Homework 1: Linear Models

(Due Date: Jun 10, 2025)

MATH QUESTIONS

Problem 1: Gradient Calculation

(1)

$$f(x, y) = x^2 + \ln(y) + xy + y^3$$

$$\nabla f(x, y) = \left[\frac{\partial f}{\partial x},\ \frac{\partial f}{\partial y}\right] = \left[2x + y,\ \frac{1}{y} + x + 3y^2\right]$$

Since the domain of natural log function is all positive reals, $\ln(y)|_{y=-10}$ is not defined. Therefore, $f(10, -10)$ is not defined. Thus, $\nabla f(10, -10)$ is undefined.

For any $y > 0$,

$$\nabla f(10, y) = \left[20 + y,\ \frac{1}{y} + 10 + 3y^2\right]$$

If the problem is done mechanically (i.e., without considering the fact that $f(x, y)$ is not defined at $(10, -10)$, then

$$\nabla f(10, -10) = \left[20 - 10,\ \frac{-1}{10} + 10 + 3(100)\right] = [10, 309.9]$$

(2)

$$f(x, y, z) = tanh(x^3y^3) + sin(z^2)$$

$$\nabla f(x, y, z) = \left[\frac{\partial f}{\partial x},\ \frac{\partial f}{\partial y},\ \frac{\partial f}{\partial z}\right] = \left[3x^2y^3sech^2(x^3y^3),\ 3x^3y^2sech^2(x^3y^3),\ 2zcos(z^2)\right]$$

$$\begin{aligned}
\nabla f\left(-1, 0, \frac{\pi}{2}\right) &= \left[\frac{\partial f}{\partial x},\ \frac{\partial f}{\partial y},\ \frac{\partial f}{\partial z}\right]_{(-1,0,\frac{\pi}{2})} \\
&= \left[3(-1)^2(0)^3\,\mathrm{sech}^2\left((-1)^3(0)^3\right),\ 3(-1)^3(0)^2\,\mathrm{sech}^2\left((-1)^3(0)^3\right),\ 2\left(\frac{\pi}{2}\right)\cos\left(\left(\frac{\pi}{2}\right)^2\right)\right] \\
&= \left[0,\ 0,\ \pi\cos\left(\frac{\pi^2}{4}\right)\right] \\
&\approx [0,\ 0,\ -2.454]
\end{aligned}$$

Problem 2: Matrix Multiplication

(1) Since the first vector has dimension 4×1 and the second has dimension 1×4, the product will have dimension 4×4.

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

(2) Given two matrices below, say $A_{4\times3}$ and $B_{4\times3}$, we need to multiply A and B. Let that product be $C_{4\times3}$.

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

We can express matrix A as a *column vector of row vectors*:

$$A = \begin{bmatrix} \mathbf{a}_1^T \\ \mathbf{a}_2^T \\ \mathbf{a}_3^T \\ \mathbf{a}_4^T \end{bmatrix} \quad \text{where} \quad \begin{aligned} \mathbf{a}_1^T &= [1 \quad -1 \quad 6 \quad 7] \\ \mathbf{a}_2^T &= [9 \quad 0 \quad 8 \quad 1] \\ \mathbf{a}_3^T &= [-8 \quad 1 \quad 2 \quad 3] \\ \mathbf{a}_4^T &= [10 \quad 4 \quad 0 \quad 1] \end{aligned}$$

We can express matrix B as a *row vector of column vectors*:

$$B = \begin{bmatrix} \mathbf{b}_1 & \mathbf{b}_2 & \mathbf{b}_3 \end{bmatrix} \quad \text{where} \quad \mathbf{b}_1 = \begin{bmatrix} 6 \\ 0 \\ -3 \\ 3 \end{bmatrix}, \quad \mathbf{b}_2 = \begin{bmatrix} 2 \\ -1 \\ 0 \\ 4 \end{bmatrix}, \quad \mathbf{b}_3 = \begin{bmatrix} 0 \\ 1 \\ 4 \\ 7 \end{bmatrix}$$

