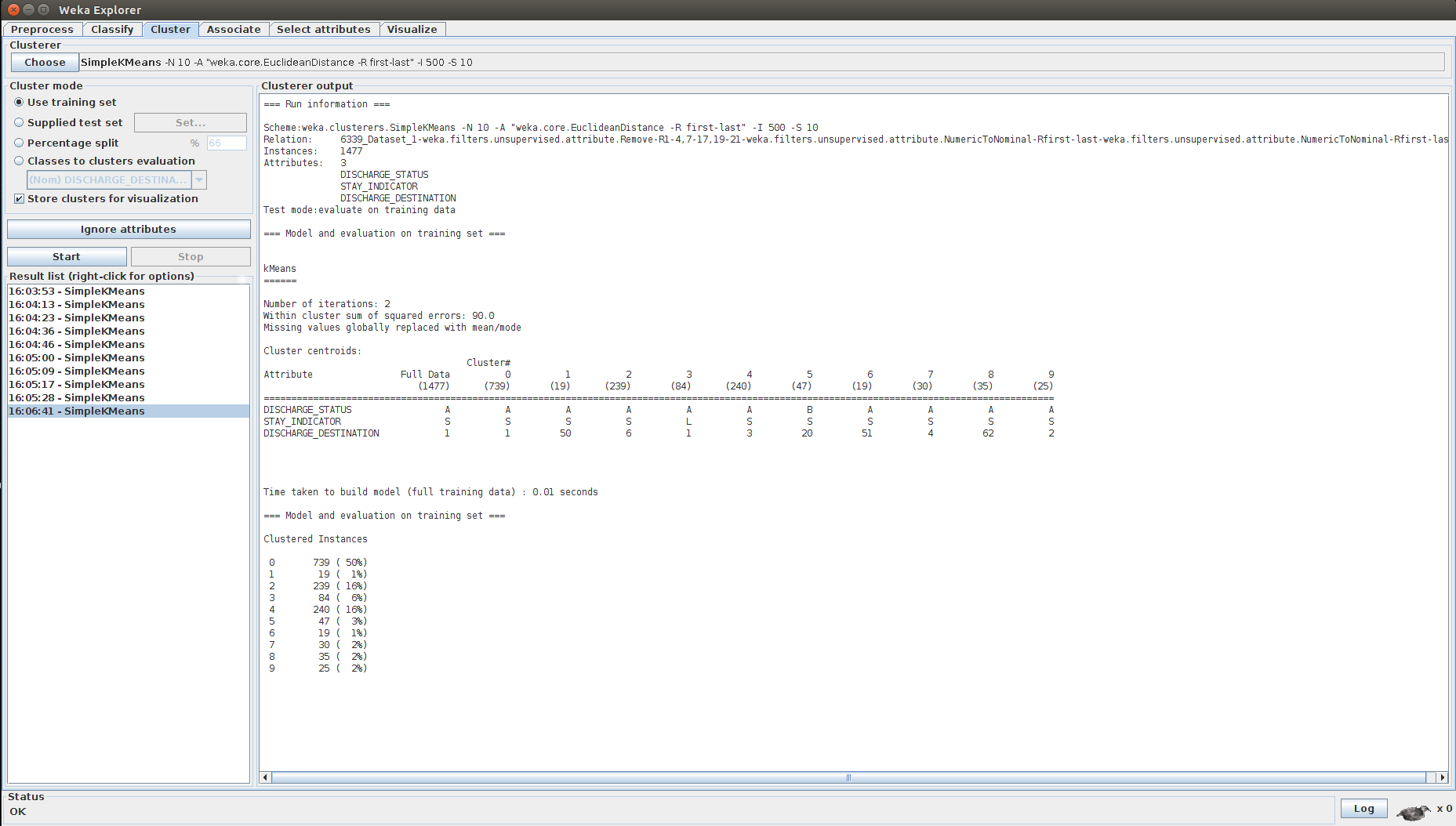


# Task 3.

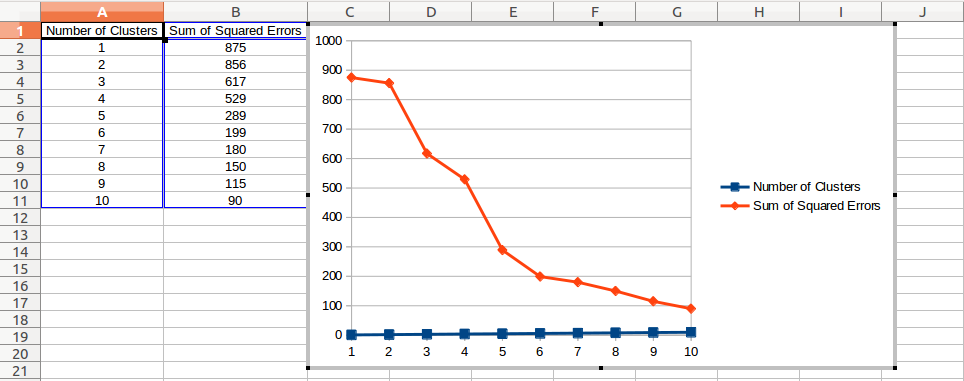
## Appropriate Number of Clusters

The appropriate number of clusters which are required to adequately describe’ the discharge characteristics of the patients (discharge destination, discharge status, stay indicator). Use the elbow method to define the number, by evaluating the‘within cluster sum of squared errors’ you get as a result in your Weka output. Draw an appropriate graph to explain your answer.

* 1. Select DISCHARGE\_DESTINATION, DISCHARGE\_STATUS and STAY\_INDICATOR
  2. Deselect all other attributes
  3. Change types to nominal
  4. Go to clusters tab and select SimpleKMeans
  5. Note the Sum of Squared Error values
  6. Plot the graph of Number of Clusters vs Sum of Square Error values



The elbow is point is 6 clusters. Adding more clusters does not provide a better modeling of data given to us. After 6 clusters the variation is not much and is negligible. Hence, we have chosen the elbow as 6 clusters.



## Calculate Number of Clusters

|  |  |
| --- | --- |
| Number of Clusters | Sum of Squared Errors |
| 1 | 875 |
| 2 | 856 |
| 3 | 617 |
| 4 | 529 |
| 5 | 289 |
| 6 | 199 |
| 7 | 180 |
| 8 | 150 |
| 9 | 115 |
| 1- | 90 |

## Interesting Profiles

Here we are consider for number of clusters as 6.

There are two majority clusters Cluster0 and Cluster4.

In Cluster0 739 out of 1477 were discharged Alive at Discharge Destination 1. We can assume that, Discharge Destination 1 is the most commonly used for discharging patients.

