1.1 Batch Gradient Descent

a. Derive the gradient of the negative log-likelihood in terms of w for this setting. [5 points]

$$NLL(D,w) = -\sum_{i=1}^{N}\left[(1-y_i)\log\left(1-\sigma(w^Tx_i)\right) + y_i log\sigma(w^Tx_i)\right]$$

$$\frac{\partial NNL(D,w)}{\partial w_j} =$$

$$-\sum_{i=1}^{N}(1-y_j)\frac{1}{1-\sigma(w^Tx_i)}\sigma(w^Tx_i)(\sigma(w^Tx_i)-1)x_i$$
$$+ y_j\frac{1}{\sigma(w^Tx_i)}\sigma(w^Tx_i)(\sigma(w^Tx_i)-1)\left(1-\sigma(w^Tx_i)\right)x_i$$
$$= -\sum_{i=1}^{N}\left[(y_i-1)\sigma(w^Tx_i)x_i + y_i\left(1-\sigma(w^Tx_i)\right)x_i\right]$$

$$= -\sum_{i=1}^{N}x_i\left(y_i-\sigma(w^Tx_i)\right)$$

1.2 Stochastic Gradient Descent

a. Show the positive log likelihood, l, of a single (xt,yt) pair. [5 points]

$$l(w) = (1-y_t)\log\left(1-\sigma(w^Tx_t)\right) + y_t\log\left(\sigma(w^Tx_t)\right)$$

b.

$$\frac{\partial l}{\partial w_j} = x_t\left(y_t - \sigma(w_j^Tx_t)\right)$$

$$w_t = w_{t-1} + \eta x_t\left(y_t - \sigma(w_{t-1}^Tx_t)\right)$$

c. Suppose m is the total number of features (regardless of whether they are non-zero), n is the total number of non-zero features for each sample and T is the number of iterations. What is the smallest time complexity (in big-O notation) of the update rule from b if xt by all iterations?

Because n is the total number of non-zero features, and we only need to update the non-zero features, which is n. The smallest time complexity O (nT)

What if dimension is very sparse, i.e., n is small constant, what can the complexity be (in big-O notation)?

The dimension is very sparse, n is a small constant.

O (nT) -> O(T)

d. Briefly explain the consequence of using a very large η and very small η. [3 points]

Large η: large η can lead to converging too quickly to a suboptimal solution or it can cause oscillations around the optimum, and in the worst-case scenario, it can lead to outright divergence, infinite iterations

Small η: small η can take too many iterations to converge to the optimum or it can get stuck in a local optimum.

e. Show how to update wt under the penalty of L2 norm regularization. In other words, update wt according to l − μ‖w‖2, where μ is a constant. The learning rate η should be applied to ∂ (l − μ‖w‖2). What's the time complexity (use the same notation from c)? [5 points]

$$\frac{\delta\left(l - \mu\|w\|_2^2\right)}{\delta w} = x_t\left(y_t - \sigma(w^T x_t)\right) - 2\mu w$$

$$w_t = w_{t-1} + \eta\left(x_t\left(y_t - \sigma(w^T x_t)\right) - 2\mu w\right)$$

Suppose we have n non-zero features, the time complexity is O(nT).

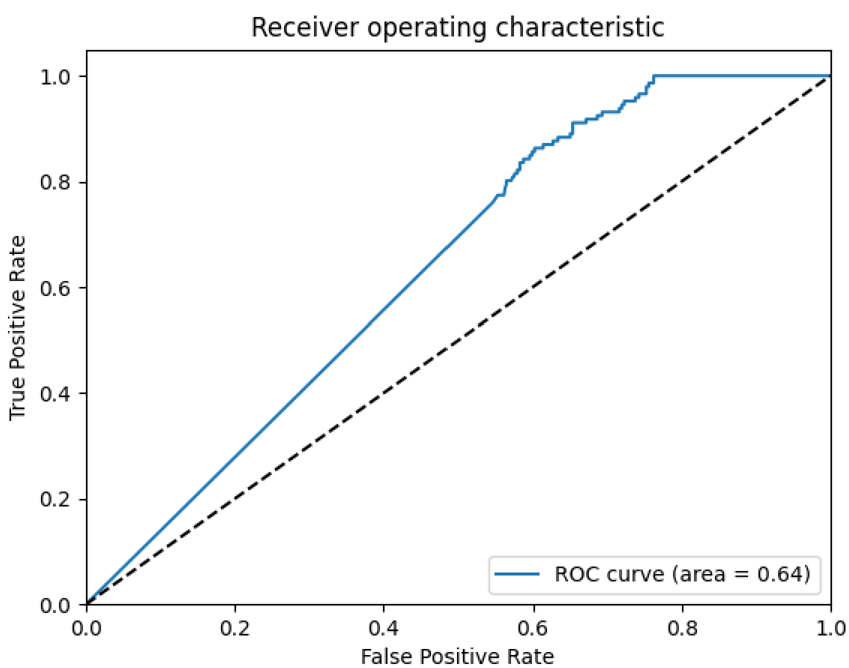
If n is a small constant, the time complexity is O(T)