Then, the matrix product AB can be expressed as:

$$AB = \begin{bmatrix} \mathbf{a}_1^T\mathbf{b}_1 & \mathbf{a}_1^T\mathbf{b}_2 & \mathbf{a}_1^T\mathbf{b}_3 \\ \mathbf{a}_2^T\mathbf{b}_1 & \mathbf{a}_2^T\mathbf{b}_2 & \mathbf{a}_2^T\mathbf{b}_3 \\ \mathbf{a}_3^T\mathbf{b}_1 & \mathbf{a}_3^T\mathbf{b}_2 & \mathbf{a}_3^T\mathbf{b}_3 \\ \mathbf{a}_4^T\mathbf{b}_1 & \mathbf{a}_4^T\mathbf{b}_2 & \mathbf{a}_4^T\mathbf{b}_3 \end{bmatrix}$$

$$C = AB = \begin{bmatrix} 1 & -1 & 6 & 7 \\ 9 & 0 & 8 & 1 \\ -8 & 1 & 2 & 3 \\ 10 & 4 & 0 & 1 \end{bmatrix} \begin{bmatrix} 6 & 2 & 0 \\ 0 & -1 & 1 \\ -3 & 0 & 4 \\ 3 & 4 & 7 \end{bmatrix} = \begin{bmatrix} 9 & 31 & 72 \\ 33 & 22 & 39 \\ -45 & -5 & 30 \\ 63 & 20 & 11 \end{bmatrix}$$

Where,

$$\begin{aligned} \mathbf{a}_1^T\mathbf{b}_1 &= 1 \cdot 6 + (-1) \cdot 0 + 6 \cdot (-3) + 7 \cdot 3 \\ &= 6 + 0 - 18 + 21 = 9 \end{aligned}$$

$$\begin{aligned} \mathbf{a}_1^T\mathbf{b}_2 &= 1 \cdot 2 + (-1) \cdot (-1) + 6 \cdot 0 + 7 \cdot 4 \\ &= 2 + 1 + 0 + 28 = 31 \end{aligned}$$

$$\begin{aligned} \mathbf{a}_1^T\mathbf{b}_3 &= 1 \cdot 0 + (-1) \cdot 1 + 6 \cdot 4 + 7 \cdot 7 \\ &= 0 - 1 + 24 + 49 = 72 \end{aligned}$$

and so on.

The Programming Questions start on the next page.

PROGRAMMING QUESTIONS

(1) Data Processing

(1) Data are loaded. The training data has shape (6250, 12), and the test data has shape (3221, 11).

(2) The training data has 77 rows with missing values. The total number of missing values in the training data is 924. The testing data has 37 rows with missing values. The total number of missing values in the testing data is 407.

(3) After dropping the rows with missing values, the shape of the training data is now (6250, 12). After dropping the rows with missing values, the testing data shape is now (3184, 11).

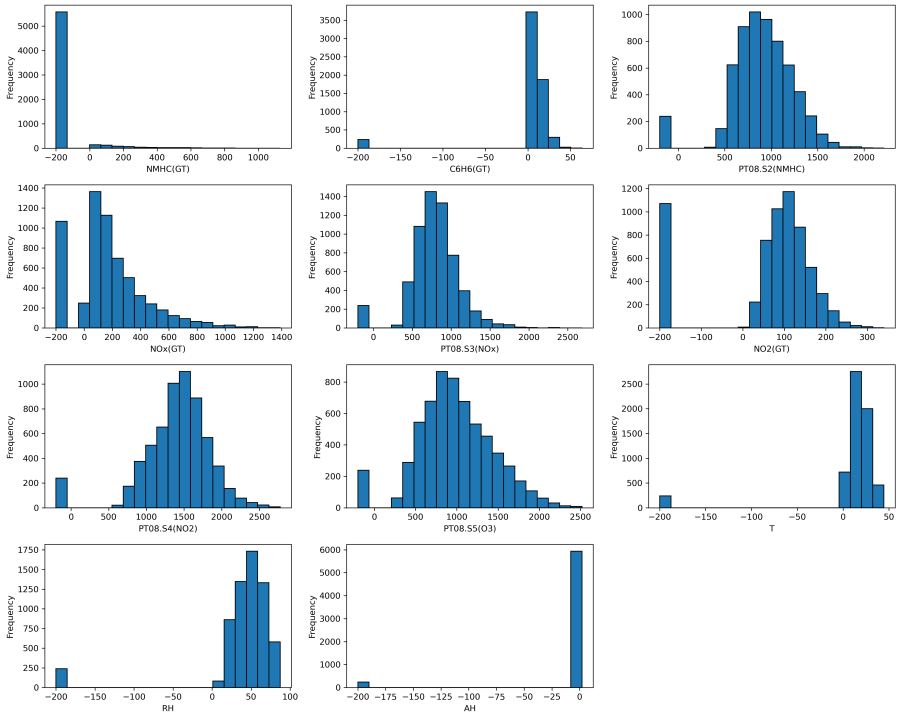
(4) The training data has these features - `NMHC(GT)`, `C6H6(GT)`, `PT08.S2(NMHC)`, `NOx(GT)`, `PT08.S3(NOx)`, `NO2(GT)`, `PT08.S4(NO2)`, `PT08.S5(O3)`, `T`, `RH`, `AH`. The target or label name is `PT08.S1(CO)`.

