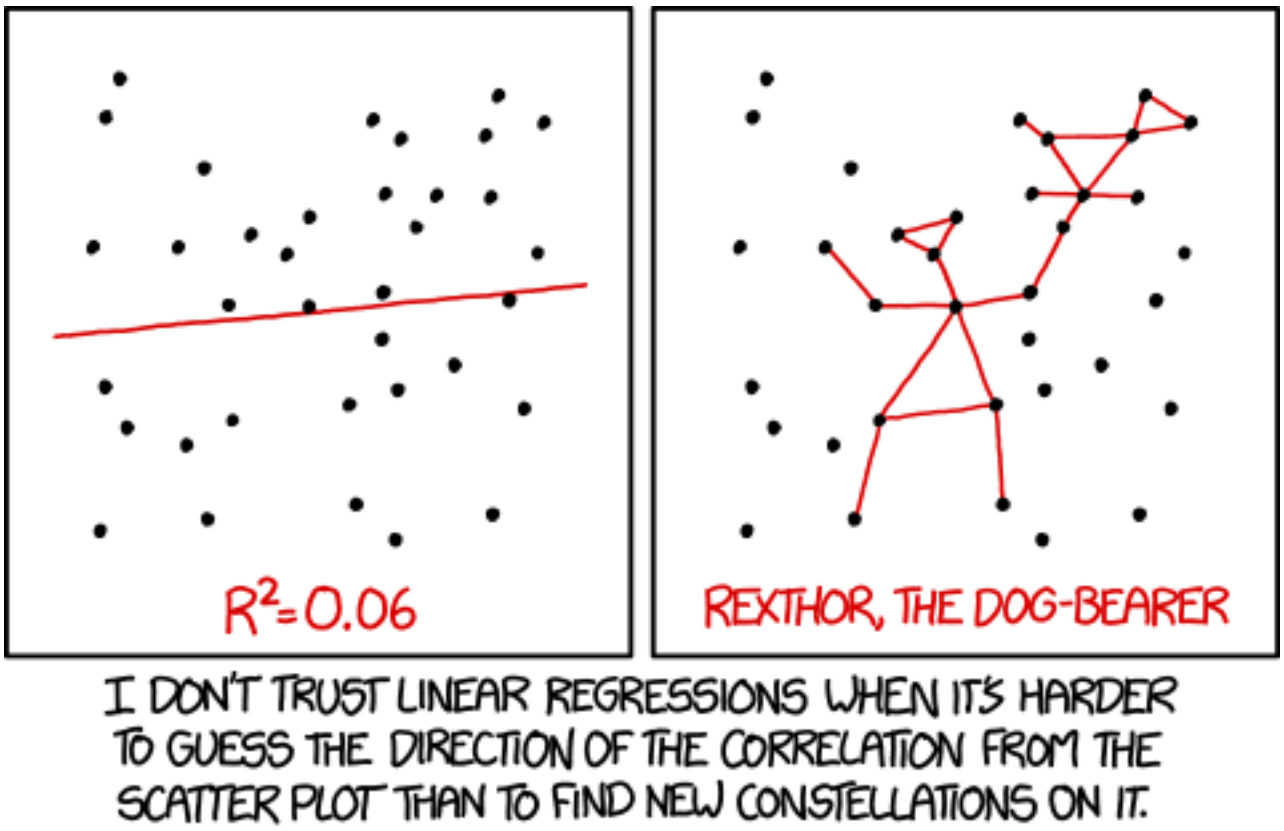
DASC 5304 - MACHINE LEARNING

ASSIGNMENT 1
LINEAR AND LOGISTIC REGRESSION



Introduction

This report records the findings of Linear Regression and Logistic Regression implemented from scratch for DASC 5304 - Machine Learning under Dr. Soumaya Gharsallaoui. This report is part of Assignment 1 and is submitted by Vandit Goel (MavID: 1002245699).