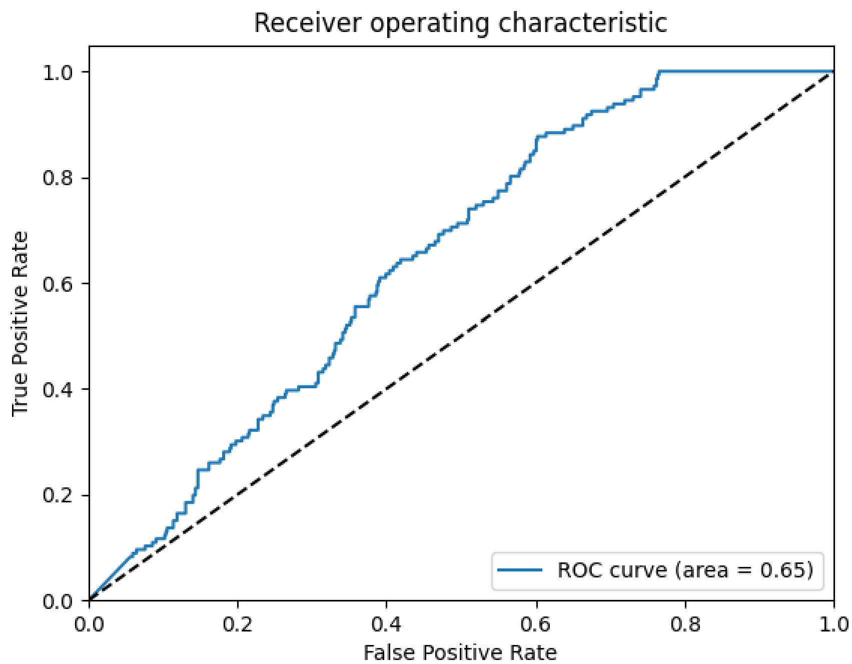
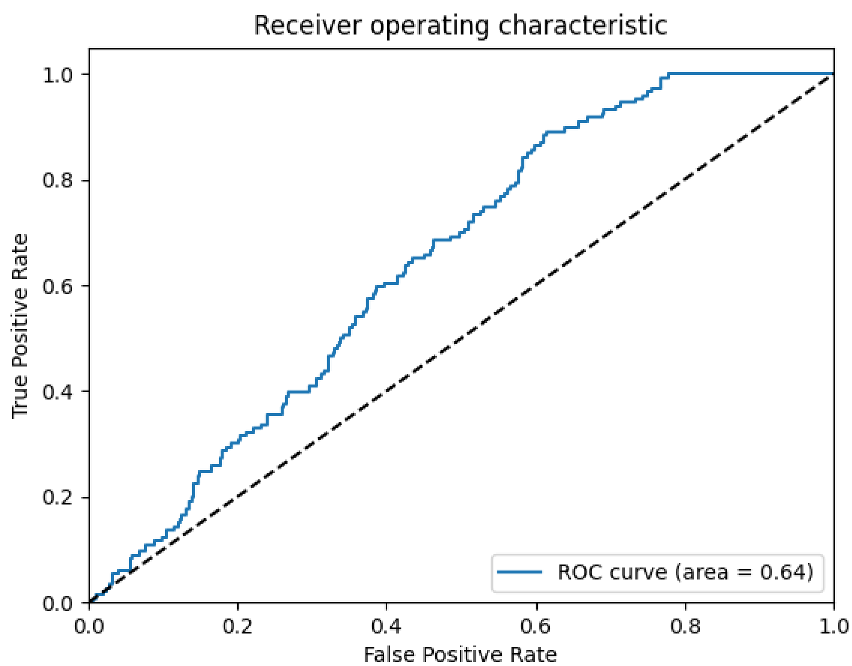
2.1 b. Use events.csv and mortality.csv provided in data as input and fill Table 2 with ac- tual values (you can keep two decimal places for float numbers when fill the form) [6 points]. We only need the top 5 codes for common diagnoses, labs and medications. Their respective counts are not required.

