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.

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

2. *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

i. Total In Hospital Mortality from above formulas = 14

   Men = 4/14 = 28.57%

   Women = 10/14 = 71.42%

ii. 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

3. Demographics

i. 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

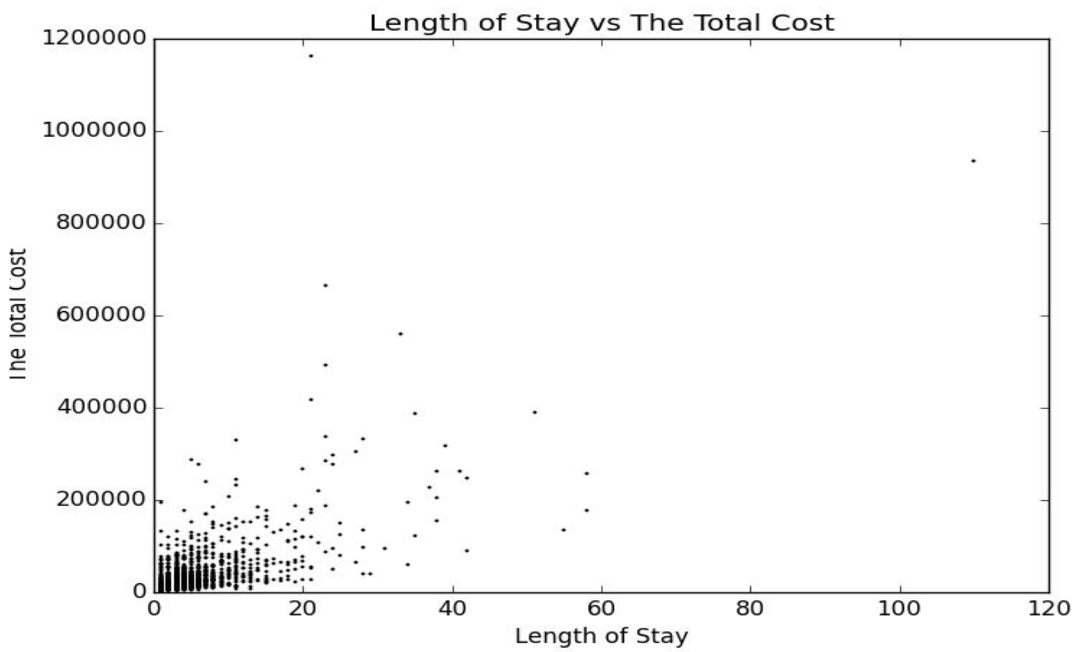
```
SELECT
    DIAGNOSIS_CODE_1, COUNT(DIAGNOSIS_CODE_1) AS TOTAL
FROM
    `hospital`
WHERE
    STAY_INDICATOR = 'S'
GROUP BY DIAGNOSIS_CODE_1
ORDER BY TOTAL DESC
LIMIT 1
```

DIAGNOSIS CODE	COUNT

389	65

4. Effect of Length of Stay

i. 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)

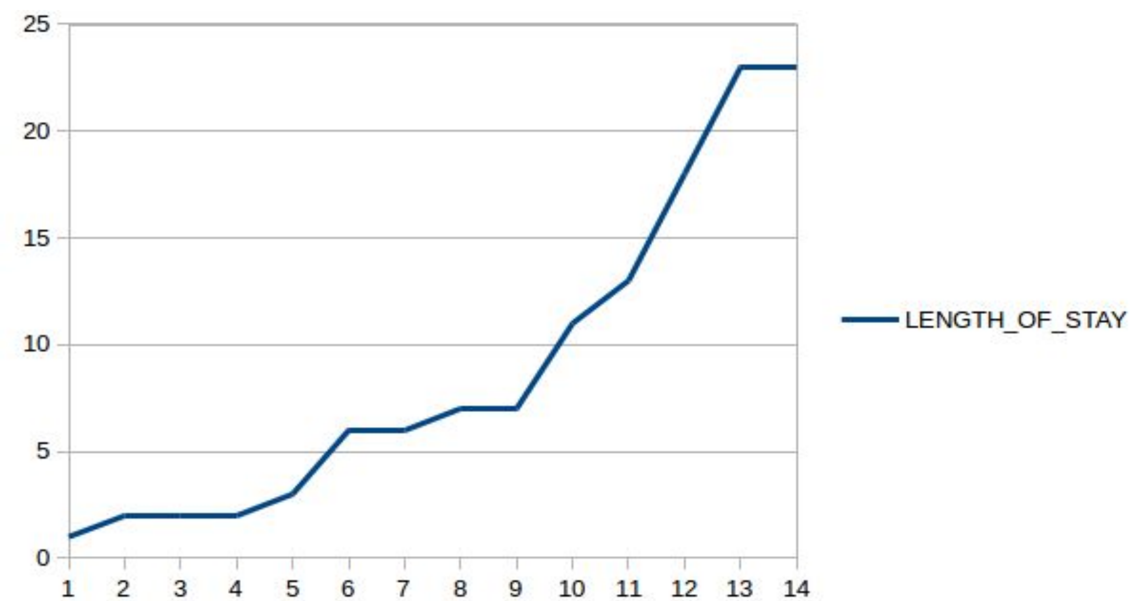


ii. As per the graph of Length of Stay vs In Hospital Mortality (IHM), the IHM depends on whether the disease was contracted after the patient was admitted and subsequently died because of it.