In Cluster4 240 out of 1477 were discharged Alive at Discharge Destination 3. We can assume that, Discharge Destination 3 is the next most commonly used for discharging patients.

## Supervised vs Unsupervised

We have used unsupervised data mining technique. This is because of the following

* 1. We change the type of the data
  2. KMeans clustering works on non-labeled data.
  3. Clustering is always a part of unsupervised learning.

# Task 4.

## Method to find DRG\_PRICE\_BINARY

Changing DRG\_PRICE to binary values

* 1. Add a column next to DRG\_PRICE
  2. Add condition IF(DRG\_PRICE>80000,1,0)
  3. If DRG\_PRICE is greater than $80000, value is set as 1
  4. If DRG\_PRICE is lesser than $80000, value is set as 0
  5. Copy paste the formula to all the cells in the column

## Clinicians/Administrators know the following details

1. When the patient enters the hospital
2. AGE
3. SEX
4. RACE
5. DAY\_OF\_ADMISSION
6. SOURCE\_OF\_ADMISSION
7. TYPE\_OF\_ADMISSION
8. ADMITTING\_DIAGNOSIS\_CODE

When the patient is discharged from the hospital

1. AGE
2. SEX
3. RACE
4. DAY\_OF\_ADMISSION
5. DISCHARGE\_STATUS
6. STAY\_INDICATOR
7. DRG\_CODE
8. LENGTH\_OF\_STAY
9. DRG\_PRICE
10. TOTAL\_CHARGES
11. POA\_DIAGNOSIS\_INDICATOR\_1
12. POA\_DIAGNOSIS\_INDICATOR\_2
13. DIAGNOSIS\_CODE\_1
14. DIAGNOSIS\_CODE\_1
15. PROCEDURE\_CODE\_1
16. PROCEDURE\_CODE\_2
17. DISCHARGE\_DESTINATION
18. SOURCE\_OF\_ADMISSION
19. TYPE\_OF\_ADMISSION
20. ADMITTING\_DIAGNOSIS\_CODE

## Scenarios

### Exclude 7, 10 and 11.

DRG\_CODE, TOTAL\_CHARGES and COVERED\_CHARGES will be unknown at the time of admission and during hospital stay.

