September 1, 2025 Joseph DeLeonardis
UIN: 0820000866

CSCE-633

Machine Learning Homework #1

Problem 1: Gradient Calculation (8 points)

In this question you are required to calculate gradients for 2 scalar functions.

(a) Calculate the gradient of the function $f(x, y) = x^2 + \ln(y) + xy + y^3$. What is the gradient value for $(x, y) = (10, -10)$?

The gradient is: $\nabla f = \left(\frac{\partial f}{\partial x}, \frac{\partial f}{\partial y}\right)$

Taking partial derivatives: $\frac{\partial f}{\partial x} = 2x + y$

$\frac{\partial f}{\partial y} = \frac{1}{y} + x + 3y^2$

$$\nabla f = \left\langle 2x + y, \frac{1}{y} + x + 3y^2 \right\rangle$$

At $(x, y) = (10, -10)$:

$$\nabla f = \left\langle 2(10) + (-10), \frac{1}{-10} + 10 + 3(-10)^2 \right\rangle = \langle 10, 309.9 \rangle$$

(b) Calculate the gradient of the function $f(x, y, z) = \tanh(x^3y^3) + \sin(z^2)$. What is the gradient value for $(x, y, z) = (-1, 0, \pi/2)$?

The gradient is:

$$\nabla f = \left(\frac{\partial f}{\partial x}, \frac{\partial f}{\partial y}, \frac{\partial f}{\partial z}\right)$$

Taking partial derivatives:

$$\frac{\partial f}{\partial x} = 3x^2y^3 sech^2(x^3y^3)$$

$$\frac{\partial f}{\partial y} = 3x^3y^2 sech^2(x^3y^3)$$

$$\frac{\partial f}{\partial z} = 2z\cos(z^2)$$

$$\nabla f = \langle 3x^2y^3 sech^2(x^3y^3), 3x^3y^2 sech^2(x^3y^3), 2z\cos(z^2) \rangle$$

At $(x, y, z) = (-1, 0, \pi/2)$:

Note: With y being 0, we can just say that the X and Y components of the gradient will go to 0

Simplifying the Expression $2z\cos(z^2)$ at $z = \frac{\pi}{2}$

We start with the original expression:

$$2z\cos(z^2)$$

Step 1: Substitute $z = \frac{\pi}{2}$

$$2 \cdot \frac{\pi}{2} \cdot \cos\left(\left(\frac{\pi}{2}\right)^2\right)$$

Step 2: Simplify the Coefficient

$$\pi \cdot \cos\left(\left(\frac{\pi}{2}\right)^2\right)$$

Step 3: Simplify the Exponent

$$\left(\frac{\pi}{2}\right)^2 = \frac{\pi^2}{4}$$

So the expression becomes:

$$\pi \cdot \cos\left(\frac{\pi^2}{4}\right)$$

$$\nabla f = \langle 0, 0, \pi \cos\left(\frac{\pi^2}{4}\right) \rangle$$

Problem 2: Matrix Multiplication (8 points)

(a)

Outer Product of Two Vectors

Let the vertical (column) vector be:

$$\mathbf{u} = \begin{bmatrix} 10 \\ -5 \\ 2 \\ 8 \end{bmatrix}$$

And the horizontal (row) vector be:

$$\mathbf{v} = \begin{bmatrix} 0 & 3 & 0 & 1 \end{bmatrix}$$

The outer product **uv** is:

$$\mathbf{uv} = \begin{bmatrix} 10 \\ -5 \\ 2 \\ 8 \end{bmatrix} \begin{bmatrix} 0 & 3 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 10 \cdot 0 & 10 \cdot 3 & 10 \cdot 0 & 10 \cdot 1 \\ -5 \cdot 0 & -5 \cdot 3 & -5 \cdot 0 & -5 \cdot 1 \\ 2 \cdot 0 & 2 \cdot 3 & 2 \cdot 0 & 2 \cdot 1 \\ 8 \cdot 0 & 8 \cdot 3 & 8 \cdot 0 & 8 \cdot 1 \end{bmatrix} = \begin{bmatrix} 0 & 30 & 0 & 10 \\ 0 & -15 & 0 & -5 \\ 0 & 6 & 0 & 2 \\ 0 & 24 & 0 & 8 \end{bmatrix}$$

(b)

