Report for Assignment 1

Weijia Sun

Note: There is a page break before each question

Question 2

(a) [1pt] Give a brief summary of the patient's demographics (race, age, marital status, etc).

From the SQL script below, we can know:
Gender: female
Age: 79 (dob: 2027-08-04, dod: 2106-06-14)
Race: black/ African American
Marital status: widowed
Insurance: Medicaid
Language: Hait
Religion: Unobtainable

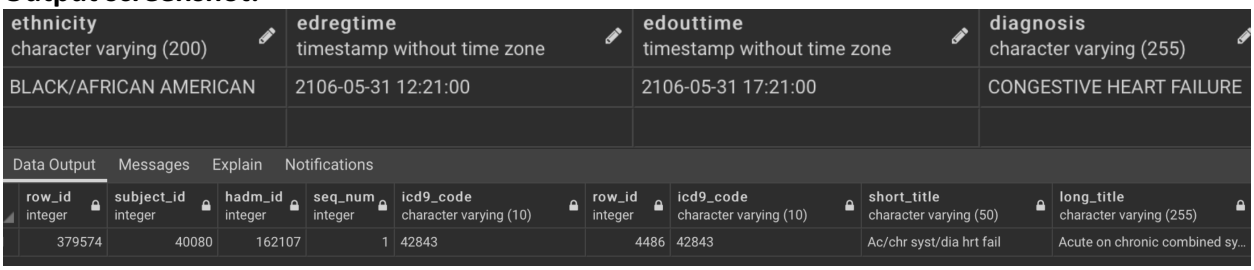
SQL script:

```
SELECT *
FROM patients
WHERE patients.subject_id = 40080
SELECT *
FROM admissions
WHERE subject_id = 40080
```

(b) [1pt] What was the patient's primary diagnosis (seq_num = 1) and the ICD-9 code?

ICD-9 code: 42843
Short title: Ac/chr syst/dia hrt fail
Long title: Acute on chronic combined systolic and diastolic heart failure

Output screenshot:



ethnicity character varying (200)	edregtime timestamp without time zone	edouttime timestamp without time zone	diagnosis character varying (255)
BLACK/AFRICAN AMERICAN	2106-05-31 12:21:00	2106-05-31 17:21:00	CONGESTIVE HEART FAILURE

Data Output | Messages | Explain | Notifications

row_id integer	subject_id integer	hadm_id integer	seq_num integer	icd9_code character varying (10)	row_id integer	icd9_code character varying (10)	short_title character varying (50)	long_title character varying (255)
379574	40080	162107	1	42843	4486	42843	Ac/chr syst/dia hrt fail	Acute on chronic combined sy...

SQL script:

```
SELECT *
FROM diagnoses_icd JOIN d_icd_diagnoses
ON diagnoses_icd.icd9_code = d_icd_diagnoses.icd9_code
WHERE diagnoses_icd.subject_id = 40080 and diagnoses_icd.seq_num
= 1
```

(c) [1pt] How long did the patient stay in the ICU? According to the discharge report, what was her condition when she was discharged?

How long: 4.8577 days (los: 4.8577, intime: 2106-05-31, outtime: 2106-06-05)

Discharge condition:

Mental Status: Minimally clear and coherent.

Level of Consciousness: Minimally Alert and somewhat

interactive.

Activity Status: Bed bound - dependent hemiplegia.

SQL script:

```
SELECT *
FROM icustays
WHERE subject_id = 40080

SELECT *
FROM noteevents
WHERE category = 'Discharge summary' and subject_id = 40080
```

(d) [1pt] What was the patient's highest and lowest heart rates during the stay?

Highest: 141
Lowest: 80
SQL script:

```
SELECT MAX(valuenum)
FROM chartevents
WHERE itemid = 220045 and subject_id = 40080

SELECT MIN(valuenum)
FROM chartevents
WHERE itemid = 220045 and subject_id = 40080
```

Question 3

(a) Create a histogram plot of the distribution of patient ages in years. Note that you may have to merge tables. You may notice some patients with abnormal or nonsensical ages due to the neonatal ICU and the HIPAA-compliant obscuring of ages for people over 90. What does this plot tell us about our patients?

Data Output | Messages | Explain | N

	age_group text	numberofpatients bigint
1	60+	7840
2	neonate	7874
3	50+	6675
4	80+	5863
5	70+	7941
6	20+	1706
7	40+	4161
8	30+	2088
9	>89	1991
10	10+	381

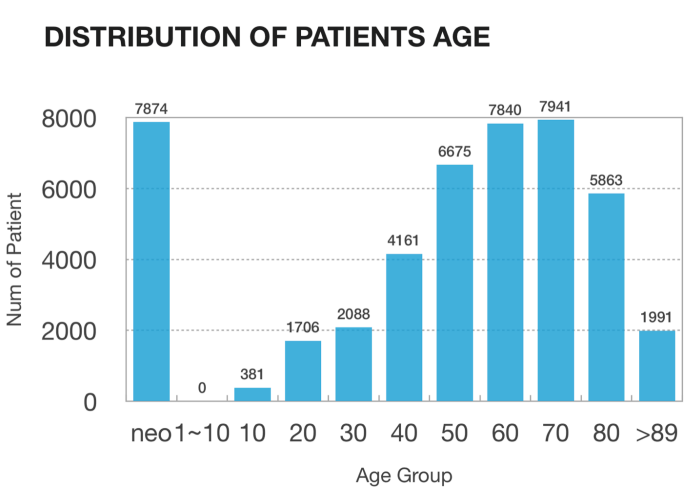


Figure 1. Left: the data output after querying
Right: the histogram of the distribution of patient ages in years

This plot tells us:
The patients are from neonate group and 50~80 age group.

Please note:
Since all ages > 89 in the database were replaced with 300 in MIMIC-III, in this question, ages in the output data will be assigned to “>89” if the age is “>100” in MIMIC-III.