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.

i. Open weka tool

ii. Choose weka explorer

iii. Choose ‘open file’ and import the dataset

iv. Choose AGE and DISCHARGE_DESTINATION

v. Click invert

vi. Click remove

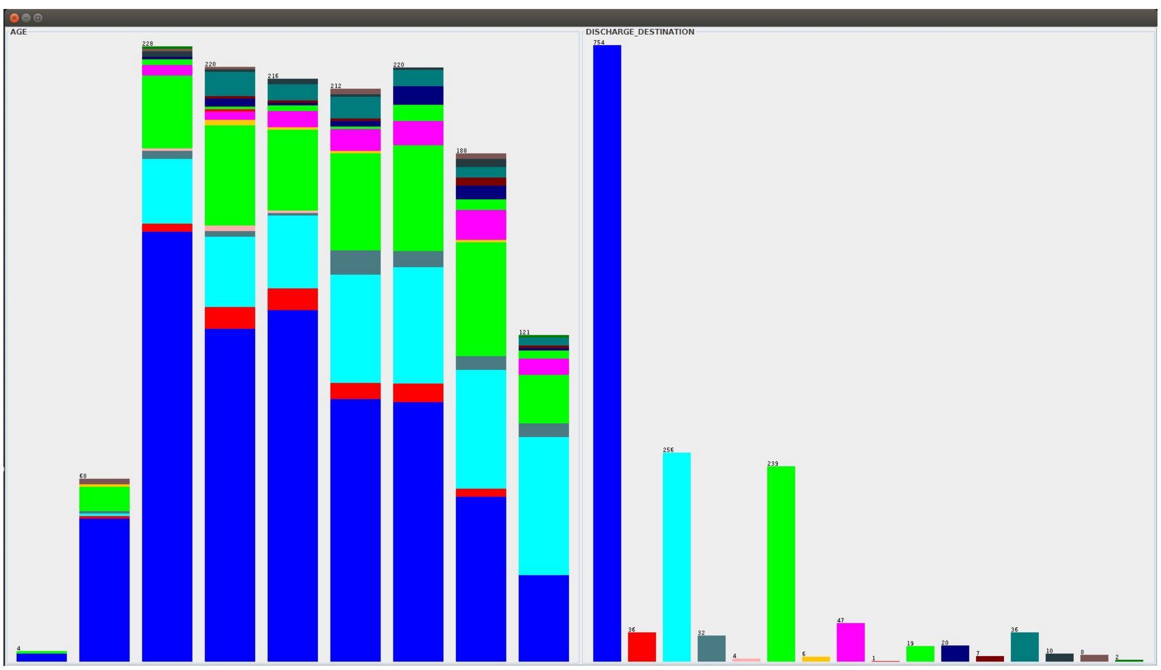
vii. Choose all the attributes

viii. From filter choose NumericToNominal

ix. 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.



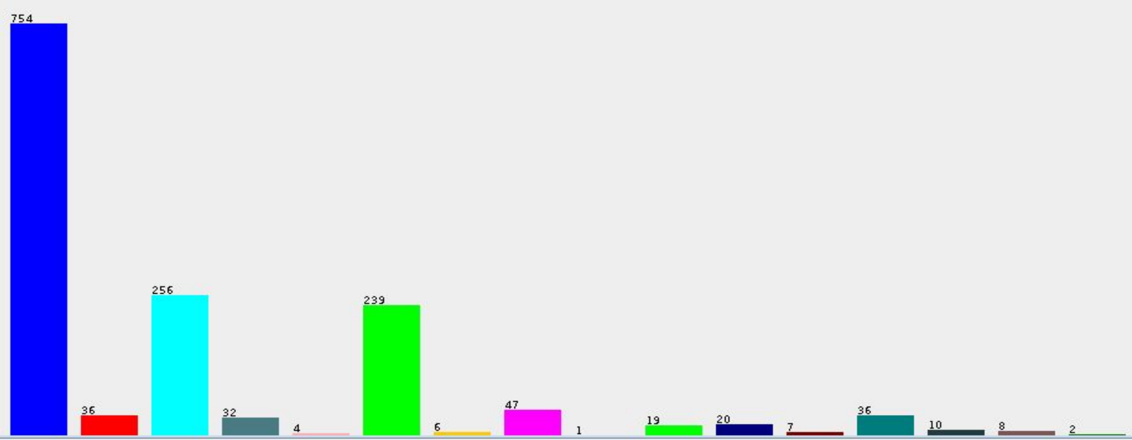
Selected attribute

Name: DISCHARGE_DESTINATION
Missing: 0 (0%)
Distinct: 16
Type: Nominal
Unique: 1 (0%)

No.	Label	Count
1	1	754
2	2	36
3	3	256
4	4	32
5	5	4
6	6	239
7	7	6
8	20	47
9	43	1
10	50	19
11	51	20
12	61	7
13	62	36
14	63	10
15	65	8
16	70	2

Class: DISCHARGE_DESTINATION (Nom)

Visualize All



Task 2.