Matrix Multiplication

We are given the following two matrices:

$$A = \begin{bmatrix} 1 & -1 & 6 & 7 \\ 9 & 0 & 8 & 1 \\ -8 & 1 & 2 & 3 \\ -10 & 4 & 0 & 1 \end{bmatrix} \quad \text{and} \quad B = \begin{bmatrix} 6 & 2 & 0 \\ 0 & -1 & 1 \\ -3 & 0 & 4 \\ 3 & 4 & 7 \end{bmatrix}$$

The multiplication of two matrices A and B is performed by taking the dot product of each row of matrix A with each column of matrix B. We will calculate the resulting matrix $C = A \times B$.

$$C = A \times B$$

$$C = \begin{bmatrix} (1\cdot 6 + (-1)\cdot 0 + 6\cdot(-3) + 7\cdot 3) & (1\cdot 2 + (-1)\cdot(-1) + 6\cdot 0 + 7\cdot 4) & (1\cdot 0 + (-1)\cdot 1 + 6\cdot 4 + 7\cdot 7) \\ (9\cdot 6 + 0\cdot 0 + 8\cdot(-3) + 1\cdot 3) & (9\cdot 2 + 0\cdot(-1) + 8\cdot 0 + 1\cdot 4) & (9\cdot 0 + 0\cdot 1 + 8\cdot 4 + 1\cdot 7) \\ (-8\cdot 6 + 1\cdot 0 + 2\cdot(-3) + 3\cdot 3) & (-8\cdot 2 + 1\cdot(-1) + 2\cdot 0 + 3\cdot 4) & (-8\cdot 0 + 1\cdot 1 + 2\cdot 4 + 3\cdot 7) \\ (-10\cdot 6 + 4\cdot 0 + 0\cdot(-3) + 1\cdot 3) & (-10\cdot 2 + 4\cdot(-1) + 0\cdot 0 + 1\cdot 4) & (-10\cdot 0 + 4\cdot 1 + 0\cdot 4 + 1\cdot 7) \end{bmatrix}$$

Now, we will calculate each element in the resulting matrix.

$$C = \begin{bmatrix} (6 + 0 - 18 + 21) & (2 + 1 + 0 + 28) & (0 - 1 + 24 + 49) \\ (54 + 0 - 24 + 3) & (18 + 0 + 0 + 4) & (0 + 0 + 32 + 7) \\ (-48 + 0 - 6 + 9) & (-16 - 1 + 0 + 12) & (0 + 1 + 8 + 21) \\ (-60 + 0 + 0 + 3) & (-20 - 4 + 0 + 4) & (0 + 4 + 0 + 7) \end{bmatrix}$$

Simplifying the entries:

$$C = \begin{bmatrix} 9 & 31 & 72 \\ 33 & 22 & 39 \\ -45 & -5 & 30 \\ -57 & -20 & 11 \end{bmatrix}$$

Thus, the result of multiplying $A \times B$ is:

$$C = \begin{bmatrix} 9 & 31 & 72 \\ 33 & 22 & 39 \\ -45 & -5 & 30 \\ -57 & -20 & 11 \end{bmatrix}$$

Programming Section

Data Processing (4 points)

To ensure the datasets were properly loaded, I printed out the shapes of both the training and testing sets and obtained the following results:

Dataset	Samples	Features
Training	6,250	12 (11 features + 1 target)
Test	3,221	11 (features only)

The testing set has one less column in the shape because it does not contain the target variable which is typical for this type of problem.

Missing Values Analysis: The dataset contained **924** missing values, which were handled using the `dropna()` method to remove incomplete records in the clean the data.

Feature Extraction: Successfully separated the feature matrix **X** (all columns except the last) from the target variable **y** (PT08.S1(CO) sensor readings). Having clean data sets to work with was essential for this task.

Exploratory Data Analysis (10 points)

1. Feature Distribution Analysis
Generated histograms for all 11 features to examine their distributions and assess normality.

Distribution Assessment:

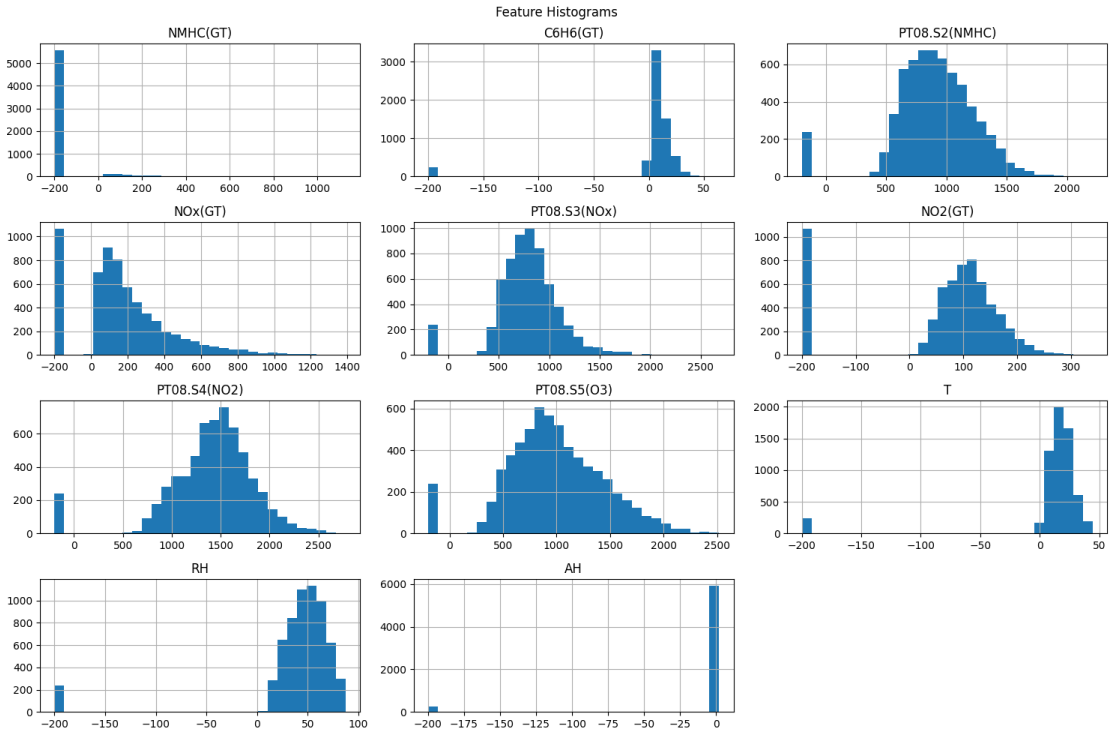

Figure 1: Distribution histograms for all 11 features in the dataset

- **Normally Distributed Features:** PT08.S2(NMHC), PT08.S4(NO2), PT08.S5(O3), NO2(GT)
- **Skewed Features:** NMHC(GT) (right-skewed), C6H6(GT) (right-skewed), NOx(GT) (right-skewed), PT08.S3(NOx) (right-skewed), T (left-skewed), RH (left-skewed)
- **Multimodal Features:** AH (Absolute Humidity) shows a sharp, spike-like distribution with most values concentrated around a single point

Outlier Analysis:

- **Features with Extreme Values:** NMHC(GT), C6H6(GT), NOx(GT), T (Temperature)
- **Outlier Impact:** Extreme values can skew the mean and standard deviation which can affect fitting the model and create undesirable results. This is because those features will dominate the algorithm and make the other features insignificant when processing the data. As result in most cases such as the specific problem analyzed this requires the features to be normalized between 0 and 1.

2. Feature Correlation Analysis

Selected features for correlation analysis: **Absolute Humidity (AH)** and **Relative Humidity (RH)**

.

AH and RH were selected in order to test for collinearity because it is expected that they will be highly correlated.

Scatter Plot Results:

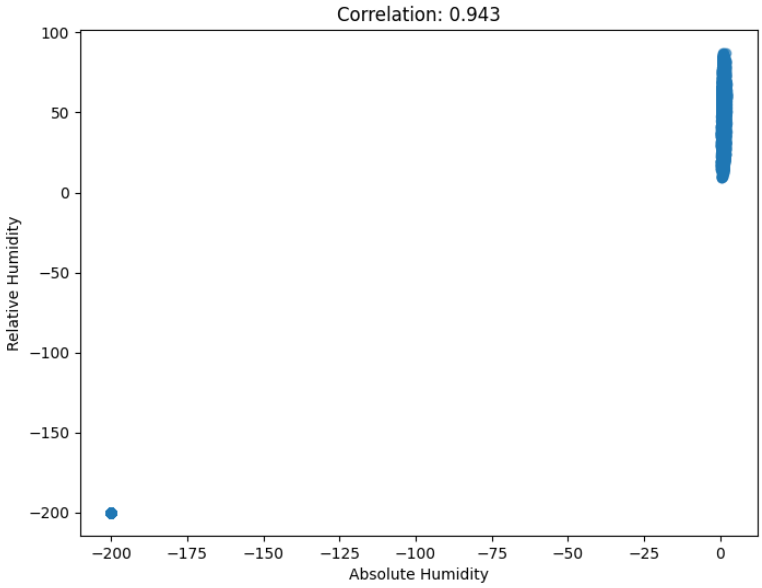

Figure 2: Scatter plot illustrating correlation between selected features

- **Pearson Correlation Coefficient: Correlation between AH and RH: 0.9433.**
- **Correlation Strength: Strong Positive**
- **Linear Relationship: Yes there is a linear relationship the line in the corner of the plot is a strong indicator of that.**
- **Data Point Distribution: There is at least one data point that is an outlier that could be removed with further analysis which is -200,-200.**