SQL script:

```
SELECT p.subject_id, p.dob, a.hadm_id,
    a.admittime, p.expire_flag
FROM admissions a
INNER JOIN patients p
ON p.subject_id = a.subject_id;

SELECT p.subject_id, p.dob, a.hadm_id,
    a.admittime, p.expire_flag,
    MIN (a.admittime) OVER (PARTITION BY p.subject_id) AS first_admittime
FROM admissions a
INNER JOIN patients p
ON p.subject_id = a.subject_id
ORDER BY a.hadm_id, p.subject_id;

WITH first_admission_time AS
(
  SELECT
      p.subject_id, p.dob, p.gender
      , MIN (a.admittime) AS first_admittime
      , MIN( ROUND( (cast(admittime as date) - cast(dob as date)) / 365.242,2) )
          AS first_admit_age


  FROM patients p
  INNER JOIN admissions a
  ON p.subject_id = a.subject_id
  GROUP BY p.subject_id, p.dob, p.gender
  ORDER BY p.subject_id
)
, age as
(
SELECT
      subject_id, first_admit_age
      , CASE
          -- all ages > 89 in the database were replaced with 300
          -- we check using > 100 as a conservative threshold to ensure we capture all these patients
          WHEN first_admit_age > 100
              then '>89'
              WHEN first_admit_age > 80
              then '80+'
                  WHEN First_admit_age > 70
              then '70+'
                  WHEN First_admit_age > 60
              then '60+'
                  WHEN First_admit_age > 50
              then '50+'
                  WHEN First_admit_age > 40
              then '40+'
                  WHEN First_admit_age > 30
              then '30+'
                  WHEN First_admit_age > 20
              then '20+'
                  WHEN First_admit_age > 10
               then '10+'
                  WHEN First_admit_age > 1
              then '1~10'
          WHEN first_admit_age <= 1
               THEN 'neonate'
          END AS age_group
  FROM first_admission_time
)
SELECT age_group
  , count(subject_id) as NumberOfPatients
from age
group by age_group
```

(b) [2pts] Create a histogram plot of the distribution of patient heart rates. Use only the first occurrence of each patient's heart rate. What does this plot tell us about our patients?

This plot tells us:

50~100 covers most patients (which is actually the range of normal heart rate).

Data Output | Messages | Explain | Noti

	heart_rate_group text	numberofpatients bigint
1	0~40	62
2	100+	1759
3	110+	823
4	120+	359
5	130+	132
6	140+	62
7	40+	205
8	50+	1025
9	60+	2798
10	70+	3910
11	80+	3716
12	90+	2863

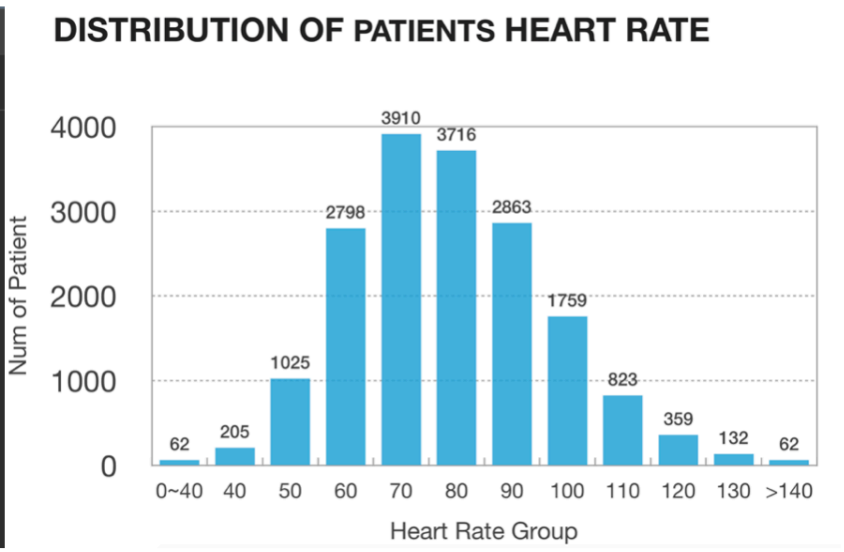


Figure 2. Left: the data output after querying
Right: the histogram of the distribution of patient heart rate

SQL script:

```
With sta AS(
     SELECT distinct on(subject_id) chartevents.valuenum
     FROM  chartevents
     WHERE itemid = 220045
     )
-- order by subject_id asc

, heart_rate as(
     SELECT valuenum
            ,CASE
                 WHEN valuenum > 140
                      then '140+'
                 WHEN valuenum > 130
                      then '130+'
                 WHEN valuenum > 120
                      then '120+'
                 WHEN valuenum > 110
                      then '110+'
                 WHEN valuenum > 100
                      then '100+'
                 WHEN valuenum > 90
                      then '90+'
                 WHEN valuenum > 80
                      then '80+'
                 WHEN valuenum > 70
                      then '70+'
                 WHEN valuenum > 60
                      then '60+'
                 WHEN valuenum > 50
                      then '50+'
                 WHEN valuenum > 40
                      then '40+'
                 WHEN valuenum >= 0
                      then '0~40'

                 END AS heart_rate_group
      FROM sta
)

SELECT heart_rate_group
  , count(valuenum) as NumberOfPatients
from heart_rate
group by heart_rate_group
```

(c) [2pts] Create a scatter plot of patient heart rate vs age. Use cutoff values of [0-400] for age and [30-250] for heart rate to remove outliers. What does this plot tell us about our patients?

Please note:
For heart rates lower than 30 in MIMIN-III, they are set to 30 in the scatter plot for the aesthetics of the plot; for age is 300, they are set to 89 in the plot.

This plot tells us:
The range of heart rates doesn't change a lot among each age group. But the elder people have slightly broader range of heart rates.