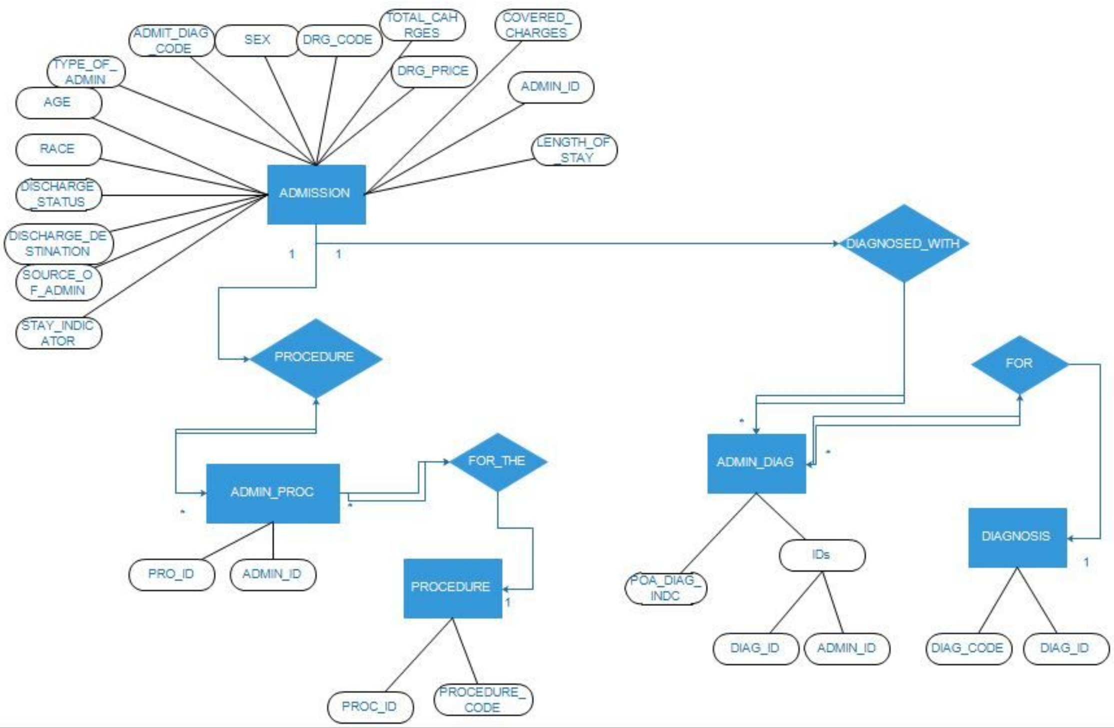
1. Relational Schema

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 ;
```

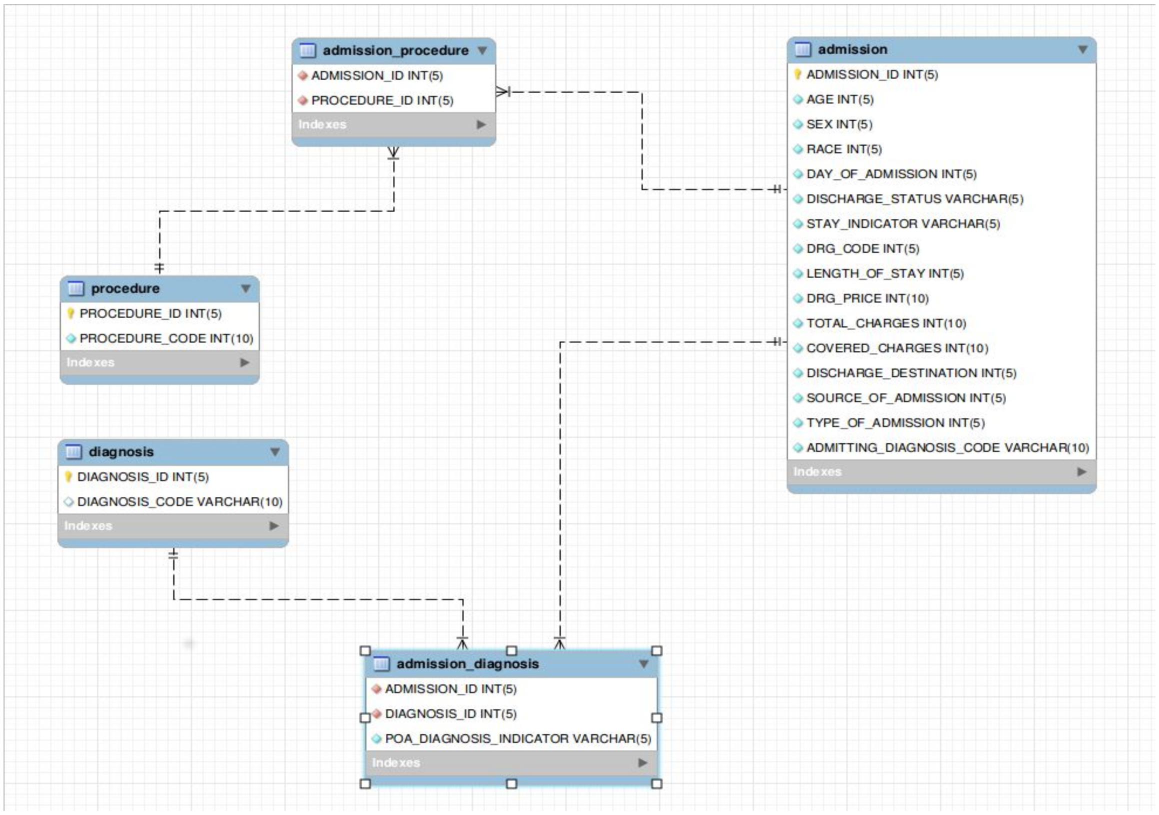
```
CREATE TABLE IF NOT EXISTS `procedure` (
   `PROCEDURE_ID` int(5) NOT NULL AUTO_INCREMENT,
   `PROCEDURE_CODE` int(10) NOT NULL,
   PRIMARY KEY (`PROCEDURE_ID`)
) ENGINE=InnoDB    DEFAULT CHARSET=LATIN1 AUTO_INCREMENT=385 ;

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.



2. 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
```

3. *SQL queries*

a. 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;
```

b. 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%.