3. Pearson Correlation Matrix Analysis

Computed 12×12 correlation matrix C for all variables (11 features + 1 target variable PT08.S1(CO)).

	NMHC(GT)	C6H6(GT)	PT08.S2(NMHC)	NOx(GT)	PT08.S3(NOx)	NO2(GT)	PT08.S4(NO2)	PT08.S5(O3)	T	RH	AH	PT08.S1(CO)
NMHC(GT)	1.0000	0.0335	0.1101	-0.0056	0.0427	0.1001	0.1603	0.1017	-0.0061	0.0054	0.0075	0.1691
C6H6(GT)	0.0335	1.0000	0.7676	-0.0049	0.5075	-0.0160	0.7766	0.6411	0.9707	0.9242	0.9842	0.8510
PT08.S2(NMHC)	0.1101	0.7676	1.0000	0.3367	-0.0793	0.1835	0.8752	0.9102	0.6664	0.5853	0.6454	0.9325
NOx(GT)	-0.0056	-0.0049	0.3367	1.0000	-0.4496	0.8167	0.0385	0.4684	-0.1458	-0.0590	-0.1030	0.2826
PT08.S3(NOx)	0.0427	0.5075	-0.0793	-0.4496	1.0000	-0.2701	0.1184	-0.2165	0.5866	0.5678	0.6188	0.0773
NO2(GT)	0.1001	-0.0160	0.1835	0.8167	-0.2701	1.0000	-0.0172	0.2621	-0.0923	-0.0907	-0.0694	0.1584
PT08.S4(NO2)	0.1603	0.7766	0.8752	0.0385	0.1184	-0.0172	1.0000	0.7248	0.7547	0.6415	0.6932	0.8448
PT08.S5(O3)	0.1017	0.6411	0.9102	0.4684	-0.2165	0.2621	0.7248	1.0000	0.5011	0.5238	0.5178	0.8938
T	-0.0061	0.9707	0.6664	-0.1458	0.5866	-0.0923	0.7547	0.5011	1.0000	0.8842	0.9809	0.7504
RH	0.0054	0.9242	0.5853	-0.0590	0.5678	-0.0907	0.6415	0.5238	0.8842	1.0000	0.9433	0.7441
AH	0.0075	0.9842	0.6454	-0.1030	0.6188	-0.0694	0.6932	0.5178	0.9809	0.9433	1.0000	0.7620
PT08.S1(CO)	0.1691	0.8510	0.9325	0.2826	0.0773	0.1584	0.8448	0.8938	0.7504	0.7441	0.7620	1.0000

Table 1: Correlation matrix of sensor and environmental data.

Correlation Matrix Findings:

- **Strongest Positive Correlation: PT08.S2(NMHC) and PT08.S1(CO), correlation = 0.9325**
- **Strongest Negative Correlation: NOx(GT) and PT08.S3(NOx): -0.4496, NOx(GT) and PT08.S3(NOx), correlation = -0.4496**

- **Target Variable Correlations: PT08.S2(NMHC) = 0.9325, PT08.S5(O3) = 0.8938, C6H6(GT) = 0.8510**

Variable Associations:

- **High Multicollinearity: PT08.S2(NMHC) & PT08.S1(CO) = 0.9325, PT08.S2(NMHC) & PT08.S5(O3) = 0.9102, C6H6(GT) & AH = 0.9842, C6H6(GT) & T = 0.9707, PT08.S1(CO) & PT08.S5(O3) = 0.8938, PT08.S4(NO2) & PT08.S2(NMHC) = 0.8752.**
- **Independent Features: NMHC(GT) = 0.17, NO2(GT) = 0.16, PT08.S3(NOx) = 0.12**
- **Feature Clusters: Group 1: PT08.S2(NMHC), PT08.S1(CO), PT08.S5(O3), PT08.S4(NO2) Group 2: T, RH, AH Group 3: NOx(GT), NO2(GT), PT08.S3(NOx)**

Implications for Modeling: Features with high collinearity can create redundancy this makes it tough to pin point effects of individual features and can skew model results.