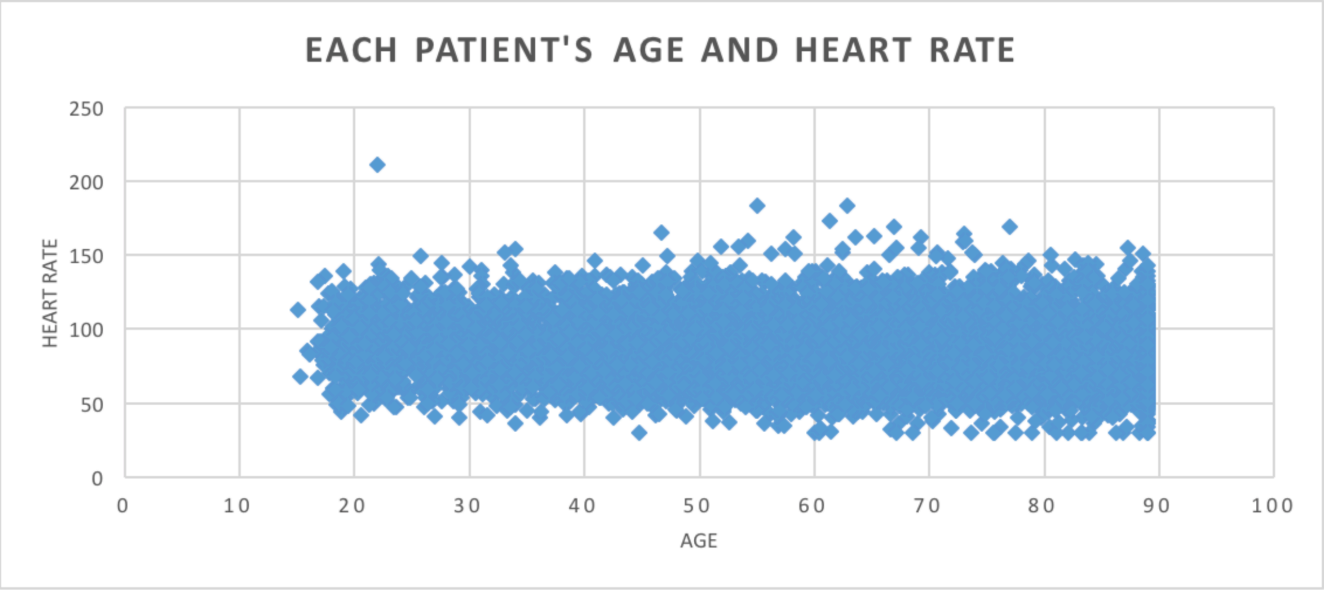


Figure 3. The scatter plot of patient heart rate vs age.

SQL script:

```
CREATE TABLE age AS
    SELECT p.subject_id as subject_id,
          MIN( ROUND( (cast(admittime as date) - cast(dob as
date)) / 365.242,2) ) AS first_addmission_age
     FROM patients p
     INNER JOIN admissions a
     ON p.subject_id = a.subject_id
     GROUP BY p.subject_id

UPDATE age
SET first_addmission_age = 89
WHERE first_addmission_age = 300;
---------------------------
CREATE TABLE heart_rate AS
    SELECT distinct on(subject_id) chartevents.subject_id as
subject_id, chartevents.valuenum as valuenum
     FROM  chartevents
     WHERE itemid = 220045

UPDATE heart_rate
SET valuenum = 30
WHERE valuenum < 30;
--------------------
SELECT age.first_addmission_age, heart_rate.valuenum
FROM age JOIN heart_rate
ON age.subject_id = heart_rate.subject_id
```

Question 4

(a) Train a logistic regression model (using scikit-learn) to predict in-ICU mortality from all of the variables. Use L2 regularization.

Thoughts on doing this task:

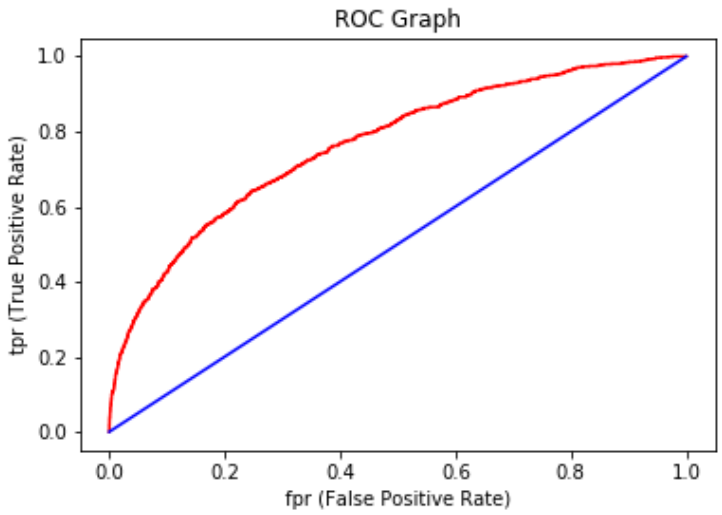
In the hint given, to standardize the non-binary variables is needed. However, the way of standardization is not given. I came up with two methods to do that and will show both results.

The first method to use min-max normalization, which can normalize non-binary data to data range from 0 to 1. The reason for doing so is to keep non-binary data as the same range as binary data. The second method is to scale the non-binary data to have zero unit and unit variance, which is also known as z-normalization.

Personally, I believe the first method makes more sense. Min-max normalization is fairer to binary data since in this case, all data will have a range from 0 to 1, and the coefficient of each feature won't be biased due to their significant difference. If all data don't have a uniform range, the coefficient won't be so convinced for rating the importance of features.

Method 1: Min-max Normalization

- **Plot the ROC graph and compute the AUC score.**



AUC score: 0.7653165425420847

- **Comment on the model performance.**

Here is the performance of the model:

```
[9]: y_predict = clf.predict(X_test)
     y_predict_prob = clf.predict_proba(X_test)

     # comment on model performance
     print("Score on training set: ", clf.score(X_train,y_train))
     print("Score on testing set: ", clf.score(X_test,y_test))
     print("Accuracy score:",metrics.accuracy_score(y_test, y_predict))
     print("No. of iterations to converge: ", clf.n_iter_)

     Score on training set:  0.9056929269695227
     Score on testing set:  0.9058279596057914
     Accuracy score: 0.9058279596057914
     No. of iterations to converge:  [242]
```

We can see that the accuracy of the L2 logistic regression model can reach 90%. The performance on training set and testing is consistence with the final accuracy score.