### CfsSubsetEval

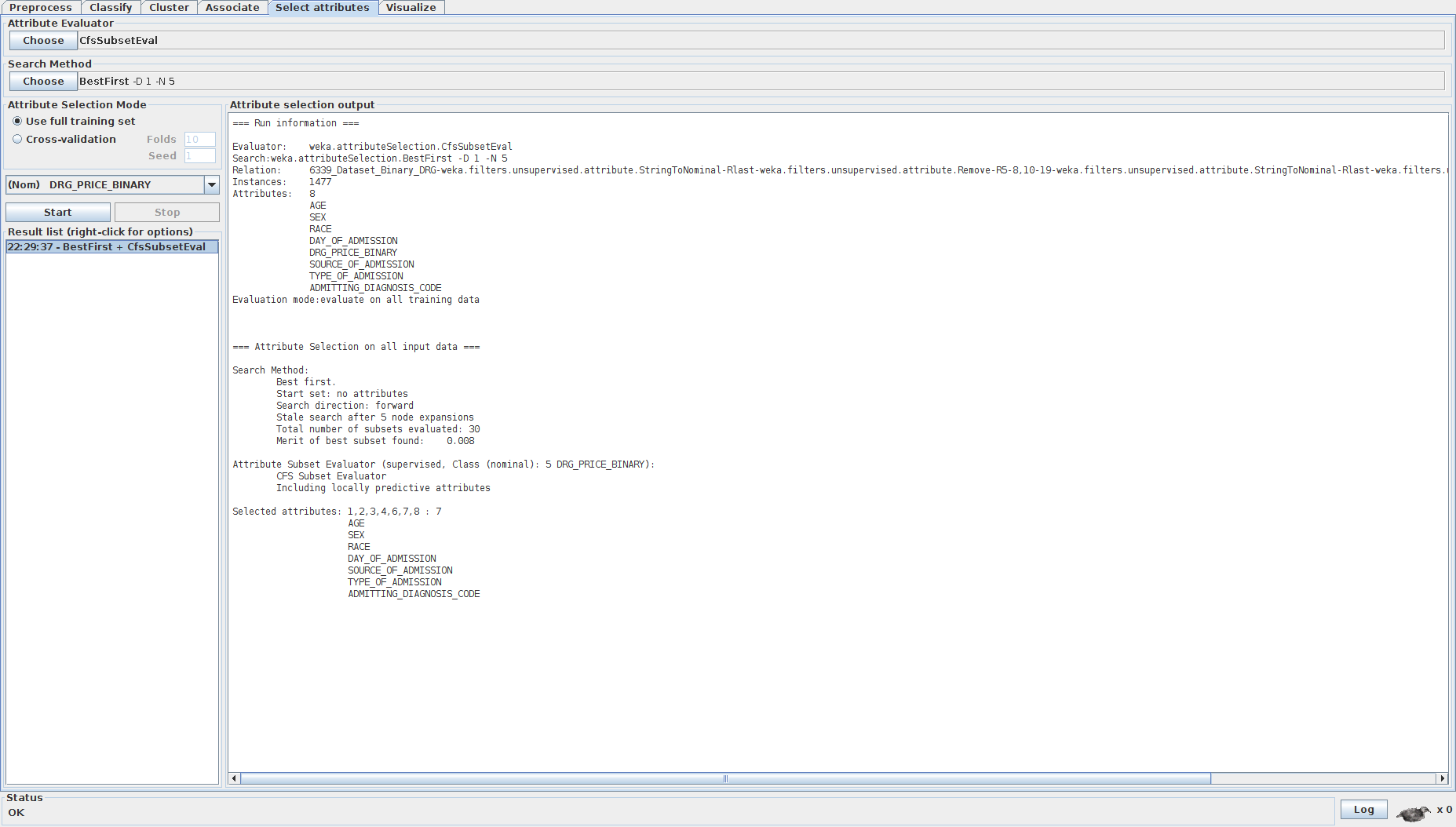
CfsSubsetEval and BestFirst are used to select the features. Following is the process

1. Select the required attributes
2. Remove the rest of the attributes
3. Convert string values to nominal
4. Go to select attributes and select CfsSubsetEval and BestFirst

Scenario 1 - AGE, SEX, RACE, DAY\_OF\_ADMISSION, SOURCE\_OF\_ADMISSION, TYPE\_OF\_ADMISSION and ADMITTING\_DIAGNOSIS\_CODE.