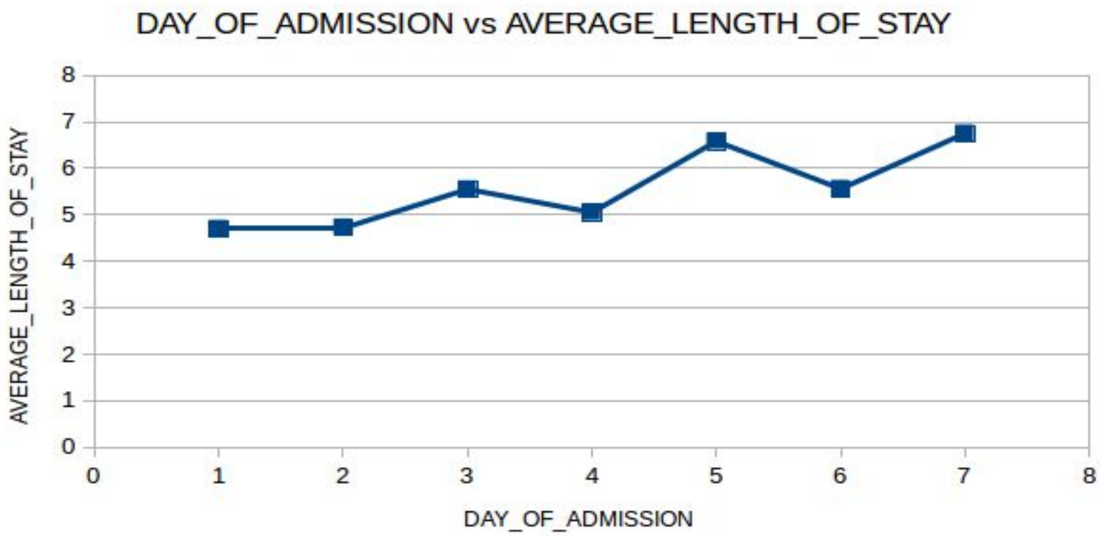
c. Variation and Length of Stay

We have used the following SQL query for this task.

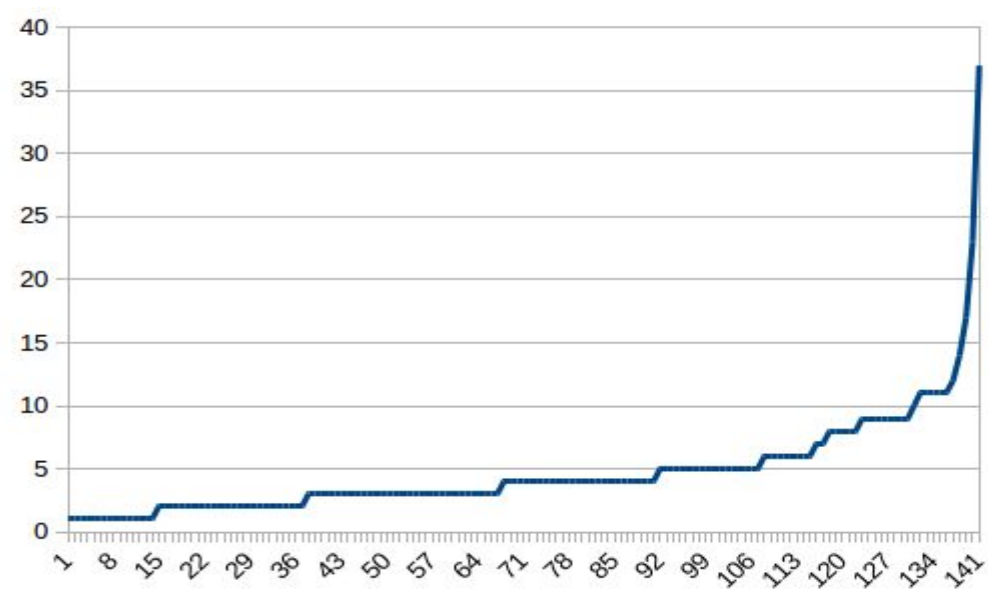
```
SELECT
    DAY_OF_ADMISSION, AVG(LENGTH_OF_STAY) AS AVERAGE
FROM
    ADMISSION
GROUP BY DAY_OF_ADMISSION
```

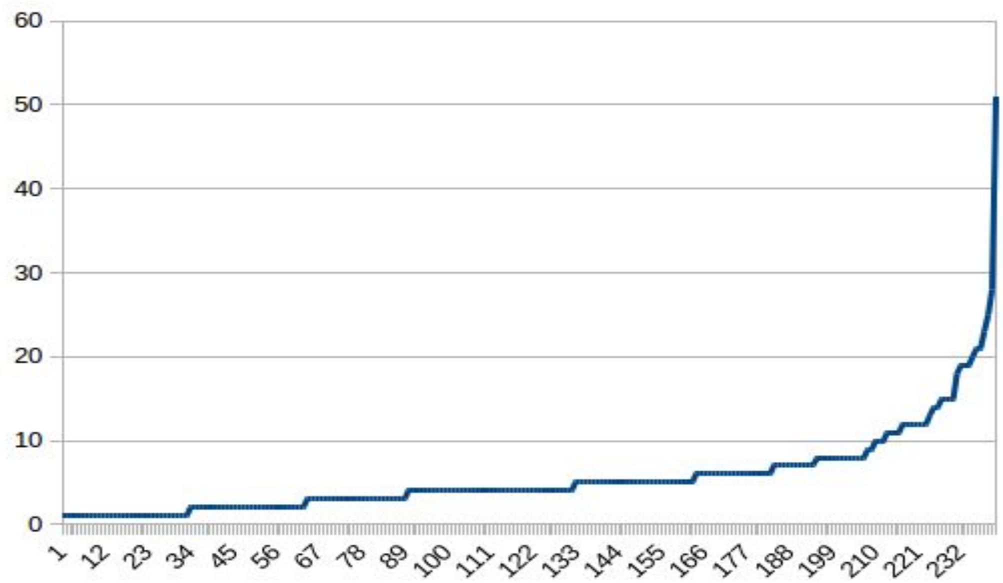
The patients admitted on day 5 (Thursday) and day 7 (Saturday) have stayed longer in the hospital.