(2) Exploratory Data Analysis

(1) Figure 1 represent the histograms of all 11 features in the training dataset. It is clear that the features exhibit a variety of distributions. Several sensor readings such as `PT08.S2(NMHC)`, `PT08.S4(NO2)`, and `PT08.S5(O3)` appear roughly symmetric and unimodal, suggesting near-normal distributions. However, features like `NMHC(GT)`, `NOx(GT)`, and `NO2(GT)` are heavily right-skewed and display long tails. Others such as `C6H6(GT)` and `AH` include notable outlier bars at extreme values, likely indicating missing or erroneous entries (e.g., -200 or -100), which need to be handled carefully. In this assignment, these extreme values are not addressed.

In addition, most features have different ranges and scales. To ensure that no single feature dominates due to scale differences - especially important in algorithms that rely on distance or gradient magnitudes— the function `MinMaxScaler` is written and applied to all features. This transformation scales each feature to the range [0, 1], preserving relative shapes but aligning them to a uniform numerical range. The normalization was implemented in the `DataProcessor` module and enables better comparability across features while preparing the dataset for statistical learning tasks.

Figure 1: Histogram of all features and the target variable.

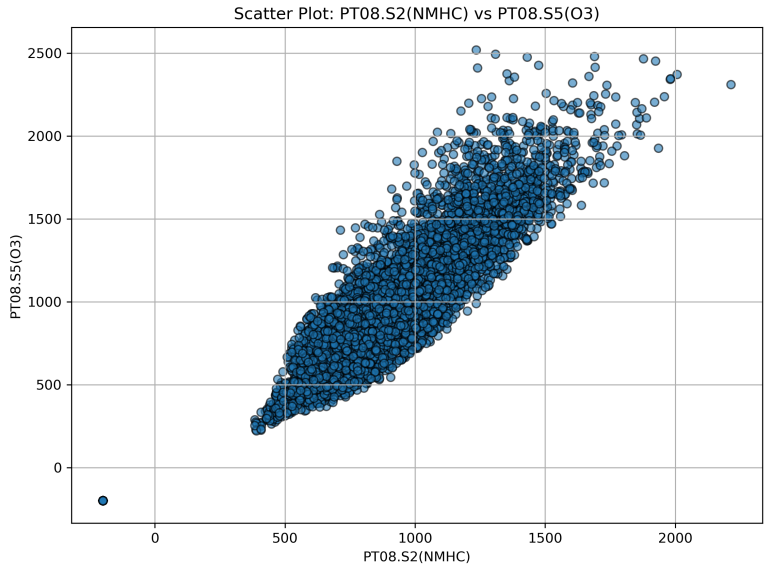


(2) To explore feature relationships, a scatter plot is generated comparing `PT08.S2(NMHC)` and `PT08.S5(O3)`, which represent the sensor responses to NMHC and O_3 respectively. The scatter plot shows a strong positive linear pattern, where increases in the sensor response for NMHC are consistently associated with increases in the sensor response for O_3.

This suggests a high correlation between the two features. Visually, the points are tightly clustered along a positively sloped band, indicating that one sensor's reading could reasonably predict the other. There is a visible outlier in the lower left corner, but the overall trend remains strong. This relationship may arise due to common environmental factors or shared measurement dependencies in sensor hardware.

Given this apparent correlation, it may be worth checking the Pearson correlation coefficient to quantify the relationship.

Figure 2: Scatter plot of `PT08.S2(NMHC)` and `PT08.S5(O3)`.



(3) To analyze pairwise relationships among the features, the Pearson's correlation coefficient for all 12 variables are computed and the results are stored in a 12×12 matrix C, where $C(i, j) = \text{corr}(i, j)$. The resulting matrix is shown in Figure 3.