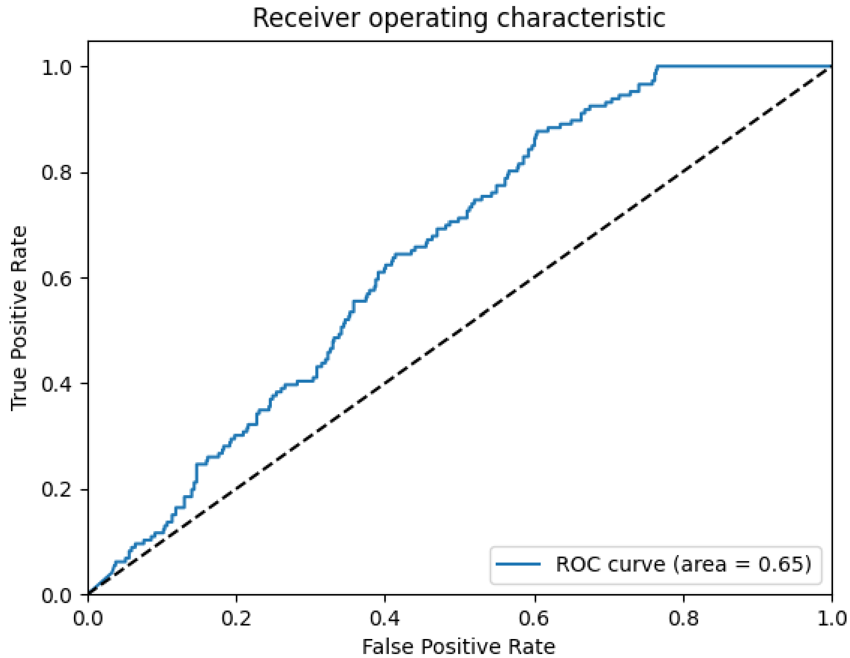
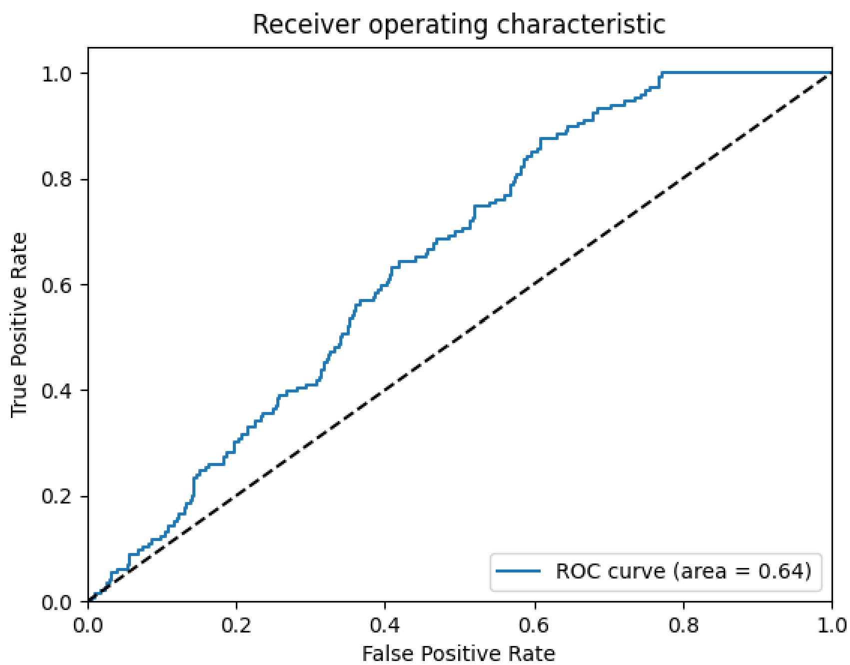
Metric	Deceased patients	Alive patients
Event Count		
1. Average Event Count	1027.74	683.16
2. Max Event Count	16829	12627
3. Min Event Count	2	1
Encounter Count		
1. Average Encounter Count	24.84	18.70
2. Median Encounter Count	14	9
3. Max Encounter Count	375	391
4. Min Encounter Count	1	1
Record Length		
1. Average Record Length	157.04	194.70
2. Median Record Length	25	16
3. Max Record Length	5364	3103
4. Min Record Length	0	0
Common Diagnosis	DIAG320128 DIAG319835 DIAG313217 DIAG197320	DIAG320128 DIAG319835 DIAG317576 DIAG42872402