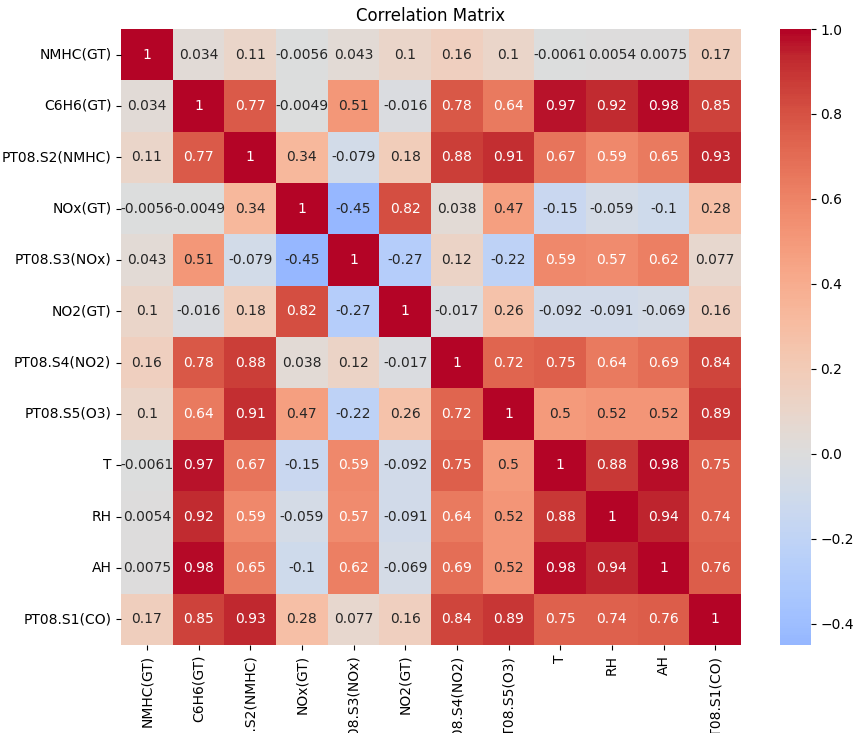

Figure 3: Pearson correlation matrix heatmap for all 12 variables

Linear Regression (20 points)

The MSE drops rapidly from approximately 1.2×10^6 to near zero within the first few hundred iterations, indicating that the model quickly finds the optimal solution. However, given the constraints of achieving an RMSE under 71 for the assignment, many different hyperparameter values were experimented with. This showed the model has stable convergence over many iterations. The smooth pattern is characteristic of effectively normalizing the features and applying the linear regression method.

Training Loss Plot: Model Implementation:

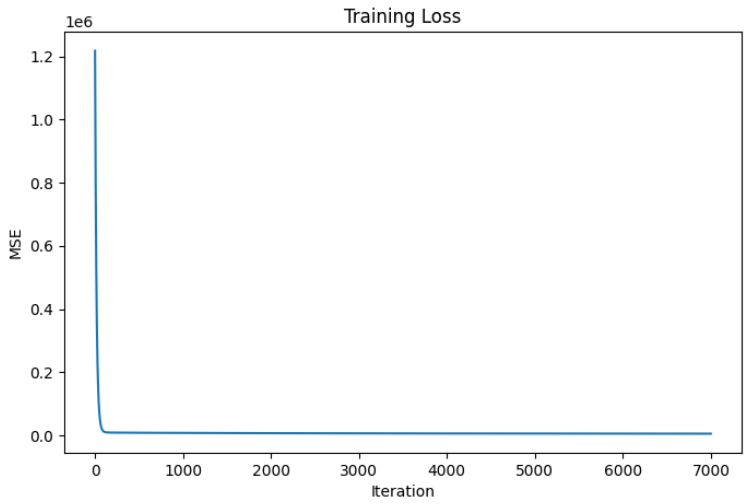

Figure 4: Training loss curve showing MSE convergence over iterations

Two approaches for linear regression were implemented:
1. Closed-Form Solution: Used the normal equation:

$$\mathbf{w} = (\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\mathbf{y}$$

Results:

- RMSE: **71.0793**

2. Gradient Descent: Implemented iterative optimization with the following hyperparameters:

- Learning rate: **0.1**
- Max iterations: **100000**
- Feature normalization: **Applied**

Results:

- Final MSE: **5052.27**
- Final RMSE: **71.0793560590635**

Gradient Descent (with cross validation): K-fold proved that the below 71 metric can be met in subsets of the data, more detailed explanation in section 5.