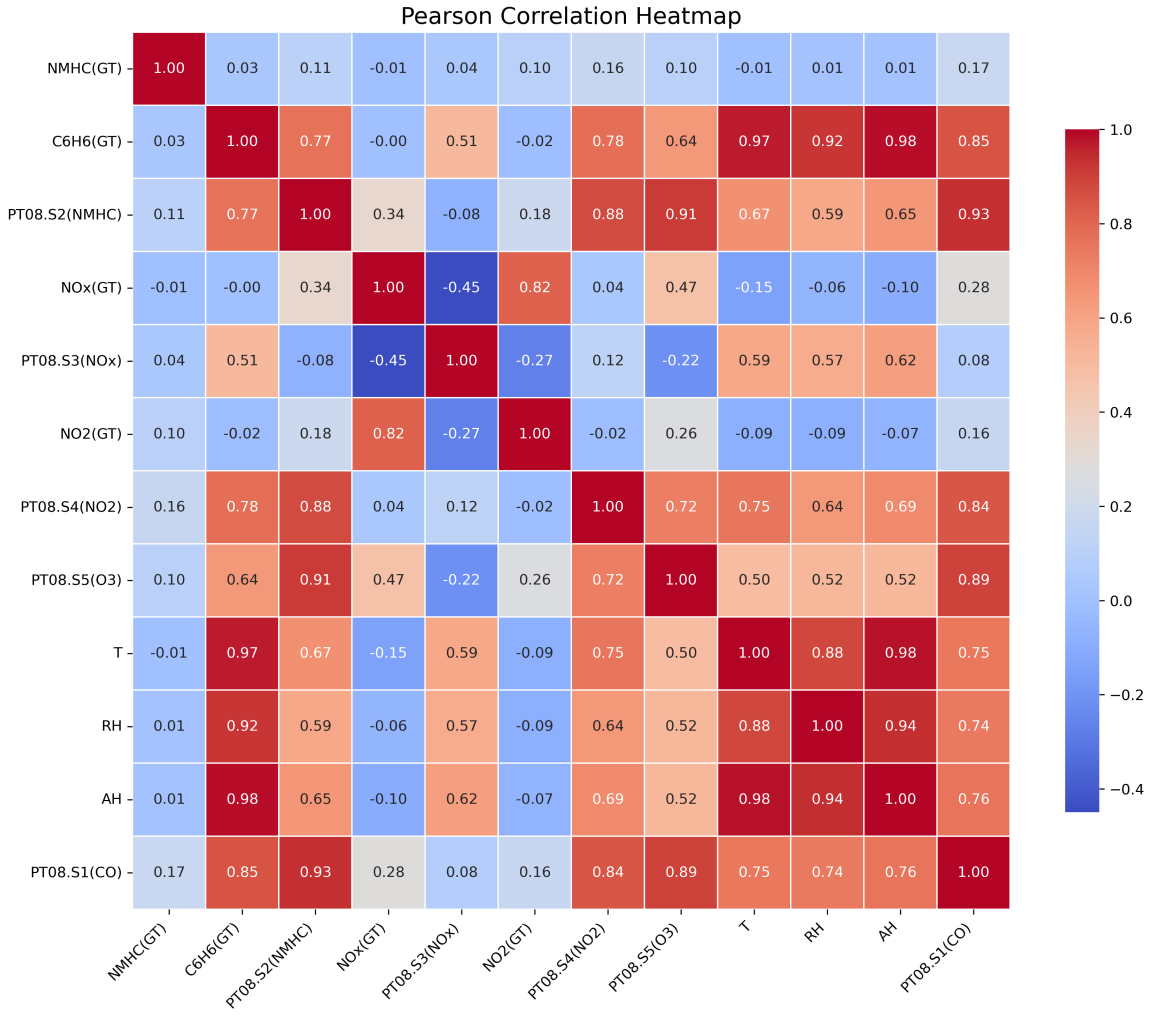
From the heatmap, several strong linear relationships are evident. For instance, `PT08.S2(NMHC)` shows a very high positive correlation with `PT08.S1(CO)` (0.93), `PT08.S4(NO2)` (0.88), and `C6H6(GT)` (0.77). This suggests that these sensor responses are likely influenced by similar environmental factors or may even be functionally linked in how they respond to gas concentrations.

Another notable cluster is the strong interdependence among meteorological variables: `T` (temperature), `RH` (relative humidity), and `AH` (absolute humidity), which exhibit very high mutual correlations (e.g., `T` and `AH`: 0.98). These associations reflect expected physical relationships between temperature and humidity measures.