- **Look at the top 5 risk factors of mortality and the lowest 5 and explain what they mean.**

The data output is the below screenshot:

```
[12]: risk_factors(X_train.columns, clf.coef_)

      All feature coefficients with signs:
      [[1.023671969456406, 'age'], [0.10053096566288583, 'first_hosp_stay'], [-0.13625273667916063, 'first_icu_stay'], [0.012526061499588978, 'adult_icu
      '], [-0.004676882308203735, 'eth_asian'], [-0.1847528440113108, 'eth_black'], [-0.4013445640192856, 'eth_hispanic'], [0.5625592193313363, 'eth_oth
      er'], [0.040741132507079135, 'eth_white'], [-0.74873427184475, 'admType_ELECTIVE'], [0.4811584842188535, 'admType_EMERGENCY'], [0.0, 'admType_NEWB
      ORN'], [0.280101849125522, 'admType_URGENT'], [-0.9723011256223312, 'heartrate_min'], [1.7764747733063142, 'heartrate_max'], [1.078992038200697, '
      heartrate_mean'], [-2.0508478371949503, 'sysbp_min'], [0.7109882164953113, 'sysbp_max'], [-0.9969299939771001, 'sysbp_mean'], [0.4995447653474141,
      'diasbp_min'], [-1.5855075456092582, 'diasbp_max'], [-0.8998524232898572, 'diasbp_mean'], [-1.0357257893537966, 'meanbp_min'], [0.3348208845681865
      , 'meanbp_max'], [1.412616199353262, 'meanbp_mean'], [-0.041712429709247115, 'resprate_min'], [0.38058518140772685, 'resprate_max'], [2.1359060620
      893957, 'resprate_mean'], [-3.2664498962268613, 'tempc_min'], [2.3629495031571, 'tempc_max'], [-2.272804092319191, 'tempc_mean'], [-1.138602434816
      08, 'spo2_min'], [-0.3776615659162428, 'spo2_max'], [-1.2922085330216053, 'spo2_mean'], [1.3643285888566667, 'glucose_min'], [0.01573022780942283,
      'glucose_max'], [0.07918455272201308, 'glucose_mean'], [1.77315089925302, 'aniongap'], [-1.9052839188200403, 'albumin'], [0.5916627492346583, 'bic
      arbonate'], [3.490983815092436, 'bilirubin'], [-1.692146734377437, 'creatinine'], [-0.8290401346319892, 'chloride'], [0.14824047465414944, 'glucos
      e'], [1.0258620750628862, 'hematocrit'], [-1.397059759740363, 'hemoglobin'], [2.564475927107407, 'lactate'], [0.39209826672205134, 'magnesium'], [
      0.5041561958756702, 'phosphate'], [-2.7284803609219233, 'platelet'], [-0.03219568964186391, 'potassium'], [0.1288258660472923 2, 'ptt'], [-0.110277
      01808216196, 'inr'], [1.0948620876152955, 'pt'], [0.3306964538523983, 'sodium'], [2.0249733104180105, 'bun'], [1.8774926630882627, 'wbc']]
      ------------------------------
      Sort the value itself:
      Top 5 risk factors:
      2.0249733104180105 bun
      2.1359060620893957 resprate_mean
      2.3629495031571 tempc_max
      2.564475927107407 lactate
      3.490983815092436 bilirubin
      Lowest 5 risk factors:
      -3.2664498962268613 tempc_min
      -2.7284803609219233 platelet
      -2.272804092319191 tempc_mean
      -2.0508478371949503 sysbp_min
      -1.9052839188200403 albumin
      ------------------------------
      Sort using absolute value:
      Top 5 risk factors:
      2.3629495031571 tempc_max
      2.564475927107407 lactate
      2.7284803609219233 platelet
      3.2664498962268613 tempc_min
      3.490983815092436 bilirubin
      Lowest 5 risk factors:
      0.0 admType_NEWBORN
      0.004676882308203735 eth_asian
      0.012526061499588978 adult_icu
      0.01573022780942283 glucose_max
      0.03219568964186391 potassium
```

The first part is the coefficients of all features.

The second part of the result is sorted by the numerical value. It is not the final result since positive signs and negative signs represent the positive influence and negative influence. So, I sorted according to the absolute value in the third part, and arrange the result in the following two tables:

Rank(top)	Feature	Influence	Value	Meaning
1	bilirubin	+	3.490983815092436	High levels of bilirubin indicate liver inefficiency
2	tempc_min	-	3.2664498962268613	The minimum body temperature of a patient has a negative influence.
3	platelet	-	2.7284803609219233	Platelet content has a negative influence.
4	lactate	+	2.564475927107407	Lactate content has a positive influence.
5	tempc_max	+	2.3629495031571	The maximum body temperature of a patient has a positive influence.

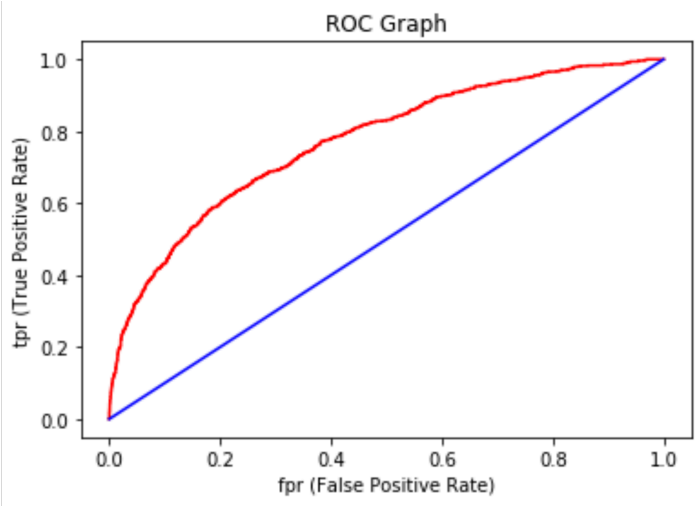
Table 1. Top 5 risk factors of mortality min-max normalization

Rank(low)	Feature	Influence	Value	Meaning
1	admType_NEWBORN	N/A	0.0	Whether the patient is new born or not doesn't influence the mortality.
2	eth_asian	-	0.004676882308203735	Whether the patient is Asian or not
3	adult_icu	+	0.012526061499588978	The patient is an adult and is sent to ICU
4	glucose_max	+	.01573022780942283	Maximum value of the glucose across all the readings. The glucose level could be controlled for the patients.
5	potassium	-	0.03219568964186391	Potassium content in the patient's body

Table 2. Lowest 5 risk factors of mortality using min-max normalization

Method 2: Z-Normalization

- **Plot the ROC graph and compute the AUC score.**



AUC score: 0.7722211174195345

We can see that the AUC score after using z-normalization is slightly better than min-max normalization. But I still believe min-max normalization makes more sense, since AUC score doesn't mean everything.