- Final RMSE: **K-fold 2, 70.40735343709633**
- Final RMSE: **K-fold 5, 70.60562912579253**

Performance: This part of the assignment involved a bit of trial and error. My first approach was gradient descent without normalizing any of the features. Upon tweaking the hyperparameters, this approach felt difficult, and it was noted that higher-valued features can drive our datasets. That was something I did not want, so I normalized all the features using (X - mean) / std. This still appeared to be exceedingly difficult to get an RMSE under 71. A separate .fit() function for a closed-form solution was written to get some insight on whether the algorithm was reaching the floor or if the hyperparameters could have been further optimized. In both cases, I got almost the same exact answer for RMSE—extremely close to the desired output but not quite there. Regularization was also attempted to see what the effects on the output data would be since it was covered in Module 3, but it did not show improvements to the data and was removed.

(4) Logistic Regression Implementation (20 points)

1. Binary Label Creation: Using the column PT08.S1(CO), binary labels were created where values greater than 1000 correspond to label 1 and values less than or equal to 1000 correspond to label 0. Results:

- Label 0 (less than or equal to 1000): 3090 samples 50.1%)
- Label 1 (greater than 1000): 3083 samples 49.9%)

2. Loss Function (Criterion): The model uses Binary Cross Entropy (BCE) loss as the criterion function. The Final BCE Loss was recorded at 0.2203.

3. Training Loop and Loss Plot: Gradient descent optimization was used because in the lectures we learned that in logistic regression we can not use a closed-form solution.

Training Parameters:

- Learning Rate: [0.1]
- Maximum Iterations: [5000]
- L2 Regularization: [0.001]
- Final BCE Loss: [0.2203]

4. Model Predictions: The trained model makes predictions using the sigmoid activation function to output probabilities, which are then converted to binary classifications using a 0.5 threshold.

5. Hyperparameter Tuning Results: Hyperparameters were tuned to achieve the target performance metrics on the validation set. Initially the model was failing and it was found that I accidentally used binary values instead of probability which led to a AUROC of around 0.49. This is an important lesson learned it is easy to mix those values up. Final Performance:

- F1 Score: 0.91 (Target: greater than or equal to 0.90)
- AUROC: 0.9707 (Target: greater than or equal to 0.90)

Both metrics exceed the required thresholds, demonstrating successful model performance.

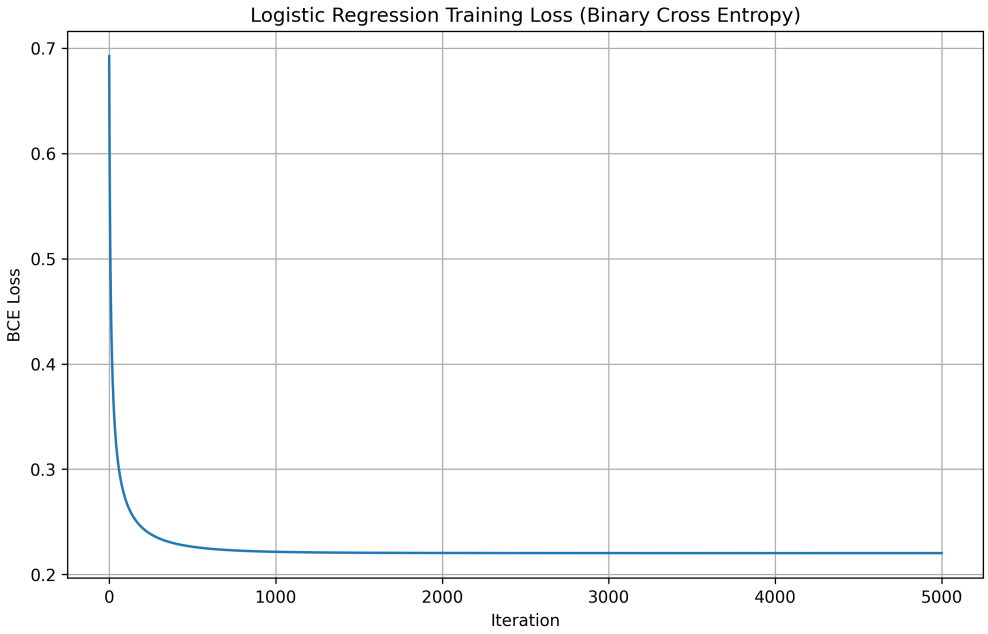

Figure 5: Binary Cross Entropy Loss during Logistic Regression Training

(5) Result Analysis - Cross Validation (20 points)