It is also observed that there is moderate to strong correlation between `NOx(GT)` and `NO2(GT)` (0.82), which is consistent with their chemical similarity and shared emission sources. Sensor readings like `PT08.S5(O3)` show moderate correlation with both `PT08.S2(NMHC)` (0.47) and `PT08.S4(NO2)` (0.72), suggesting that O_3 sensor response may be partially explained by the levels of other pollutants.

Overall, the correlation matrix and heatmap reveal both expected relationships and latent dependencies across sensor signals and environmental variables. These patterns can inform feature selection, redundancy reduction, or multivariate modeling in subsequent analysis. It should be noted that feature selection and redundancy reduction are not addressed in this assignment.

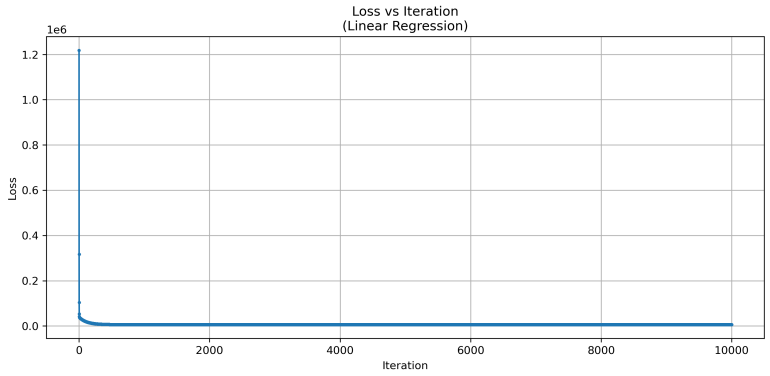
Figure 3: Pearson's correlation between all pairs of variables.



(3) Linear Regression Implementation

(3) Figure 4 shows the training loss per iteration for Linear Regression model.

Figure 4: Training loss per iteration - Linear Regression



(5) To tune the hyperparameters of the linear regression model, a systematic grid search was conducted over combinations of learning rate, L2 regularization strength (`l2_lambda`), and the maximum number of training iterations. Specifically, learning rates ranged from 0.01 to 0.1, `l2_lambda` values ranged from 0.0001 to 0.0008, and iterations varied across 5000, 10000, and 100000.

Each configuration was evaluated on the basis of the resulting root-mean-square error (RMSE). The detailed configurations and the corresponding RMSE's can found in the Table 1. The goal was to minimize RMSE while also considering model efficiency (i.e., avoiding unnecessarily large iteration counts). The sets of hyperparameters that performed the best were those that consistently achieved RMSE values close to 71.9. The final configuration selected for the linear regression model was: a learning rate of 0.05, L2 regularization parameter (`l2_lambda`) set to 0.0004, and a maximum iteration count of 10,000. This setup was chosen based on its ability to achieve RMSE values close to the target range (≤ 71) while maintaining computational efficiency.

Table 1: RMSE for Various Hyperparameter Combinations (Learning Rate, Iterations, and L2 Regularization)

LR	Iter	L2-reg	RMSE
0.1000	5000	0.0004	71.9168
0.1000	5000	0.0002	71.8412
0.1000	5000	0.0001	71.8100
0.0800	5000	0.0004	71.9849
0.0800	5000	0.0002	71.9152
0.0800	5000	0.0001	71.8865
0.1000	10000	0.0006	71.8896
0.1000	10000	0.0004	71.7762
0.1000	10000	0.0002	71.6783
0.1000	10000	0.0001	71.6377
0.0800	10000	0.0006	71.9168
0.0800	10000	0.0004	71.8108