- **Comment on the model performance.**

```
# comment on model performance
print("Score on training set: ", clf.score(X_train,y_train))
print("Score on testing set: ", clf.score(X_test,y_test))
print("Accuracy score:",metrics.accuracy_score(y_test, y_predict))
print("No. of iterations to converge: ", clf.n_iter_)
```

```
Score on training set:  0.9055883736734801
Score on testing set:  0.9059496289086264
Accuracy score: 0.9059496289086264
No. of iterations to converge:  [335]
```

The accuracy score is nearly same as method 1, reaching 90%.

- **Look at the top 5 risk factors of mortality and the lowest 5 and explain what they mean.**

The data output is the below screenshot:

```
risk_factors(X_train.columns, clf.coef_)

Sort the value itself:
Top 5 risk factors:
0.37875863225772155 hematocrit
0.3828319298999089 meanbp_mean
0.4883912423173155 admType_EMERGENCY
0.549568448052315 eth_other
2.4214819769009717 glucose_mean
Lowest 5 risk factors:
-0.758243338699275 admType_ELECTIVE
-0.44324607990759207 hemoglobin
-0.4280551726999256 eth_hispanic
-0.25220091466653477 diasbp_mean
-0.22779092328432035 sysbp_min
------------------------------
Sort using absolute value:
Top 5 risk factors:
0.44324607990759207 hemoglobin
0.4883912423173155 admType_EMERGENCY
0.549568448052315 eth_other
0.758243338699275 admType_ELECTIVE
2.4214819769009717 glucose_mean
Lowest 5 risk factors:
0.0 admType_NEWBORN
0.0002404015328206492 eth_asian
0.004080589065291331 potassium
0.007562552592888492 ptt
0.00830997109291142 resprate_min
```

The first part of the result is sorted by the numerical value. It is not the final result since positive signs and negative signs represent the positive influence and negative influence. So, I sorted according to the absolute value in the second part, and arrange the result in the following two tables:

Rank(top)	Feature	Influence	Value	Meaning
1	glucose_mean	+	2.421481976900917	Mean value of the glucose across all the readings.
2	admType_ELECTIVE	-	0.758243338699275	A type of patients being admitted has negative impact.
3	eth_other	+	0.549568448052315	Other ethnics that aren't listsed have positive impact to results.
4	admType_EMERGENCY	+	0.488391242317315 5	Patients being admitted with emergency condition has positive impact.

5	hemoglobin	-	0.443246079907592 07	The level of hemoglobin has negative influence.

Table 3. Top 5 risk factors of mortality using z-normalization

Rank(low)	Feature	Influence	Value	Meaning
1	admType_NEWBORN	N/A	0.0	Whether the patient is new born or not doesn't influence the mortality.
2	eth_asian	+	0.004676882308203735	Whether the patient is Asian or not
3	potassium	-	0.012526061499588978	Potassium content in the patient's body
4	ptt	+	.01573022780942283	Partial Thromboplastin Time. It indicates the body's capacity to form clots of blood. High ptt values may increase mortality rates.
5	resprate_min	-	0.03219568964186391	Resperate is related to blodd pressure.

Table 4. Lowest 5 risk factors of mortality using z-normalization

We can see that the largest coefficient is around 2.42, but the second largest is around 0.76. The difference between these two values is larger than using min-max normalization. In this case, the level of glucose matters a lot in mortality.

(b) Report the top 5 words associated with a high risk of mortality and the lowest 5. Plot the ROC graph and compute the AUC score.

This question is similar to the last one. I used the absolute value of coefficients of each value to compare.

Rank	Coefficient	Word
1	4.113160415687661	family
2	3. 325859708735898	arrest
3	3. 1947298938884385	worsening
4	2. 8786570599398282	dnr
5	2. 8581326179529087	extubation

Table 5. Top 5 risk factors of doctor notes

Rank	Coefficient	Word
1	2.04396812743609e-06	cpme
2	2.217785536195768e-06	hco
3	2.8723165753689643e-06	b19
4	6.238605940186612e-06	recvd
5	6.295361750354964e-06	8898

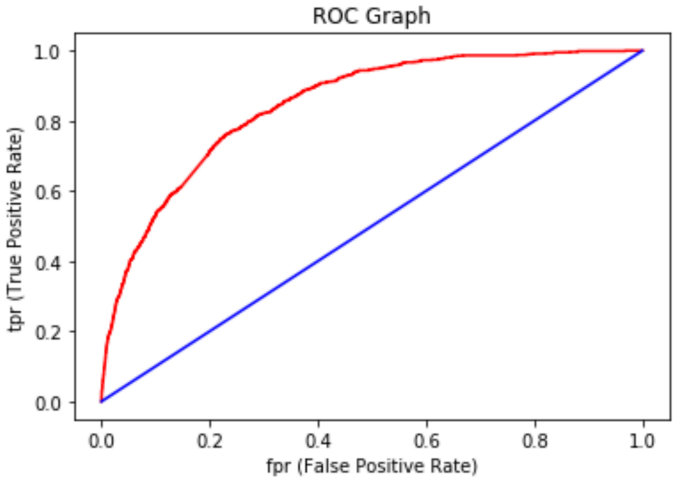
Table 6. Lowest 5 risk factors of doctor notes

Data output screenshot:

```
]: risk_factors(vectorizer.get_feature_names(), clf.coef_)

   ------------------------------
   Sort the value itself:
   Top 5 risk factors:
   2.66196727652265 cmo
   2.8786570599398282 dnr
   3.1947298938884385 worsening
   3.325859708735898 arrest
   4.113160415687661 family
   Lowest 5 risk factors:
   -2.8581326179529087 extubation
   -2.4754993146024935 clear
   -2.2323336884463436 pain
   -2.0231132913932117 extubated
   -1.9568480183601864 wean
   ------------------------------
   Sort using absolute value:
   Top 5 risk factors:
   2.8581326179529087 extubation
   2.8786570599398282 dnr
   3.1947298938884385 worsening
   3.325859708735898 arrest
   4.113160415687661 family
   Lowest 5 risk factors:
   2.04396812743609e-06 cpme
   2.217785536195768e-06 hco
   2.8723165753689643e-06 b19
   6.238605940186612e-06 recvd
   6.295361750354964e-06 8898
```

ROC:



AUC score: 0.845795093350303

(c) Combine the models in parts 2a and 2b to predict mortality using both clinical notes and tabular data. Plot the ROC graph and compute the AUC score. Comment on how the two sources of data affect the model.