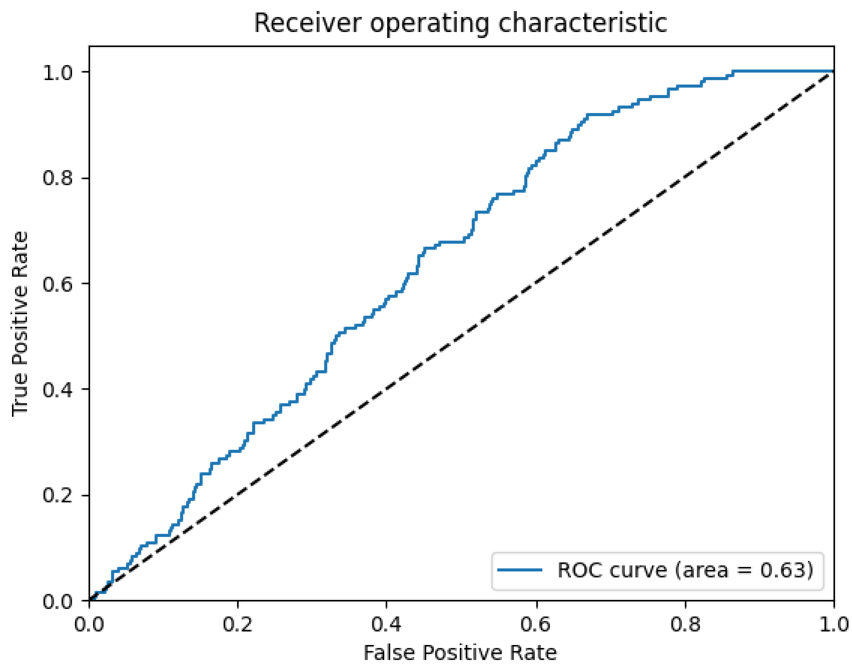
	DIAG132797	DIAG313217
Common Laboratory Test	LAB3009542 LAB3023103 LAB3000963 LAB3018572 LAB3016723	LAB3009542 LAB3000963 LAB3023103 LAB3018572 LAB3007461
Common Medication	DRUG19095164 DRUG43012825 DRUG19049105 DRUG956874 DRUG19122121	DRUG19095164 DRUG43012825 DRUG19049105 DRUG19122121 DRUG956874

2.3 b. Show the ROC curve generated by test.py in this writing report for different learning rates η and regularization parameters μ combination and briefly explain the result. [5 points]

<table>
<tr><th>η</th><th>μ</th><th></th></tr>
<tr><td>0.1</td><td>0.0</td><td>
</td></tr>
</table>

0.01	0.0	
0.001	0.0	

0.01	0.01	
0.01	0.1	

0.01	1	

When μ = 0.0 (penalty is 0), as η increases, there are less data points making the curve, the ROC curve becomes smoother.

Comparing η = 0.01, AUC = 0.65, and η = 0.001, AUC = 0.64. There is a possibility that with the smaller η = 0.01, it may get stuck in the suboptimum.

Comparing η = 0.01, AUC = 0.65, and η = 0.1, AUC = 0.64. The larger η = 0.1 causes the model to converge too quickly to a suboptimal solution

When η = 0.01 (learning rate is constant, 0.01), as μ (penalty) increases, (as we try to minimize the effect of overfitting) the AUC decreases. Because as the regularization increases, the weights decrease and the model shrinks, indicating the model becomes underfitting.