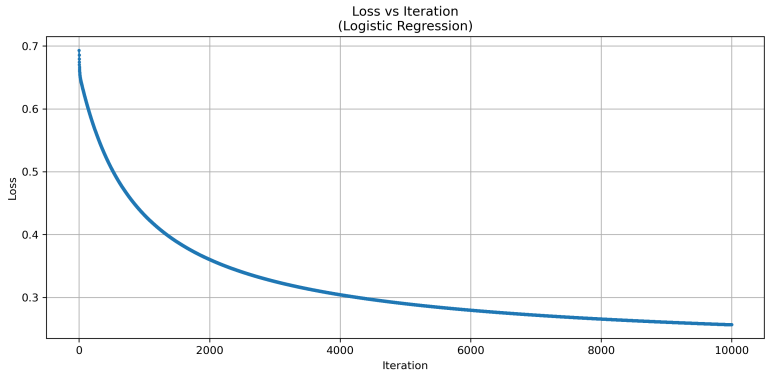
LR	Iter	L2-reg	RMSE
0.0800	10000	0.0002	71.7205
0.0800	10000	0.0001	71.6830
0.0500	10000	0.0004	71.9168
0.0500	10000	0.0002	71.8412
0.0500	10000	0.0001	71.8100
0.1000	50000	0.0008	71.9864
0.1000	50000	0.0004	71.8443
0.1000	50000	0.0002	71.6999
0.1000	50000	0.0001	71.5448
0.1000	100000	0.0008	71.4582
0.1000	100000	0.0006	71.9865
0.0800	50000	0.0006	71.8446

LR	Iter	L2-reg	RMSE
0.0800	50000	0.0004	71.7013
0.0800	100000	0.0002	71.5525
0.0800	100000	0.0001	71.4752
0.0500	100000	0.0001	71.4317
0.0800	100000	0.0006	71.9843
0.0500	100000	0.0006	71.8443
0.0500	50000	0.0002	71.5525
0.0500	50000	0.0001	71.4752
0.0500	50000	0.0008	71.9873
0.0500	50000	0.0006	71.8472
0.0500	50000	0.0004	71.7095
0.0500	50000	0.0002	71.5776

(4) Logistic Regression Implementation

(3) Figure 5 shows the training loss per iteration for Logistic Regression model.

Figure 5: Training loss per iteration - Logistic Regression



(5) To optimize model performance, a grid-based sweep was performed across relevant hyperparameters. For the logistic regression model, the key hyperparameters included the learning rate, probability threshold, and maximum number of iterations. Based on the outcomes of the hyperparameter search, the most appropriate configuration was selected (shown in bold in Table 2) by evaluating F1 score, AUROC, and convergence behavior (i.e., iteration count).

Table 2: Logistic Regression Hyperparameter Tuning Results

LR	Prob. Thresh	Loops	F1 Score	AUROC
0.1000	0.2500	50000	0.9016	0.9697
0.1000	0.3000	25000	0.9044	0.9686
0.1000	0.3500	10000	0.9043	0.9669
0.1000	0.4000	10000	0.9093	0.9669
0.0800	0.2500	50000	0.9040	0.9693
0.0800	0.3000	25000	0.9016	0.9684
0.0800	**0.3500**	**10000**	**0.9016**	**0.9663**
0.0800	0.4000	10000	0.9078	0.9664
0.0500	0.3000	25000	0.9001	0.9674
0.0500	0.3500	25000	0.9058	0.9674
0.0500	0.4000	10000	0.9028	0.9649
0.0200	0.3500	50000	0.9043	0.9670
0.0200	0.4000	25000	0.9034	0.9650
0.0100	0.4000	50000	0.9040	0.9651

(5) Result Analysis - Cross Validation

For the Linear Regression model, the RMSE consistently remained above the target threshold of 71. The best-performing configuration from earlier experiments (Problem 3, Part 5) achieved an RMSE of 71.9168. This persistent deviation suggests the model may be fundamentally limited in its ability to regress the noisy sensor data accurately. One notable issue is the presence of anomalous values such as −200, which likely indicate missing or corrupted sensor readings. These entries distort the loss landscape and can severely impair the optimization process. Although filtering out such values for improved model robustness was planned, time constraints prevented a deeper investigation. It is important to highlight that applying L2 regularization and performing K-Fold Cross-Validation did not yield improvements sufficient to bring RMSE below the desired threshold.

In contrast, the Logistic Regression model demonstrated strong classification performance. Both the F1 score and AUROC remained consistently high across folds, with averages above 0.90 and 0.96, respectively. These metrics suggest that the binary classification task (based on the `PT08.S1(CO)` thresholding at 1000) is well-captured by the logistic model. The high AUROC, in particular, indicates excellent separability between classes and a low false positive rate.

The `ModelEvaluator` class encapsulates the cross-validation logic for both models. Its `cross_validation` method utilizes 5-Fold Cross-Validation via `sklearn.model_selection.KFold`. The dataset is split into five subsets, and for each fold, one subset serves as the validation set while the remaining four are used for training. Each model is trained and evaluated independently on each fold. For Linear Regression, RMSE is computed; for Logistic Regression, both F1 score and AUROC are measured. The mean and standard deviation across all folds provide insight into model stability and generalization performance.