Result:

Top 5 risk factors:

2.2424700289038215 cabg (from notes)

2.534415656282369 worsening (from notes)

2.582782969036912 glucose_mean (measurement)

3.2199268466864206 arrest (from notes)

3.655053711738844 family (from notes)

Lowest 5 risk factors:

0.0 admType_NEWBORN (measurement)

1.0715497620909016e-06 4sec (from notes)

1.2259930200710172e-06 negligible (from notes)

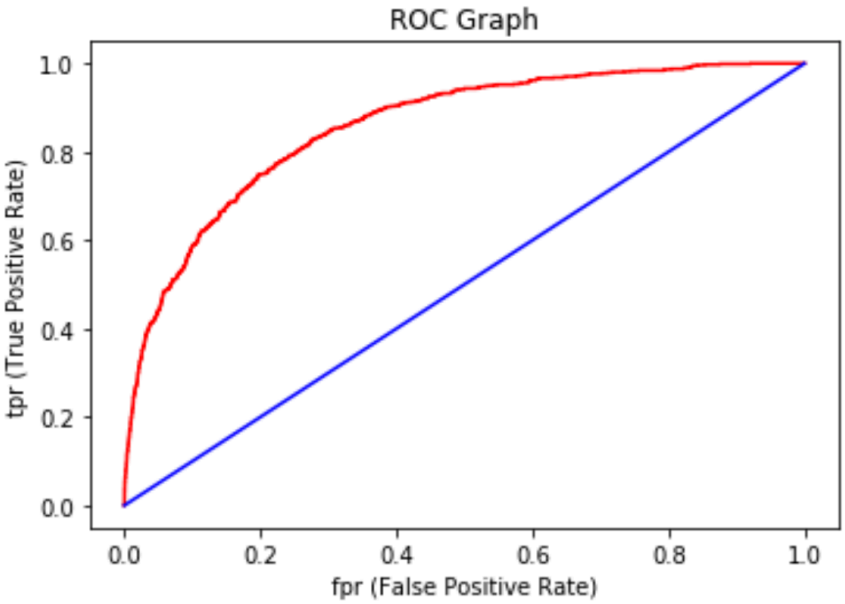
1.594405960994907e-06 varicosities (from notes)

3.007196557574272e-06 precaution (from notes)

Data output screenshot:

```
risk_factors(list(X_train_icu.columns) + list(tfidf_vectorizer.get_feature_names()), clf.coef_)
------------------------------
Sort the value itself:
Top 5 risk factors:
2.231258674499076 icp
2.534415656282369 worsening
2.582782969036912 glucose_mean
3.2199268466864206 arrest
3.655053711738844 family
Lowest 5 risk factors:
-2.2424700289038215 cabg
-2.097977441549197 extubation
-2.0235543310693096 wean
-1.8989294032354649 sepsis
-1.8909768890343137 trach
------------------------------
Sort using absolute value:
Top 5 risk factors:
2.2424700289038215 cabg
2.534415656282369 worsening
2.582782969036912 glucose_mean
3.2199268466864206 arrest
3.655053711738844 family
Lowest 5 risk factors:
0.0 admType_NEWBORN
1.0715497620909016e-06 4sec
1.2259930200710172e-06 negligible
1.594405960994907e-06 varicosities
3.007196557574272e-06 precaution
```

ROC:

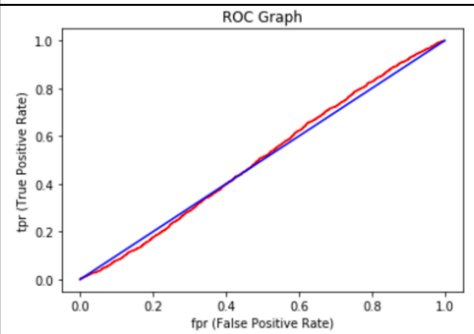
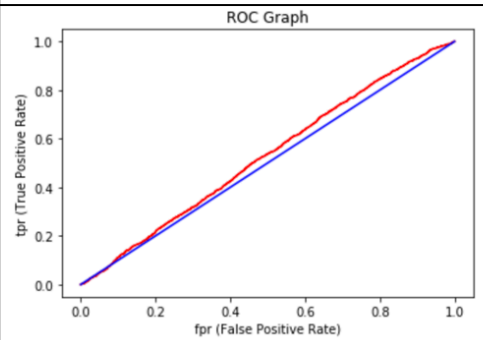
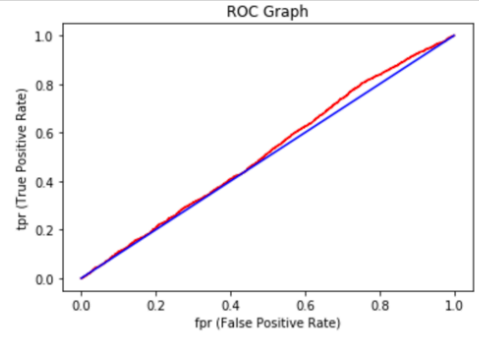
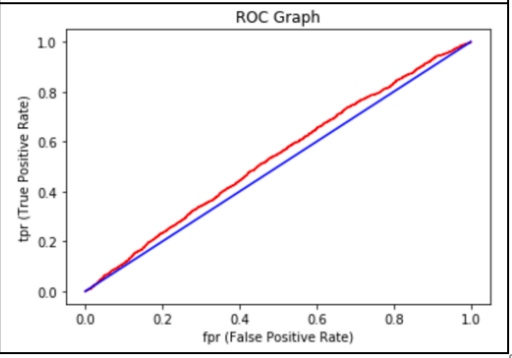