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



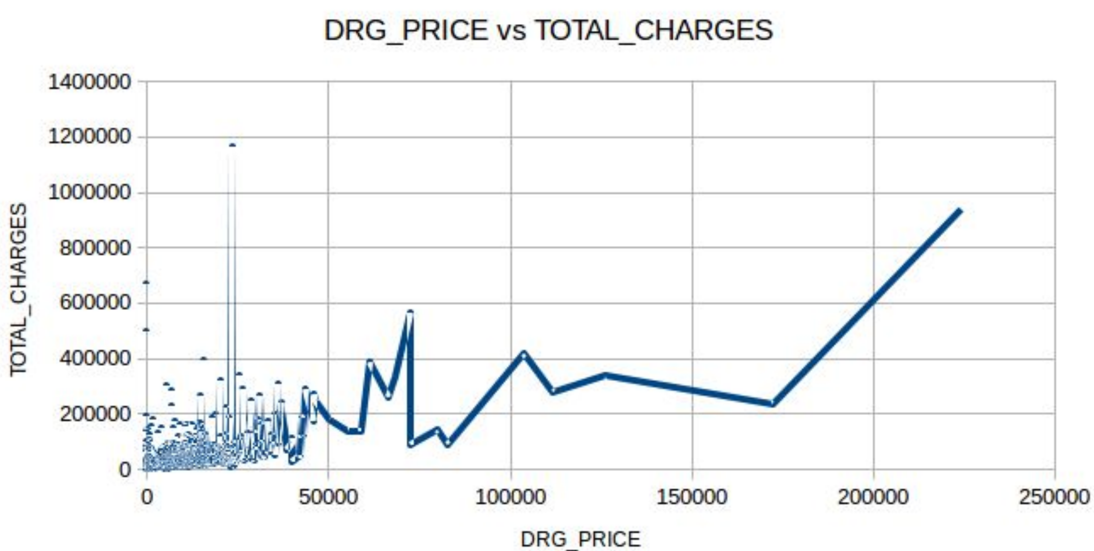
Average patients admitted on Friday have stayed longer than those admitted on Monday. Based on our web search and personal experience, we can say that Friday is more prone towards partying and traveling. This prevalence can give rise to more admissions than on Mondays. As per this web article 'Study: Higher risk of death for patients admitted to hospitals on weekends', 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.





d. 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)

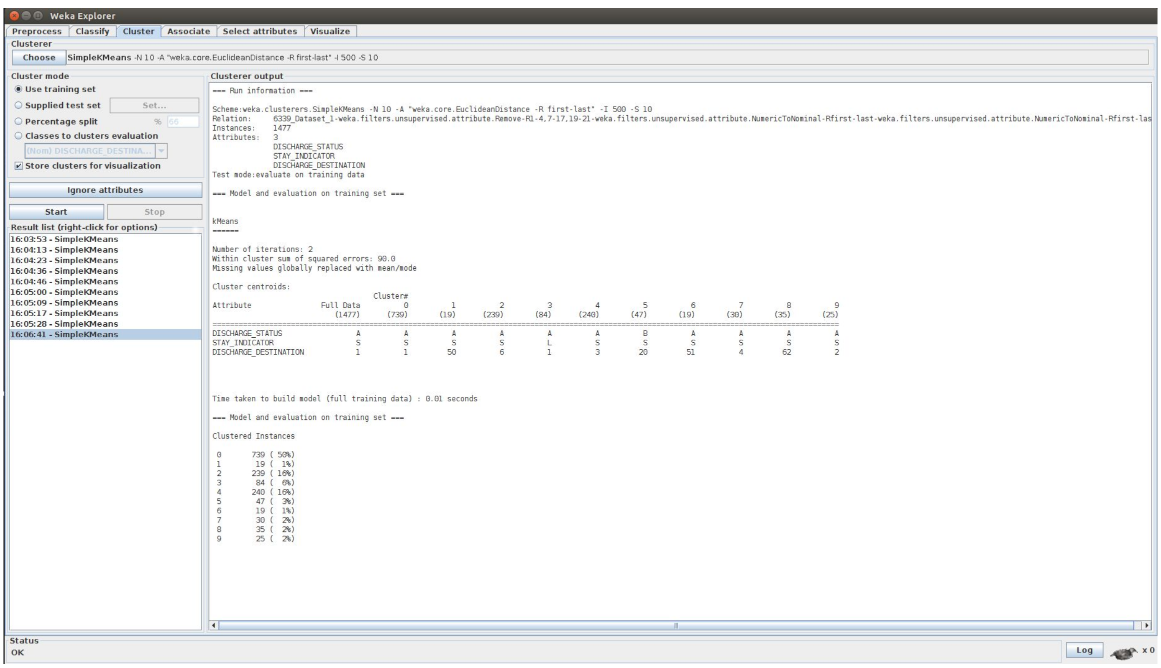


Task 3.