Features selected - AGE, SEX, RACE, DAY\_OF\_ADMISSION, SOURCE\_OF\_ADMISSION, TYPE\_OF\_ADMISSION and ADMITTING\_DIAGNOSIS\_CODE.

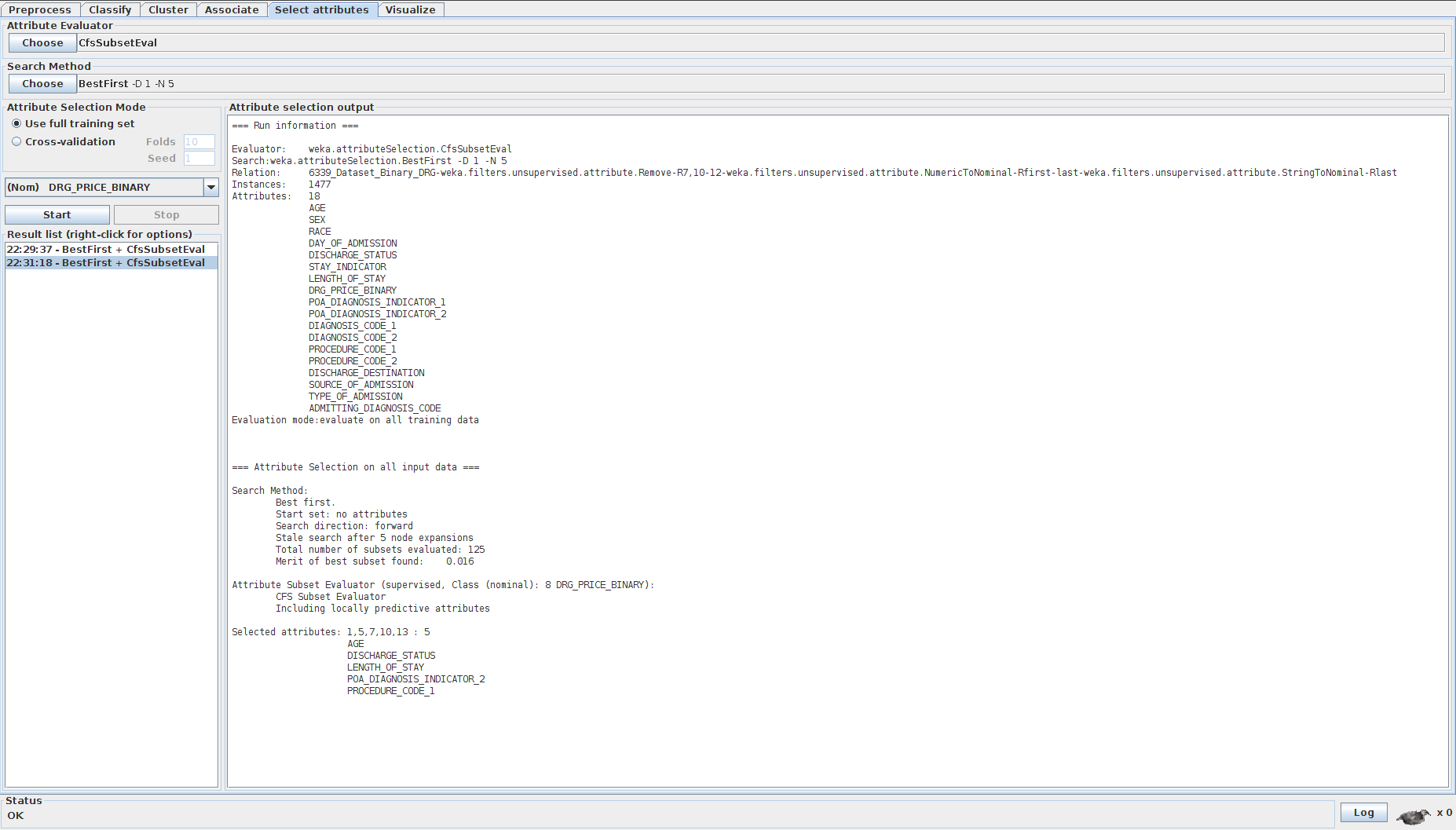
These are listed, best to worst, according to their individual predictive ability of DRG\_PRICE\_BINARY field. These are fields have high correlation with DRG\_PRICE\_BINARY and will help in getting better classification.