AUC score: 0.8575662964472861

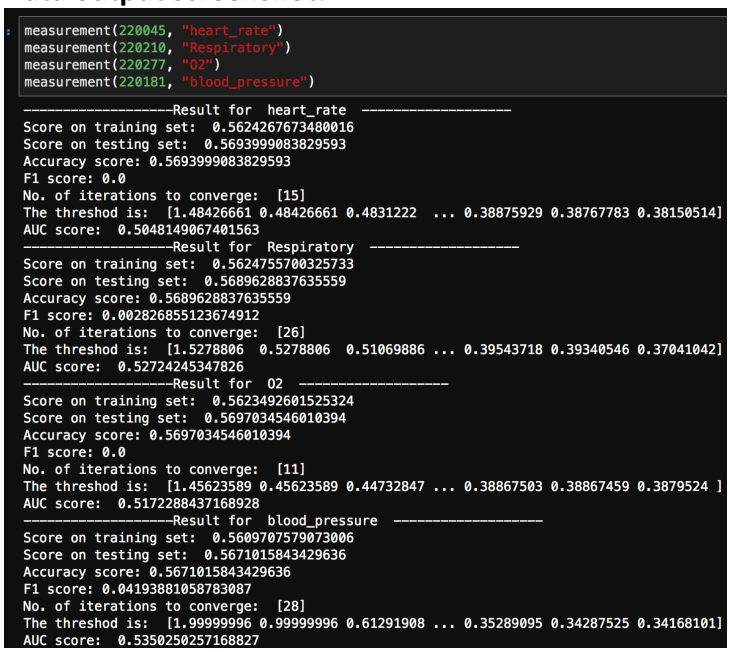
Comment on how the two sources of data affect the model:
The AUC score of ICU data is around 0.77, and the AUC score of notes data is around 0.84. After combining them, the AUC increases to 0.85, which means more sources will present a more accurate model.

Question 5

(a) As a baseline, train a logistic regression to predict hypertension, using the min, max, and mean heart rate/ respiratory rate/ O2 saturation/ blood pressure as features. Remove patients with fewer than two heart rate measurements. Report the AUC and F1 scores on the test set.

Measurement	AUC score	F1 score	ROC graph
Heart rate	0.5048149067401563	0.0	
Respiratory rate	0.52724245347826	0.002826855123674912	
O2 saturation	0.5172288437168928	0.0	
Blood pressure	0.5350250257168827	0.04193881058783087	

Data output screenshot:



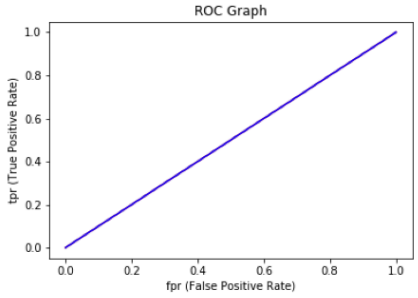
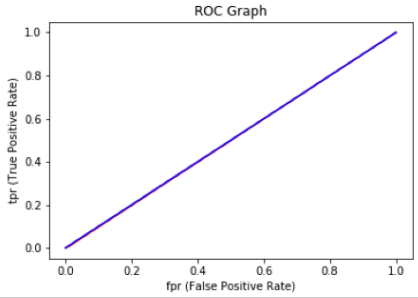
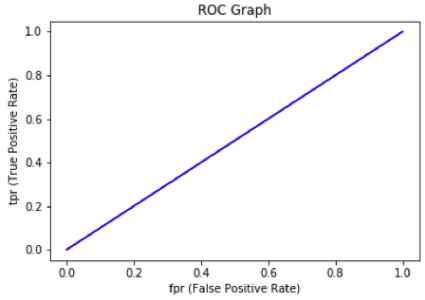
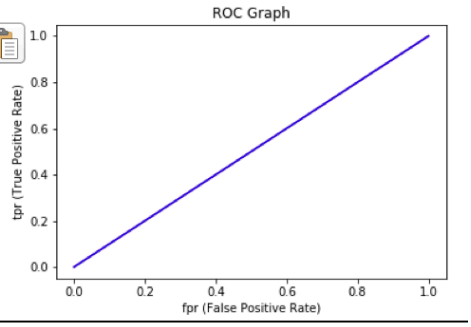
```
measurement(220045, "heart_rate")
measurement(220210, "Respiratory")
measurement(220277, "O2")
measurement(220181, "blood_pressure")

--------------------Result for  heart_rate  --------------------
Score on training set:  0.5624267673480016
Score on testing set:  0.5693999083829593
Accuracy score: 0.5693999083829593
F1 score: 0.0
No. of iterations to converge:  [15]
The threshod is:  [1.48426661 0.48426661 0.4831222  ... 0.38875929 0.38767783 0.38150514]
AUC score:  0.5048149067401563
--------------------Result for  Respiratory  --------------------
Score on training set:  0.5624755700325733
Score on testing set:  0.5689628837635559
Accuracy score: 0.5689628837635559
F1 score: 0.002826855123674912
No. of iterations to converge:  [26]
The threshod is:  [1.5278806  0.5278806  0.51069886 ... 0.39543718 0.39340546 0.37041042]
AUC score:  0.52724245347826
--------------------Result for  O2  --------------------
Score on training set:  0.5623492601525324
Score on testing set:  0.5697034546010394
Accuracy score: 0.5697034546010394
F1 score: 0.0
No. of iterations to converge:  [11]
The threshod is:  [1.45623589 0.45623589 0.44732847 ... 0.38867503 0.38867459 0.3879524 ]
AUC score:  0.5172288437168928
--------------------Result for  blood_pressure  --------------------
Score on training set:  0.5609707579073006
Score on testing set:  0.5671015843429636
Accuracy score: 0.5671015843429636
F1 score: 0.04193881058783087
No. of iterations to converge:  [28]
The threshod is:  [1.99999996 0.99999996 0.61291908 ... 0.35289095 0.34287525 0.34168101]
AUC score:  0.5350250257168827
```

As we can see the results above, the **accuracy score** ranges from 0.567~ 0.570.