This reported is divided into 2 parts:

1. **Linear Regression**: Predicting features of the Iris Dataset
2. **Classification (Logistic Regression)**: Classify the iris data into target variables.

Part 1 - Linear Regression

For this part, 6 models have been created for predicting the following:

1. Sepal Length → Sepal Width
2. Petal Length → Petal Width
3. Sepal Features → Petal Length
4. All Features → Petal Width (w & w/o Regularization)
5. Sepal Features → Petal Features (Multi-output)

A simple Linear Regression model was implemented that utilizes Gradient Descent for optimizing the model parameters and Mean Squared Error as the loss function. It supports training with mini-batches, L2 regularization, and early stopping based on validation loss. Additionally, the model parameters can be saved to and loaded from disk for persistence.

Linear Regression Class Implementation

The **LinearRegression** class includes the following key methods:

- **fit**: Implements gradient descent with early stopping using validation data
    - Takes input data, target values, batch size, regularization factor, max epochs, and patience
    - Uses 10% of training data as validation set
    - Implements early stopping based on validation loss
    - Supports L2 regularization
    - Records training and validation losses for plotting
- **predict**: Takes input data and returns predicted values
- **score**: Computes mean squared error between predictions and targets
    - Takes input data and target values
    - Returns the MSE
- **save/load**: Methods to save and load model parameters
    - Uses pickle for serialization

Training and Evaluation Scripts

Separate scripts have been created for training and evaluating four different regression models:

1. **Model 1**: Predict sepal width from sepal length
2. **Model 2**: Predict petal width from petal length
3. **Model 3**: Predict petal length from sepal features (length and width)
4. **Model 4**: Predict petal width from all other features, with and without regularization
5. **Multi-Output Model**: Predict both petal length and width from sepal features

Each training script:

- Loads and prepares the Iris dataset
- Creates and trains the model using batch gradient descent
- Saves the trained model
- Plots and saves the training and validation loss curves

```
src/lr/modeling/training
├── __init__.py
├── train_regression1.py
├── train_regression2.py
├── train_regression3.py
├── train_regression4.py
└── train_regression_multi.py
```

Each evaluation script:

- Loads the test data
- Loads the trained model
- Evaluates the model using the score method
- Prints the test mean squared error

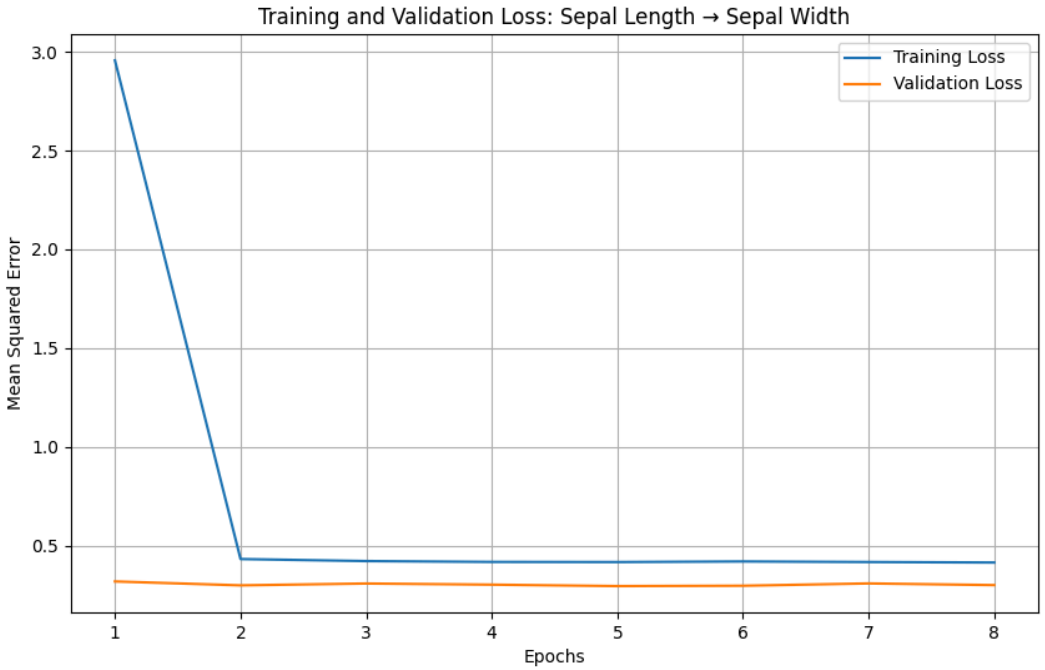
```
src/lr/modeling/evaluation
├── __init__.py
├── eval_regression1.py
├── eval_regression2.py
├── eval_regression3.py
```

```
├── eval_regression4.py
└── eval_regression_multi.py
```