Scenario 2 - Select these attirbutes AGE, SEX, RACE, DAY\_OF\_ADMISSION, DISCHARGE\_STATUS, STAY\_INDICATOR, LENGTH\_OF\_STAY, POA\_DIAGNOSIS\_INDICATOR\_1, POA\_DIAGNOSIS\_INDICATOR\_2, DIAGNOSIS\_CODE\_1, DIAGNOSIS\_CODE\_2, PROCEDURE\_CODE\_1, PROCEDURE\_CODE\_2, DISCHARGE\_DESTINATION, SOURCE\_OF\_ADMISSION, TYPE\_OF\_ADMISSION and ADMITTING\_DIAGNOSIS\_CODE

Features selected - AGE, DISCHARGE\_STATUS, LENGTH\_OF\_STAY, POA\_DIAGNOSIS\_INDICATOR\_2, PROCEDURE\_CODE\_1

These are listed, best to worst, according to their individual predictive ability of DRG\_PRICE\_BINARY field. These are fields have high correlation with DRG\_PRICE\_BINARY and will help in getting better classification.



### Classifiers

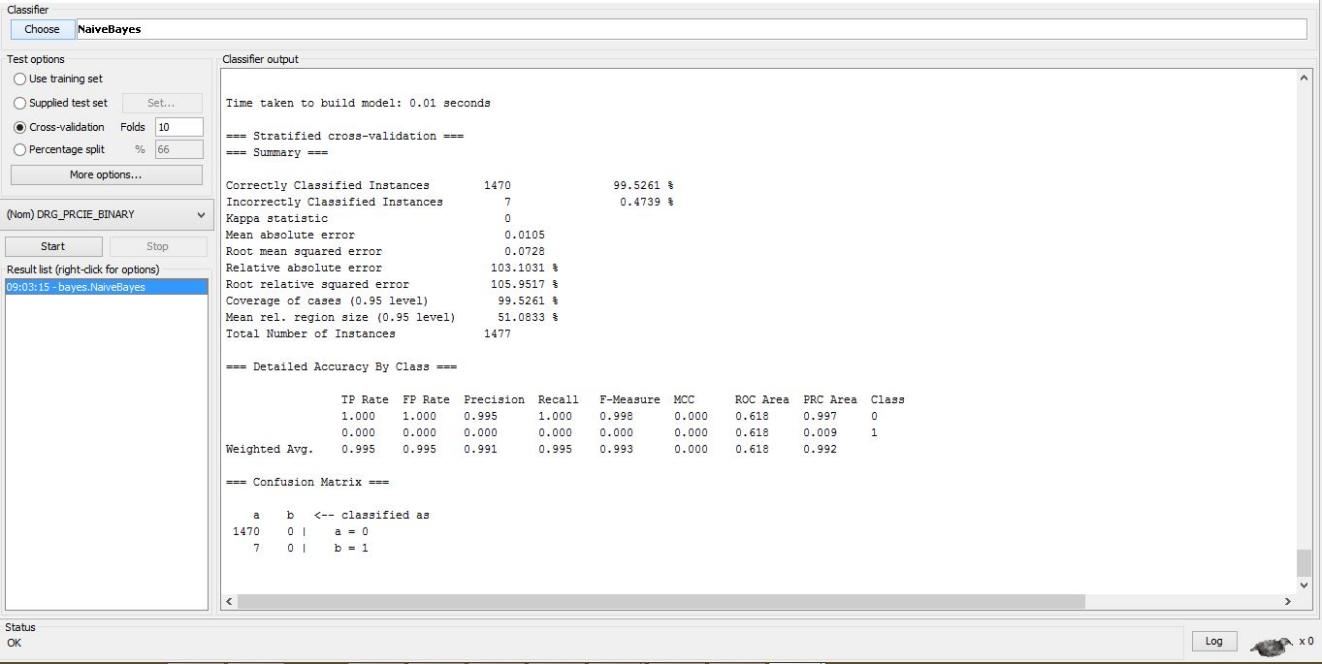
1. Naive Bayes

Scenario 1 - AGE, SEX, RACE, DAY\_OF\_ADMISSION, SOURCE\_OF\_ADMISSION, TYPE\_OF\_ADMISSION and ADMITTING\_DIAGNOSIS\_CODE.