Linear Regression K-fold results K-fold values:

- RMSE Value: 71.07935605906191, 71.7614455560681, 70.40735343709633, 71.35758700816251, 71.18207286148925, 70.60562912579253
- Average RMSE: 71.18556069703826
- Standard deviation RMSE: 1.9769521833921653

It is important to note that it was very challenging to get an RMSE value within spec until doing a K-fold approach. Folds 2 and 5 fall with the RMSE specification required in the Linear Regression section of less than 71. I was surprised to see this result. I expected the closed form solution to be the best solution prior to collecting the data. Feature reduction might have accomplished this as well. We could in theory eliminate redundancy by using feature reduction. Right now I am uncertain how we choose which feature to keep and which ones to remove when comparing a group of features with high collinearity but this could be an approach investigated for next time in order to reach the optimal parameters. **Logistic Regression K-fold results**

- AUROC scores per fold: 0.9742881508761638, 0.9739920518813723, 0.9685913185913188, 0.9713204840780371, 0.9651559073999556
- F1 scores per fold: 0.9166666666666667, 0.0, 0.908485856905158, 0.9095435684647303, 0.9105824446267433
- 0.971 ± 0.003
- Average AUROC: 0.971 ± 0.003
- Average F1 Score(without the edge case): 0.911 ± 0.003 Average F1 Score (with the edge case): 0.729 ± 0.365

There was an edge case in there to determine if TP is 0. If we divided by 0 that would return undefined and break the program. Instead lets just say the entire thing equals 0. It not ideal to get a K fold with no TPs but it is theoretically possible. I think it is important to denote the average F1 score with and with it.

(6) Logistic Regression (10 points)