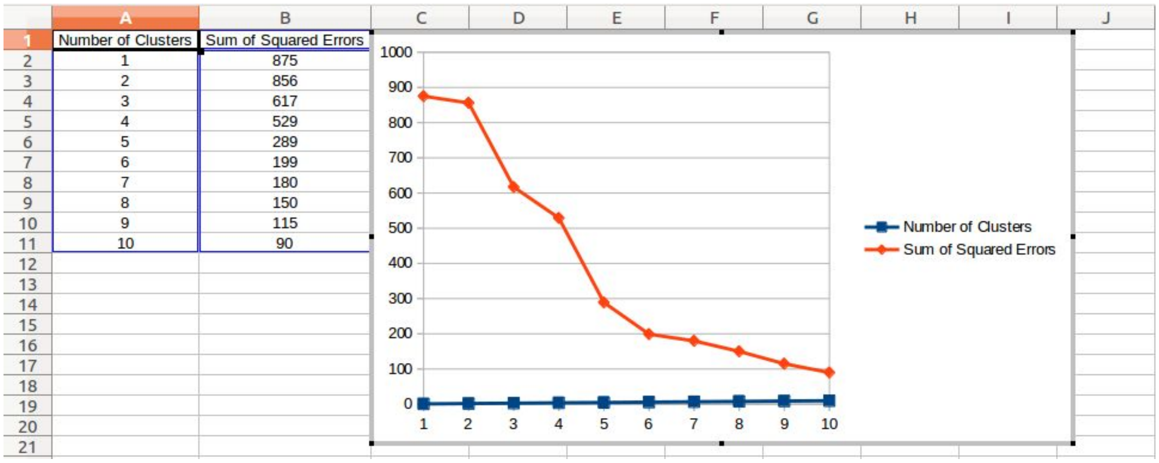
A. Appropriate Number of Clusters

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.

a) Select DISCHARGE_DESTINATION, DISCHARGE_STATUS and STAY_INDICATOR
b) Deselect all other attributes
c) Change types to nominal
d) Go to clusters tab and select SimpleKMeans
e) Note the Sum of Squared Error values
f) 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.



B. 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

C. *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.

D. *Supervised vs Unsupervised*

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

a) We change the type of the data

b) KMeans clustering works on non-labeled data.

c) Clustering is always a part of unsupervised learning.

Task 4.

a. Method to find DRG_PRICE_BINARY

Changing DRG_PRICE to binary values

a) Add a column next to DRG_PRICE
b) Add condition IF(DRG_PRICE>80000,1,0)
c) If DRG_PRICE is greater than $80000, value is set as 1
d) If DRG_PRICE is lesser than $80000, value is set as 0
e) Copy paste the formula to all the cells in the column

b. Clinicians/Administrators know the following details

1. When the patient enters the hospital
    1. AGE
    2. SEX
    3. RACE
    4. DAY_OF_ADMISSION
    5. SOURCE_OF_ADMISSION
    6. TYPE_OF_ADMISSION
    7. 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

c. *Scenarios*

1. Exclude 7, 10 and 11.

DRG_CODE, TOTAL_CHARGES and COVERED_CHARGES will be unknown at the time of admission and during hospital stay.

2. CfsSubsetEval

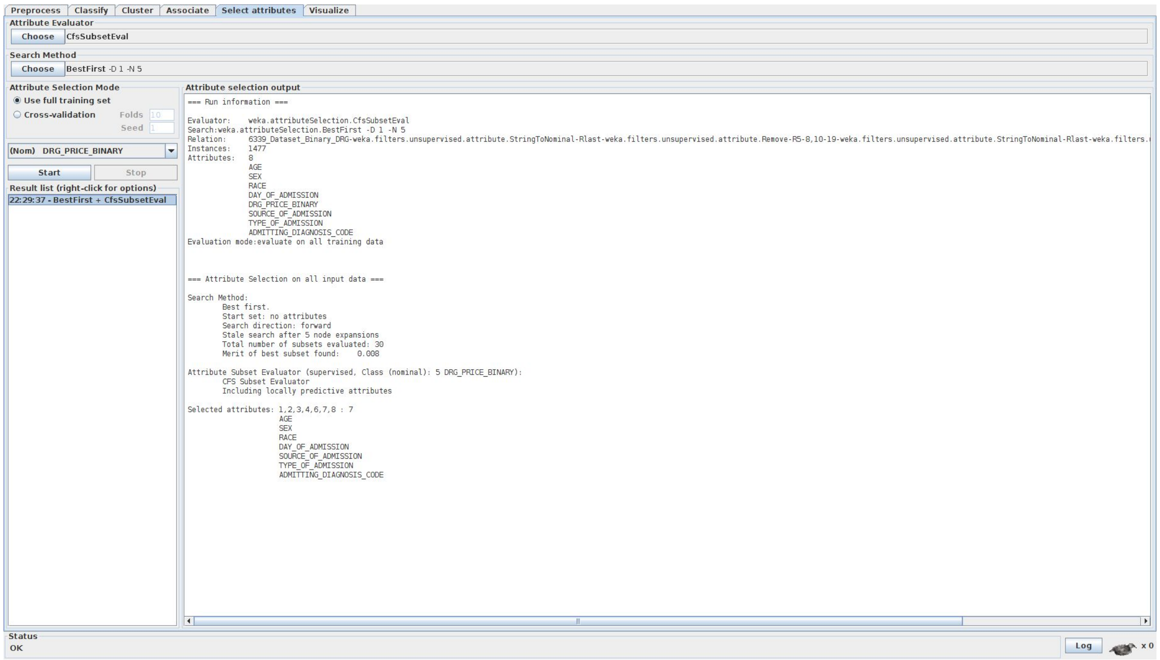
CfsSubsetEval and BestFirst are used to select the features. Following is the process

i. Select the required attributes
ii. Remove the rest of the attributes
iii. Convert string values to nominal
iv. 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.

