5304 – Linear Models for Regression and Classification

Linear Regression

**Introduction**

The goal of this project was to design and implement a custom Linear Regression class. The class includes the main methods fit, predict, score, and RMSE. An extended version with L2 regularization was also built to study the effect of penalizing large weights. The experiments were performed on the Iris dataset (sepal length, sepal width, petal length, petal width). Four regression models with different input–output combinations were trained and compared. One model was further analyzed with and without regularization, and the class was also applied to a multi-output regression task.

The Iris dataset was split into 90% training and 10% testing, ensuring stratification so that all three classes were evenly represented. Additionally, within training, 10% of the data was used as a validation set for early stopping.

4 input–output feature combinations were used for regression

* Model 1: Sepal length, Sepal width, Petal length → Petal width
* Model 2: Sepal length, Petal width → Petal length (alt 2-input model)
* Model 3: Sepal length, Petal length, Petal width → Sepal width
* Model 4: Sepal width, Petal length, Petal width → Sepal length

Implementation

* Fit Method: trains the model using mini-batch gradient descent with support for early stopping and optional L2 regularization. Parameters include input/output data, batch size, max epochs, patience, learning rate, and λ (regularization strength). Best weights and bias are restored at the end.
* Predict Method: computes predictions using  
  .
* Score Method: evaluates performance using Mean Squared Error (MSE).
* RMSE Method: returns the root mean squared error for interpretability.
* Save/Load: model parameters (weights, bias) can be saved in .npz files and reloaded later.
* L2 Regularization Subclass: extends the base class, calls fit with regularization, and stores weights/loss for comparison.

Train regression 1 - Sepal Length, Sepal Width, Petal Length vs Petal Width

A graph with a blue line

AI-generated content may be incorrect.

Model 1 (Sepal length, Sepal width, Petal length → Petal width)

* Weights: [-0.1537, -0.2646, 0.3982]
* MSE (test): 0.0574
* Training loss decreased smoothly, converging within ~15 epochs.
* Petal length has the strongest positive influence, while sepal features contribute weakly and negatively.
* Low MSE shows good generalization and accurate predictions.

Train regression 2- sepal length, petal length vs petal width

A graph of a training loss

AI-generated content may be incorrect.

2-Input Model (Sepal length, Petal length → Petal width)

* Weights (no reg): [-0.1890, 0.4721]
* Weights (with reg): [-0.0144, 0.4237]
* MSE (test): 0.0537 (no reg), 0.0540 (with reg)
* Training loss converged smoothly within ~20 epochs.
* Regularization slightly reduced coefficient magnitudes without changing their influence.
* This model predicts petal width more accurately than other 2-input setups and approaches the performance of Model 1.

Train regression 3 - Sepal Length, Petal Length, Petal Width vs Sepal Width

A graph with a line

AI-generated content may be incorrect.

Model 3 (Sepal length, Petal length, Petal width → Sepal width)

* Weights: [0.4425, -0.2939, 0.0287]
* MSE (test): 0.0326
* Loss dropped sharply in the first 10 epochs and then stabilized, showing quick convergence.
* Sepal length is the strongest positive predictor, petal length contributes negatively, and petal width has little effect.
* Lowest MSE so far, outperforming Model 1 (0.0574) and Model 2 (0.1245).

Train Regression 4 - Sepal Width, Petal Length, Petal Width vs Sepal Length

A graph with a line

AI-generated content may be incorrect.

Model 4 (Sepal width, Petal length, Petal width → Sepal length)

* Weights: [0.7151, 0.4955, -0.0288]
* MSE (test): 0.0990
* Loss dropped steeply in the first 10 epochs and quickly stabilized.
* Sepal width and petal length are strong positive predictors, while petal width has little effect.

Regression Multi Outputs (Sepal length, Sepal width -> Petal width, Petal Length)

A graph with a line

AI-generated content may be incorrect.

Predicted values (first 5):

 [[4.95948521 1.69010749]

 [4.6376861 1.56082063]

 [4.71456533 1.59361192]

 [4.09018691 1.34323603]

 [5.88397414 2.05653435]]

True values (first 5):

 [[4.6 1.3]

 [5.6 1.4]

 [5.2 2.0]

 [3.9 1.1]

 [6.3 1.8]]

* Mean Squared Error on Test Set = 0.3151721794996512
* Predicted vs True (first 5): close overall, with small deviations (e.g., 1.69 vs 1.3).
* MSE (test): 0.3152
* Training loss decreased steadily over 50 epochs, showing stable convergence.
* The model captures relationships well but with higher error than single-output models, as predicting two outputs is more complex.