Of the 1477 values 1470 were correctly classified and 7 were incorrectly classified.

99.52% were correctly classified.

0.47% were incorrectly classified.

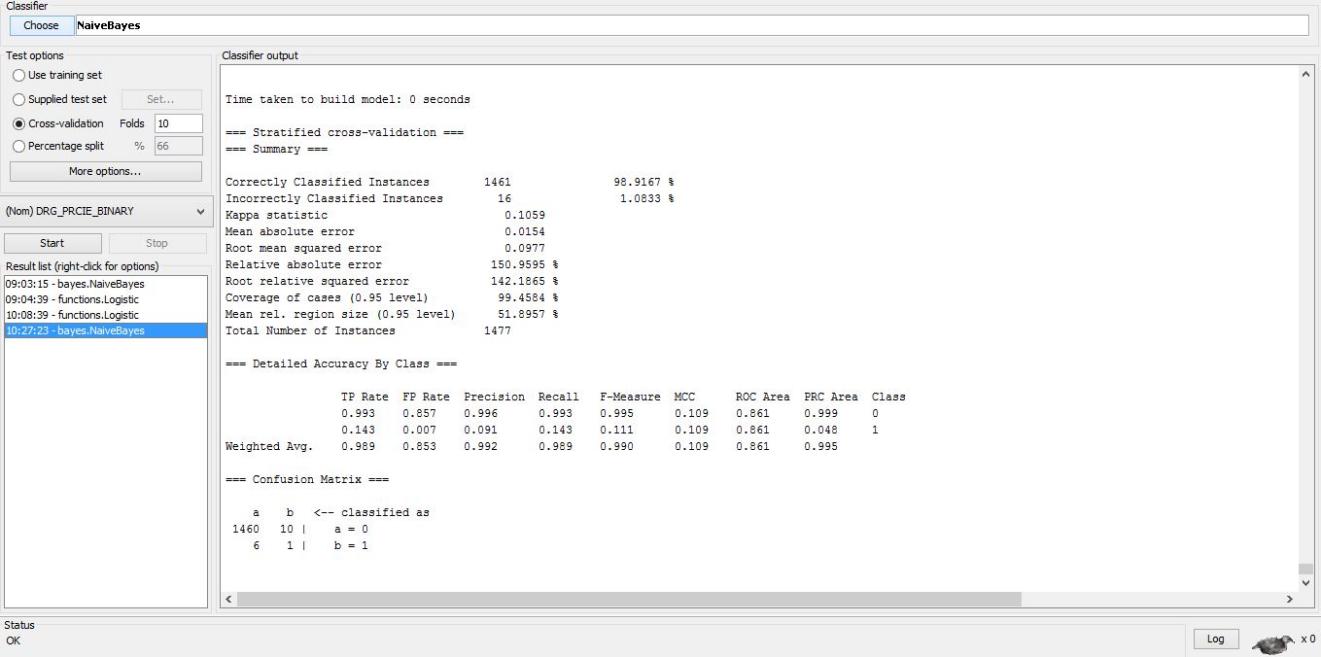


Scenario 2 - AGE, SEX, RACE, DAY\_OF\_ADMISSION, DISCHARGE\_STATUS, STAY\_INDICATOR, LENGTH\_OF\_STAY, POA\_DIAGNOSIS\_INDICATOR\_1, POA\_DIAGNOSIS\_INDICATOR\_2, DIAGNOSIS\_CODE\_1, DIAGNOSIS\_CODE\_2, PROCEDURE\_CODE\_1, PROCEDURE\_CODE\_2, DISCHARGE\_DESTINATION, SOURCE\_OF\_ADMISSION, TYPE\_OF\_ADMISSION, ADMITTING\_DIAGNOSIS\_CODE

Of the 1477 values 1461 were correctly classified and 16 were incorrectly classified.

98.91% were correctly classified.

1.08% were incorrectly classified.