Results

Sepal Length → Sepal Width

```
Learning Rate = 0.01
Batch Size = 32
Max Epochs = 100
Patience = 3
L2 regularization factor = 0
```



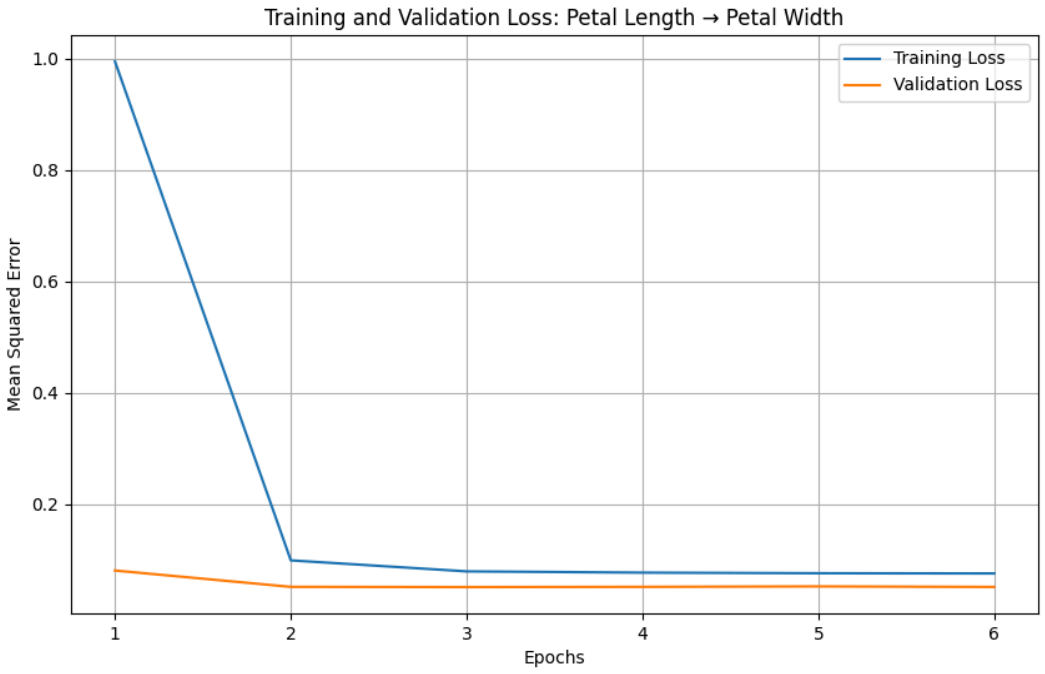

```
$ uv run src/lr/modeling/evaluation/eval_regression1.py
Model 1 - Sepal Length → Sepal Width
Test Mean Squared Error: 0.386749
```

The MSE drops drastically in just the second epoch and stays the same at about 0.3. Upon running the evaluation we find out the MSE is **0.386749**, which is significant.

Thus Sepal Length might not be the best predictor of Sepal Width alone.