**Training Loss curves -**  All models showed **smooth decreases in loss** and stabilized within 10–20 epochs. Model 3 converged the fastest with the lowest loss, while Model 2 showed higher error and slower convergence.

**features most predictive of the targets -**

* + **Petal length** strongly predicts petal width (Model 1).
  + **Petal width** strongly predicts petal length (Model 2).
  + **Sepal length** is the best predictor of sepal width (Model 3).
  + **Sepal width** and petal length help predict sepal length (Model 4).

**multi-output regression compare to single-output regression**

* + The multi-output model (sepal features → petal length & width) had a higher test MSE (0.3152) compared to single-output models (best: Model 3 with 0.0326).
  + This shows predicting multiple outputs at once is more complex and introduces higher error, though relationships were still captured reasonably well.

**What was the effect of L2 regularization?**

* + L2 regularization slightly reduced coefficient magnitudes, keeping weights smaller. On the Iris dataset, it did not significantly reduce test MSE since the dataset is small and clean with little risk of overfitting.

Logistic Regression

Logistic Regression is a supervised classification algorithm used to predict discrete class labels. For binary classification, it uses the sigmoid function to map outputs to probabilities. For multi-class problems (like the Iris dataset), it uses the softmax function to assign probabilities across multiple classes, ensuring they sum to 1. The predicted class is the one with the highest probability.

How the Logistic Regression class is structured (fit, predict, Accuracy):

* fit: Trains the model by optimizing weights and bias using gradient descent.
* predict: SoftMax applies to the model outputs and selects the class with the highest probability.
* Accuracy: Compares predicted labels against true labels and reports the percentage of correct predictions.

That the fit method uses gradient descent + cross-entropy loss, with optional L2 regularization.  
The fit method minimizes the cross-entropy loss, which measures the difference between predicted probabilities and true labels. To prevent overfitting, an L2 regularization term can be added, which penalizes large weight values. Gradient descent is used to update parameters iteratively until convergence.

That you will test it on 3 feature variants (petal, sepal, all features):  
The classifier is trained and evaluated on three different input feature sets to compare performance:

1. Petal length and petal width
2. Sepal length and sepal width
3. All four features together

Train Classifier Scripts   
These scripts load the Iris dataset with different feature variants (petal features, sepal features, or all features), train the model using gradient descent with cross-entropy loss, save the trained parameters (weights, bias, targets) into a .npz file, and generate decision region plots (for the 2D feature cases).

Eval Classifier Scripts   
These scripts load the saved model parameters from the training step, apply the model to the reserved test set, and compute the classification accuracy. This ensures reproducibility and provides a fair evaluation of each trained classifier.

Train Classifier 1 Petal Length and Petal Width VS Target

A diagram of a logistic regression

AI-generated content may be incorrect.

Features used:

* Petal length and petal width were used as inputs.

Training details:

* A custom Logistic Regression model was trained using gradient descent, cross-entropy loss, and L2 regularization (λ=0.01) for 1000 epochs with a learning rate of 0.01.

Accuracy:

* The model achieved 86.67% accuracy on the test set.

Visualization:

Petal features (length and width) provide strong discriminatory power. Setosa is perfectly classified, while Versicolor and Virginica show partial overlap, explaining why test accuracy was 86.67% instead of closer to 100%.

Train Classifier 2 Sepal Length and Sepal Width VS Target

A diagram of a logistic regression

AI-generated content may be incorrect.

Features used:  
• Sepal length and sepal width were used as inputs.

Training details:  
• A custom Logistic Regression model was trained using gradient descent, cross-entropy loss, and L2 regularization (λ=0.01) for 1000 epochs with a learning rate of 0.01.

Accuracy:  
• The model achieved 73.33% accuracy on the test set.

Visualization:  
• A decision region plot shows that Setosa is well separated, but Versicolor and Virginica overlap heavily in the middle region. This explains the lower accuracy compared to Classifier 1, showing that sepal features are weaker predictors than petal features.

Train Classifier All four features vs Target

What features were used for training this classifier?  
All four features of the Iris dataset: sepal length, sepal width, petal length, and petal width.

How was the Logistic Regression model trained (method, parameters)?  
The custom Logistic Regression model was trained using gradient descent, cross-entropy loss, and optional L2 regularization (set to 0.0) for 5000 epochs with a learning rate of 0.001.

Accuracy of Model - The model achieved 100% accuracy on the test set.

Was a visualization produced (decision regions)?

Visualization - No 2D visualization was produced because the classifier used all four features (a 4D input space).

All features as predictors - Using all features together provides the strongest predictive power, allowing the model to perfectly separate the three Iris species.