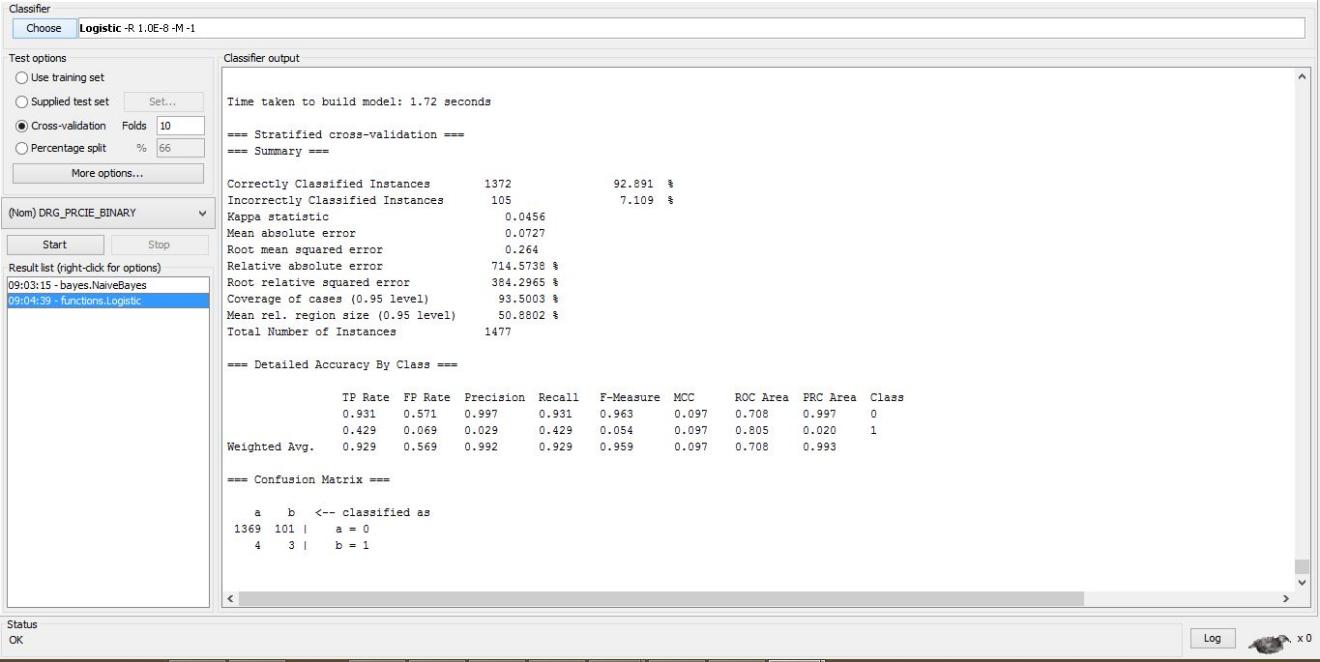
1. Logistic Regression

Scenario 1 - AGE, SEX, RACE, DAY\_OF\_ADMISSION, SOURCE\_OF\_ADMISSION, TYPE\_OF\_ADMISSION and ADMITTING\_DIAGNOSIS\_CODE.

Of the 1477 values 1372 were correctly classified and 105 were incorrectly classified.

92.89% were correctly classified.

7.10% were incorrectly classified.