Petal Length → Petal Width

```
Learning Rate = 0.01
Batch Size = 32
Max Epochs = 100
Patience = 3
L2 regularization factor = 0
```




```
$ uv run src/lr/modeling/evaluation/eval_regression2.py
Model 2 - Petal Length → Petal Width
Test Mean Squared Error: 0.066584
```

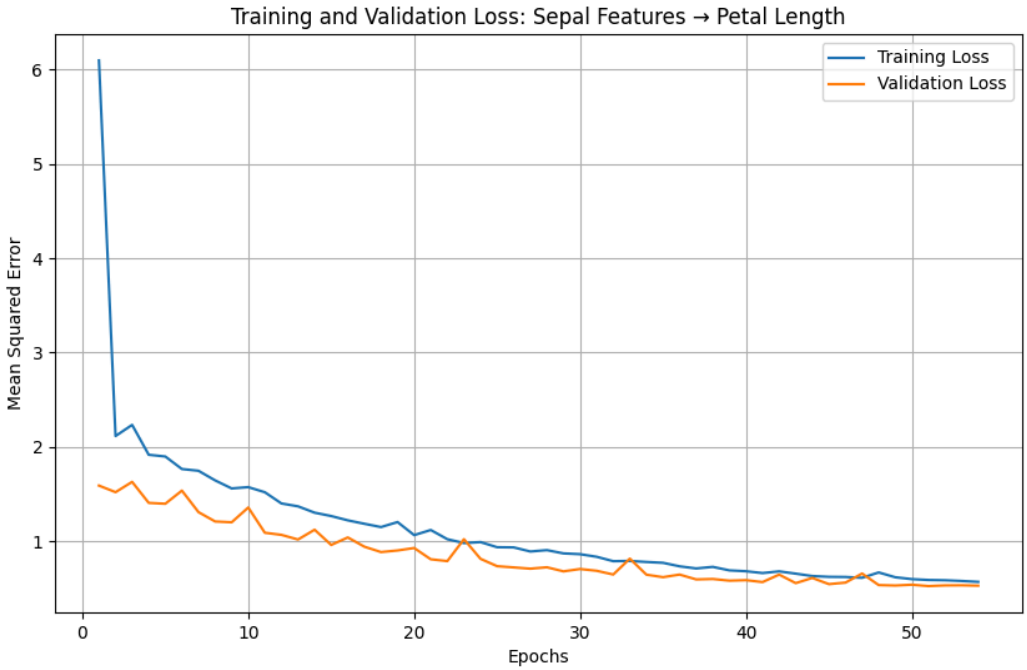
Again we see similar results but this time the MSE is significantly less. Upon running the evaluation we find out MSE is **0.066584.**

This suggests Petal Length is a good predictor of Petal Width.

Sepal Features → Petal Length

Here we train our model to predict Petal Length given Sepal Features (Length & Width). Up until now we trained the model with just one feature. This model is trained on 2 features.

```
Learning Rate = 0.01
Batch Size = 32
Max Epochs = 100
Patience = 3
L2 regularization factor = 0
```




```
$ uv run src/lr/modeling/evaluation/eval_regression3.py
Model 3 - Sepal Features → Petal Length
Test Mean Squared Error: 0.565610
```

Again the MSE is slightly on the higher side. Upon running the evaluation it came out to be **0.565610.**

Thus the Sepal Features might not be the best predictors of Petal Length alone.

All Features → Petal Width (w & w/o Regularization)