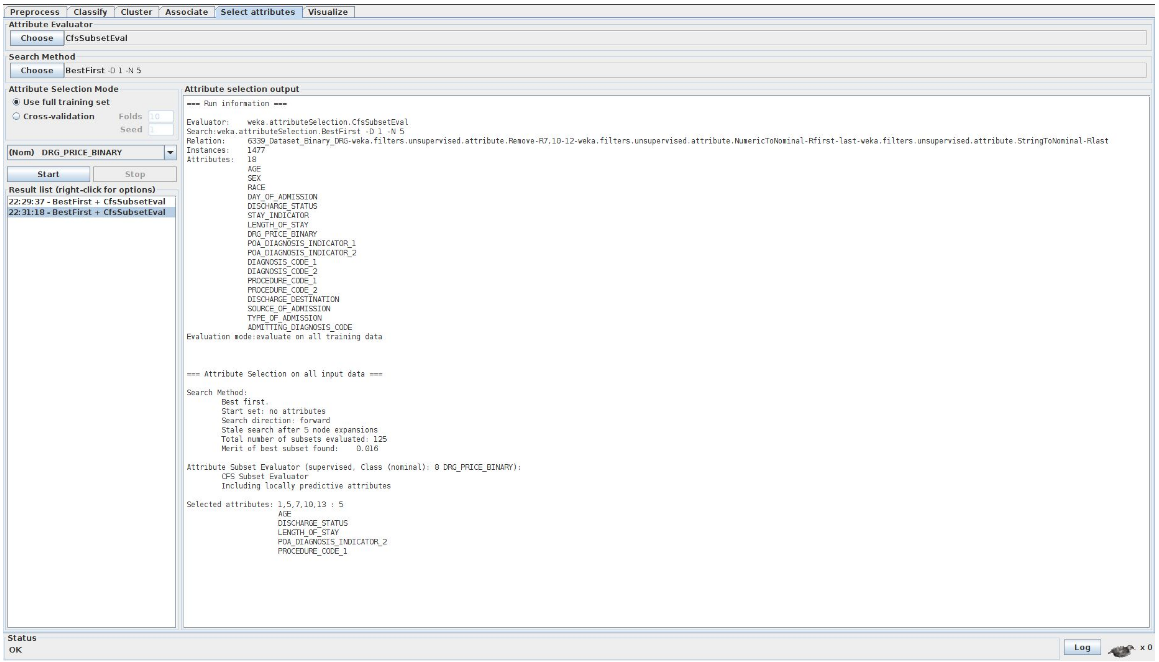
Features selected - AGE, SEX, RACE, DAY_OF_ADMISSION, SOURCE_OF_ADMISSION, TYPE_OF_ADMISSION and ADMITTING_DIAGNOSIS_CODE.

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.



3. Classifiers

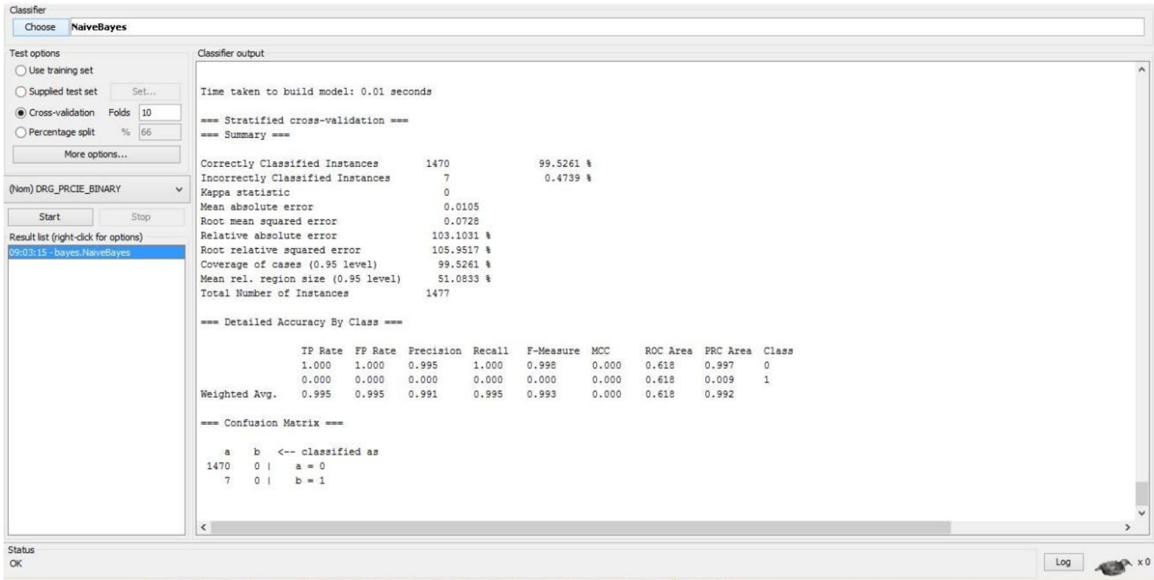
i. Naive Bayes

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.

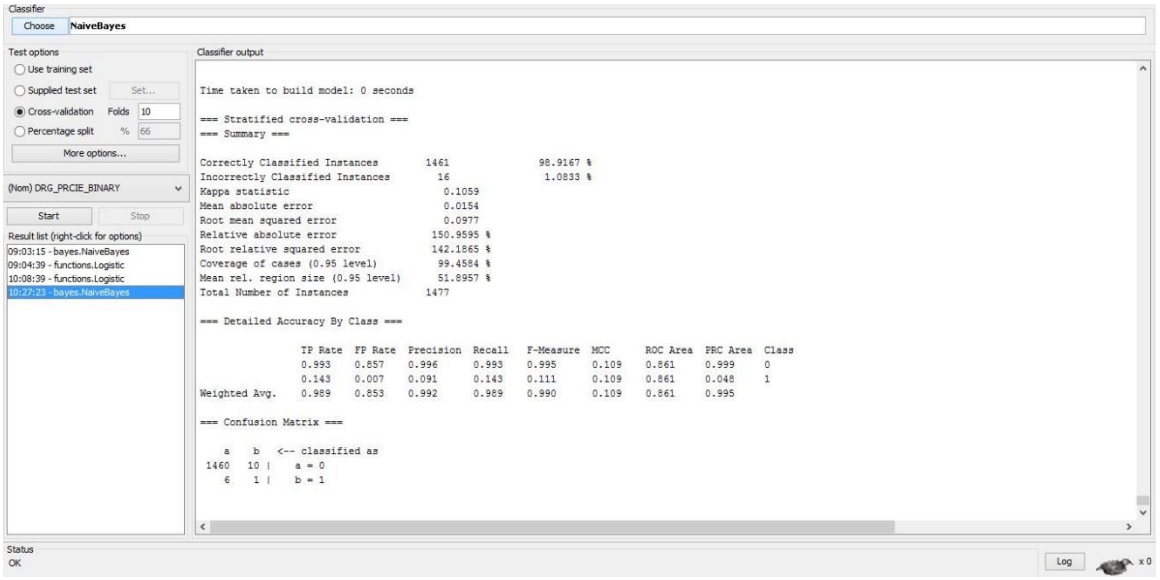
99.52% were correctly classified.