(b) Train a LSTM to predict hypertension, using the min, max, and mean heart rate/ respiratory rate/ O2 saturation/ blood pressure as features. Remove patients with fewer than two heart rate measurements. Report the AUC and F1 scores on the test set.

For this question, in order to use the LSTM model, we need to keep timesteps the same for each batch. In this dataset, the maximum timestep is larger than 40,000. If I keep such a long timestep, the training process will be tremendously slow, and the model will be greatly influenced by t such outliers. Then I drew a histogram for all timesteps for each hadm_id, I found that the majority of the timesteps are around 100. In this case, I drop the records whose timestep is larger than 300. I think 300 is quite generous since timesteps larger than 300 are only a small proportion.

Measurement	AUC score	F1 score	Accuracy score	ROC graph
Heart rate	0.5	0.0	0.565174456879526	
Respiratory rate	0.4982573874766265	0.007473841554559044	0.5630860338871525	
O2 saturation	0.5	0.0	0.565761316872428	
Blood pressure	0.5	0.0	0.5685246942988724	

Data output screenshot:

```
measurement(220045, "heart_rate")

---------------------------- heart_rate  ----------------------------
Train on 14313 samples, validate on 6076 samples
Epoch 1/4
14313/14313 [==============================] - 109s 8ms/sample - loss: 0.6960 - acc: 0.5333 - val_loss: 0.6852 - val_acc: 0.5604
Epoch 2/4
14313/14313 [==============================] - 107s 7ms/sample - loss: 0.6870 - acc: 0.5587 - val_loss: 0.6847 - val_acc: 0.5652
Epoch 3/4
14313/14313 [==============================] - 108s 8ms/sample - loss: 0.6862 - acc: 0.5605 - val_loss: 0.6847 - val_acc: 0.5652
Epoch 4/4
14313/14313 [==============================] - 109s 8ms/sample - loss: 0.6859 - acc: 0.5607 - val_loss: 0.6855 - val_acc: 0.5652
------------Performance------------
AUC score:  0.5
F1 score:  0.0
Accuracy score:  0.565174456879526

measurement(220210, "Respiratory")

---------------------------- Respiratory  ----------------------------
Train on 14317 samples, validate on 6079 samples
Epoch 1/4
14317/14317 [==============================] - 108s 8ms/sample - loss: 0.6998 - acc: 0.5361 - val_loss: 0.6810 - val_acc: 0.5656
Epoch 2/4
14317/14317 [==============================] - 108s 8ms/sample - loss: 0.6902 - acc: 0.5465 - val_loss: 0.6835 - val_acc: 0.5558
Epoch 3/4
14317/14317 [==============================] - 109s 8ms/sample - loss: 0.6869 - acc: 0.5510 - val_loss: 0.6817 - val_acc: 0.5647
Epoch 4/4
14317/14317 [==============================] - 108s 8ms/sample - loss: 0.6859 - acc: 0.5495 - val_loss: 0.6822 - val_acc: 0.5631
------------Performance------------
AUC score:  0.4982573874766265
F1 score:  0.007473841554559044
Accuracy score:  0.5630860338871525

measurement(220277, "O2")

---------------------------- O2  ----------------------------
Train on 14323 samples, validate on 6075 samples
Epoch 1/4
14323/14323 [==============================] - 110s 8ms/sample - loss: 0.6975 - acc: 0.5415 - val_loss: 0.6844 - val_acc: 0.5658
Epoch 2/4
14323/14323 [==============================] - 108s 8ms/sample - loss: 0.6949 - acc: 0.5356 - val_loss: 0.6844 - val_acc: 0.5658
Epoch 3/4
14323/14323 [==============================] - 107s 7ms/sample - loss: 0.6911 - acc: 0.5452 - val_loss: 0.6848 - val_acc: 0.5658
Epoch 4/4
14323/14323 [==============================] - 109s 8ms/sample - loss: 0.6904 - acc: 0.5460 - val_loss: 0.6844 - val_acc: 0.5658
------------Performance------------
AUC score:  0.5
F1 score:  0.0
Accuracy score:  0.565761316872428

measurement(220181, "blood_pressure")

---------------------------- blood_pressure  ----------------------------
Train on 14797 samples, validate on 6297 samples
Epoch 1/4
14797/14797 [==============================] - 112s 8ms/sample - loss: 0.7167 - acc: 0.5180 - val_loss: 0.6890 - val_acc: 0.5698
Epoch 2/4
14797/14797 [==============================] - 112s 8ms/sample - loss: 0.6974 - acc: 0.5369 - val_loss: 0.6828 - val_acc: 0.5685
Epoch 3/4
14797/14797 [==============================] - 111s 8ms/sample - loss: 0.6917 - acc: 0.5451 - val_loss: 0.6832 - val_acc: 0.5685
Epoch 4/4
14797/14797 [==============================] - 110s 7ms/sample - loss: 0.6886 - acc: 0.5483 - val_loss: 0.6832 - val_acc: 0.5685
------------Performance------------
AUC score:  0.5
F1 score:  0.0
Accuracy score:  0.5685246942988724
```

(c) Comment on the performance of your models. Which measurements are the most useful for predicting hypertension? Do the results make sense?

Both of the models didn't work well or the results don't make sense, and shouldn't be chosen for prediction, though linear regression performs slightly better than LSTM.

My reasons are as follow:
I tried to print the result of `the sum of hypertension labels / total number of labels`, and then use 1 to minus this, and still got the result around 0.56 (Please see the screenshot below). This means even we do prediction using all 0s, we can still get accuracy 0.56.

```
1 - y_test_np.sum() / y_test_np.shape[0]
```

```
0.565174456879526
```

Linear regression: The aggregated values like min, max and mean do not provide full information for predicting hypertension.
LSTM: I asked this question to TA, and TA suggested to add more epoch, add LSTM layers, and increase the number of neurons. However, I tried all these suggestions but the model still didn't work well:

- Add more epochs: the results are nearly identically the same for the 2nd epoch and 3rd epoch. Also, there were more than 14, 000 samples for each training process, so the size of the dataset is relatively large. More epoch may result in overfitting.
- Add LSTM layers: it didn't help because we only have one feature for each training process.
- Increase the number of neurons: I tried to increase neurons from 16 to 128, and the result even worse. So I kept 16 neurons for the LSTM model.