Cross-Validation Results:

- **Linear Regression**
  RMSE: Mean = 71.7718, Std Dev = 1.7759
- **Logistic Regression**
  F1 Score: Mean = 0.9004, Std Dev = 0.0088
  AUROC: Mean = 0.9663, Std Dev = 0.0033

These results reinforce the earlier conclusions: while linear regression struggles due to noise and potentially corrupted labels, logistic regression is more resilient and effectively captures meaningful patterns for binary clas-

sification. Future improvements could include outlier handling, alternative loss functions robust to label noise, or ensemble techniques.

No, we do not observe a significant change across different folds for either model. In the case of the Linear Regression model, the standard deviation of RMSE across the 5 folds was approximately 1.7759, indicating relatively stable performance despite the presence of noisy or possibly corrupted sensor values. This low variation suggests that while the overall RMSE remained above the target threshold, the model behaved consistently across different data splits.

For the Logistic Regression model, both evaluation metrics demonstrated even greater consistency. The F1 score had a standard deviation of only 0.0088, and the AUROC had a standard deviation of just 0.0033. These values reflect high model stability and generalization capability.

In summary, the variance across folds was minimal in both models, with Logistic Regression exhibiting especially consistent performance.

Feature Importance via Model Coefficients

The linear regression model provides a straightforward method for interpreting feature importance through its learned coefficients. Since all features were normalized to a common scale prior to training, the magnitude of each weight directly reflects the influence of the corresponding feature on the model's prediction. In other words, features with larger absolute weights contribute more significantly to changes in the target variable, while features with smaller weights have comparatively less impact. Table 3 depicts the linear regression coefficients across five folds for each normalized feature.

Table 3: Linear regression coefficients across five folds for each normalized feature

Fold	NMHC	C6H6	PT08.S2	NOx	PT08.S3	NO2	PT08.S4	PT08.S5	T	RH	AH
1	289.14	350.19	544.50	107.31	-282.16	-16.50	171.85	586.72	6.09	303.73	241.00
2	286.76	373.82	555.83	109.71	-276.32	-17.70	179.55	575.33	-16.72	298.83	240.43
3	289.48	383.13	550.02	95.99	-275.48	-13.30	180.97	583.01	-25.30	293.06	243.88
4	291.28	392.39	550.29	99.25	-278.36	-15.41	172.70	581.40	-32.94	297.58	245.56
5	298.80	391.59	553.25	107.14	-276.13	-18.98	177.20	575.73	-30.47	290.87	245.23

To assess consistency and robustness of feature importance, the coefficients obtained from all five folds of cross-validation are analyzed. Below is a summary of the average absolute weight for each feature across all folds:

- `PT08.S5(O3)`: Consistently has the highest positive weight (avg $\approx$ 580), indicating strong contribution.
- `C6H6(GT)` and `PT08.S2(NMHC)`: Also show high weights (avg $\approx$ 380 and 550), reflecting strong predictive signal.
- `NMHC(GT)` and `RH`: Moderate weights (avg $\approx$ 291 and 296) but still informative.

Because the input features were normalized, we can confidently interpret that `PT08.S5(O3)`, `PT08.S2(NMHC)`, and `C6H6(GT)` are the most informative features in estimating the target sensor response for CO.

(6) ROC Curve - Logistic Regression

To evaluate the performance of the logistic regression model across multiple validation splits, the ROC curves are plotted for each fold using 5-fold cross-validation. The Area Under the ROC Curve (AUROC) was computed for each fold, as shown in the legend of the ROC plot.

- Fold 1: AUROC = 0.969
- Fold 2: AUROC = 0.970
- Fold 3: AUROC = 0.964
- Fold 4: AUROC = 0.966
- Fold 5: AUROC = 0.962

The ROC curves, in Figure 6, consistently show high true positive rates across all thresholds, and all AUROC values are above 0.96. This indicates that the logistic regression model performs very well in distinguishing between classes, with only minor variance across folds.

The small spread in AUROC values suggests that the model generalizes well across different data splits and is not overly sensitive to a particular subset of training or validation data. Overall, the result is highly consistent and demonstrates the stability and reliability of logistic regression for this classification task.

Figure 6: ROC Curve per fold - Logistic Regression