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.



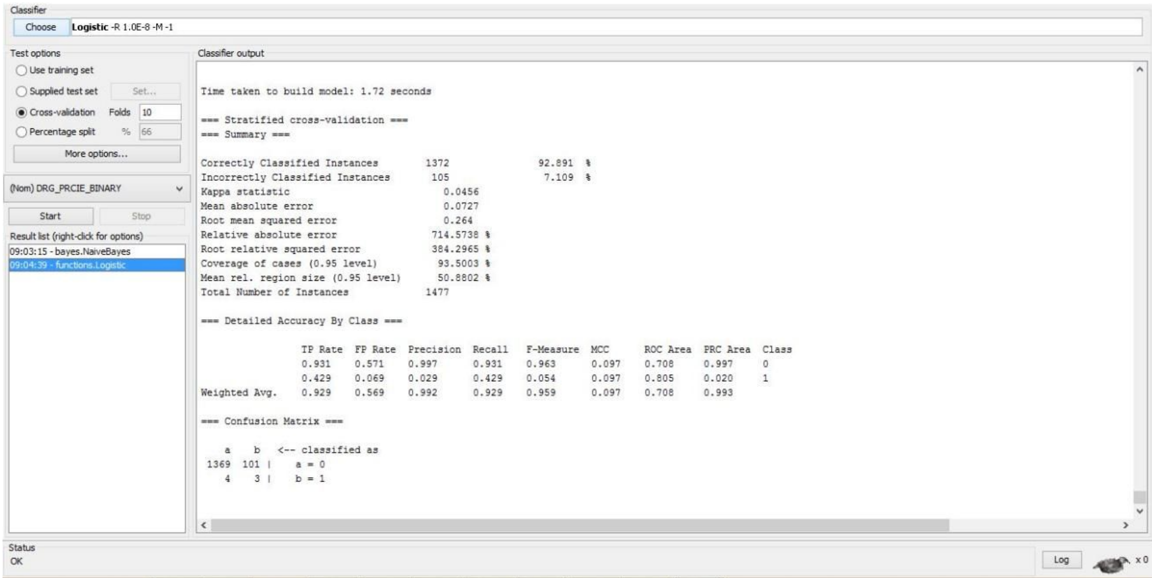
ii. Logistic Regression

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.

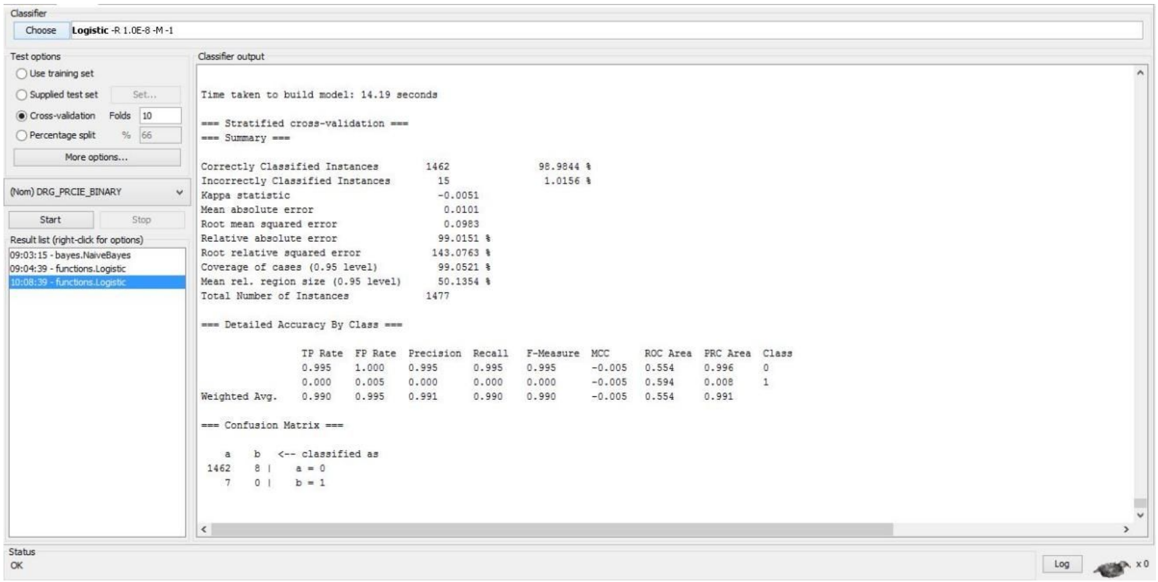
7.10% were incorrectly classified.



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.



4. Accuracy

a. Overall Accuracy

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

a. Naive Bayes
   i. Scenario 1 - 99.5261%
   ii. Scenario 2 - 98.849%

b. Logistic
   i. Scenario 1 - 92.2139%
   ii. Scenario 2 - 99.1198%

b. Greater than $80000

a. 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

b. 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

c. Less than $80000

c. 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

d. 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