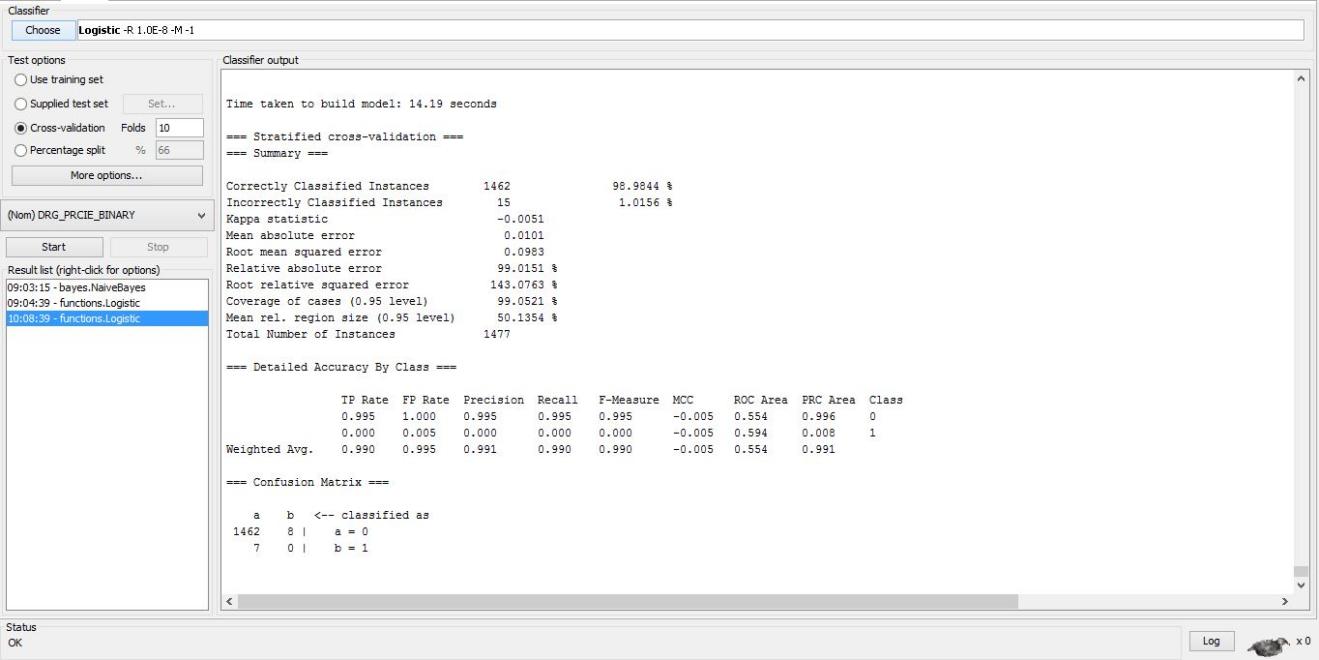


Scenario 2 - AGE, SEX, RACE, DAY\_OF\_ADMISSION, DISCHARGE\_STATUS, STAY\_INDICATOR, LENGTH\_OF\_STAY, POA\_DIAGNOSIS\_INDICATOR\_1, POA\_DIAGNOSIS\_INDICATOR\_2, DIAGNOSIS\_CODE\_1, DIAGNOSIS\_CODE\_2, PROCEDURE\_CODE\_1, PROCEDURE\_CODE\_2, DISCHARGE\_DESTINATION, SOURCE\_OF\_ADMISSION, TYPE\_OF\_ADMISSION, ADMITTING\_DIAGNOSIS\_CODE

Of the 1477 values 1462 values were correctly classified and 15 were incorrectly classified.

98.98% were correctly classified.

1.01% were incorrectly classified.



### Accuracy

### Overall Accuracy

Formula for Accuracy = (TP +TN)/(TP + FN + FP + TN)

1. Naive Bayes
2. Scenario 1 - 99.5261%
3. Scenario 2 - 98.849%
4. Logistic
5. Scenario 1 - 92.2139%
6. Scenario 2 - 99.1198%

### Greater than $80000

1. Naive Bayes

Scenario 1

Accuracy = TN/(TN + FP) = 0/(7+0) = 0

0% correctly were classified as more than $80000.

Scenario 2

Accuracy = TN/(TN + FP) = 1/(6 + 1) = .1428

14.28% were correctly classified as more than $80000

1. Logistic

Scenario 1

Accuracy = TN/(TN + FP) = 3/(4+3) = .42.85

42.85% correctly were classified as more than $80000.

Scenario 2

Accuracy = TN/(TN + FP) = 7/(7 + 0) = 0

0% were correctly classified as more than $80000

### Less than $80000

1. Naive Bayes

Scenario 1

Accuracy = TP/(TP + FN) = 1470/(1470+0) = 1

100% correctly were classified as less than $80000.

Scenario 2

Accuracy = TP/(TP + FN) = 1460/(1460+10) = 0.9931

99.31% were correctly classified as more than $80000

1. Logistic

Scenario 1

Accuracy = TP/(TP + FN) = 1369/(1369+101) = .9312

93.12% correctly were classified as more than $80000.

Scenario 2

Accuracy = TP/(TP + FN) = 1462/(1462+8) = .9945

99.45% were correctly classified as more than $80000