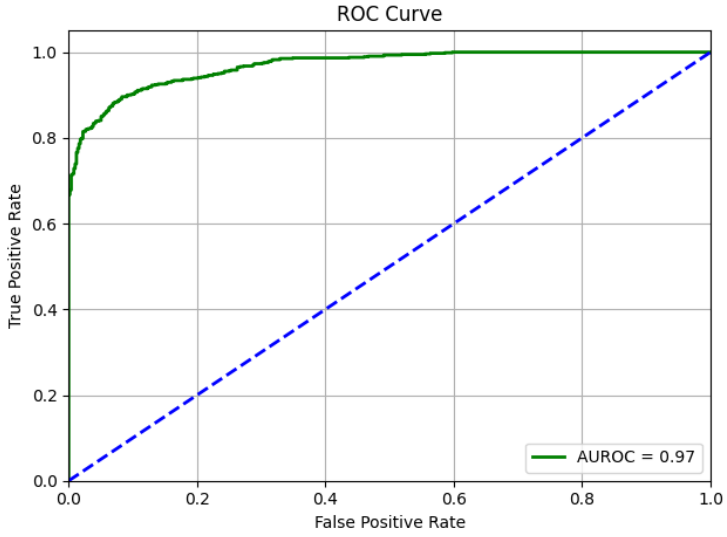

Figure 6: ROC Curve – K-Fold 1

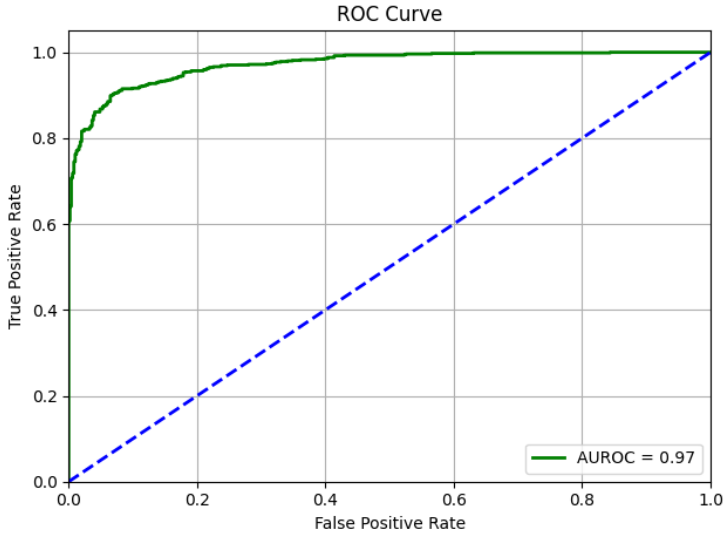

Figure 7: ROC Curve – K-Fold 2

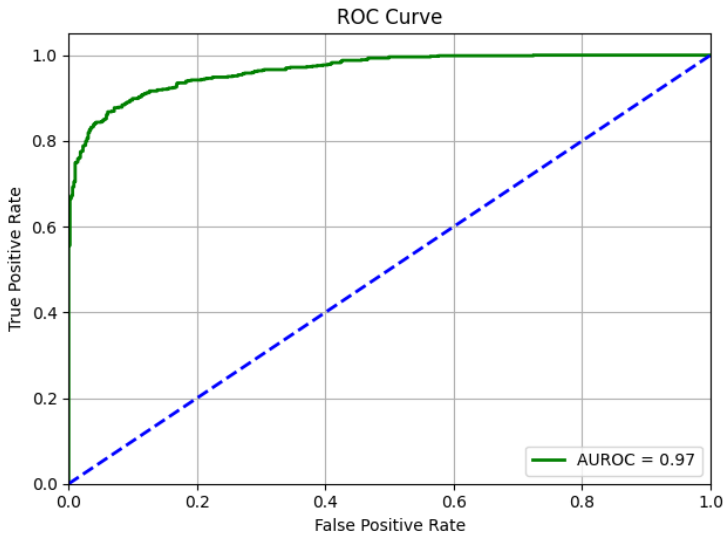

Figure 8: ROC Curve – K-Fold 3

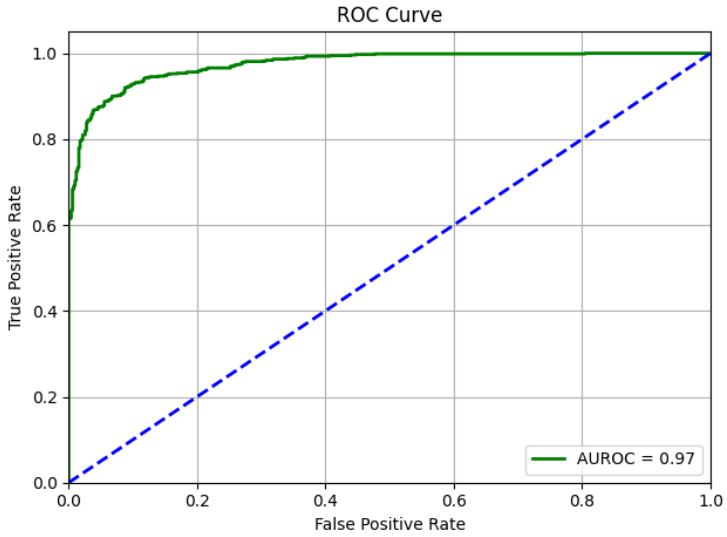

Figure 9: ROC Curve – K-Fold 4

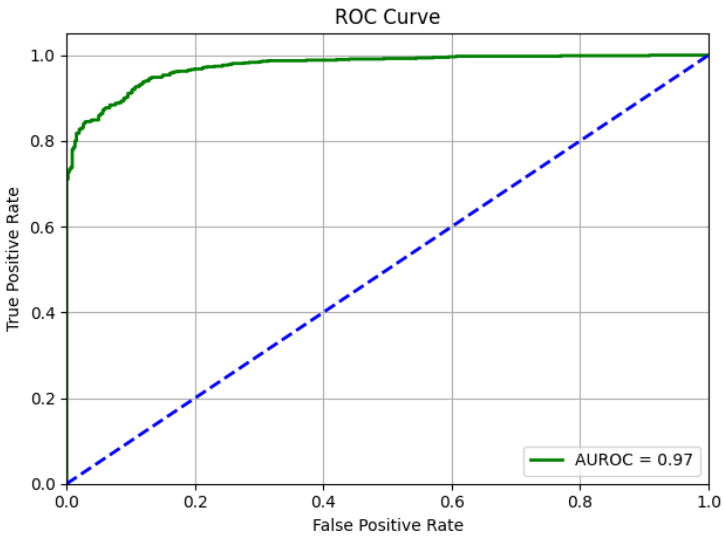

Figure 10: ROC Curve – K-Fold 5

All 5 of the k-folds look the exact same, this was expected. There's minor deviations in the dataset but not enough to visually tell when we graph an entire fold. The standard deviation is extremely small. These were the results I expected.