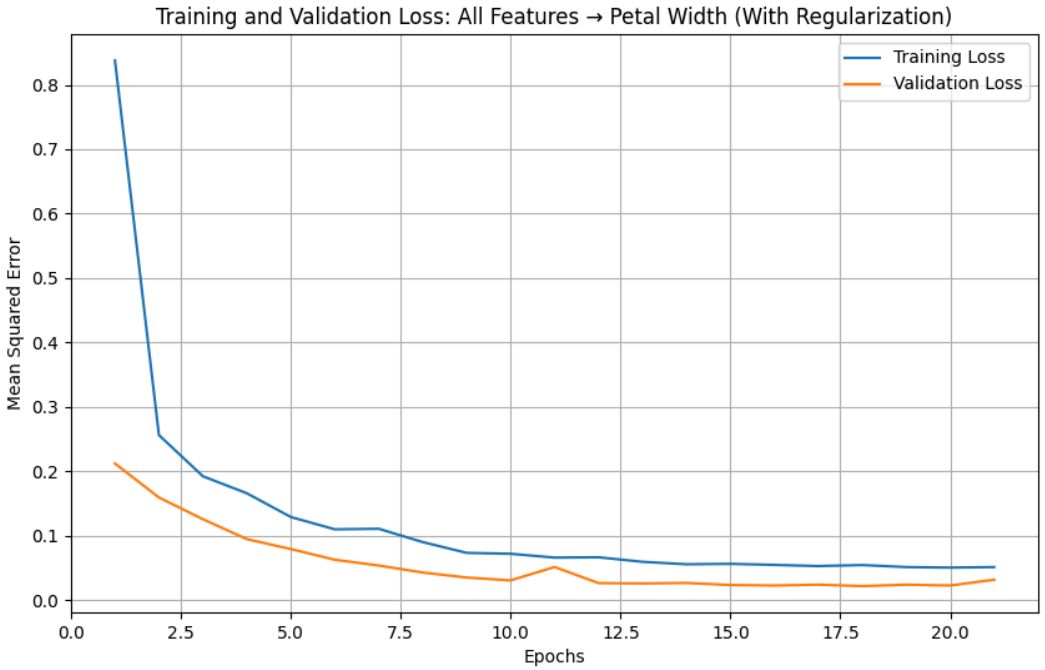
Here we train a model to predict the Petal Width with all the other features as input to the model. The same model is trained with the L2 Regularization Factor as well.

w/o Regularization

```
Learning Rate = 0.01
Batch Size = 32
Max Epochs = 100
Patience = 3
L2 regularization factor = 0
```

w Regularization

```
Learning Rate = 0.01
Batch Size = 32
Max Epochs = 100
Patience = 3
L2 regularization factor = 0.0005
```




```
$ uv run src/lr/modeling/evaluation/eval_regression4.py
Model 4 - All Features → Petal Width
Test Mean Squared Error (No Regularization): 0.078461
Test Mean Squared Error (With Regularization): 0.057575
Difference in MSE: 0.020886
```

All features (Sepal Length, Sepal Width, Petal Length) combined are good predictors of Petal Width. The MSE is quite less even without regularization. Thus the advantage of Regularization doesn't seem significant.

Thus, Sepal Length, Sepal Width, Petal Length are good predictors of Petal Width.

Part 2 - Logistic Regression

LogisticRegression Class Implementation

For this part, 3 models have been created for classifying the iris flower species with the following input features:

1. Sepal Length & Sepal Width
2. Petal Length & Petal Width
3. All Features

A simple Logistic Regression model was implemented that utilizes Gradient Descent for optimizing the model parameters.

Activation functions:

- **sigmoid** for binary classification
- **softmax** for multi-class classification

fit() method:

- Implements gradient descent with early stopping
- Handles both binary and multi-class classification

Training & Evaluation Scripts for Three Variants

Three training scripts have been implemented for different feature combinations:

1. **Model 1 (Classifier1)**: Using only petal length and width
2. **Model 2 (Classifier2)**: Using only sepal length and width
3. **Model 3 (Classifier3)**: Using all four features

Each script:

- Loads the Iris dataset
- Trains a logistic regression model

- Saves the trained model
- Generates training/validation loss plots
- For 2D feature models (1 and 2), visualizes decision boundaries

Each model has a corresponding evaluation script that:

- Loads the test data
- Loads the trained model
- Evaluates classification accuracy
- Generates a classification report and confusion matrix

A comparison script is also included that:

- Evaluates all three models on the test set
- Displays a bar chart comparing their accuracy
- Prints a summary of results

Results

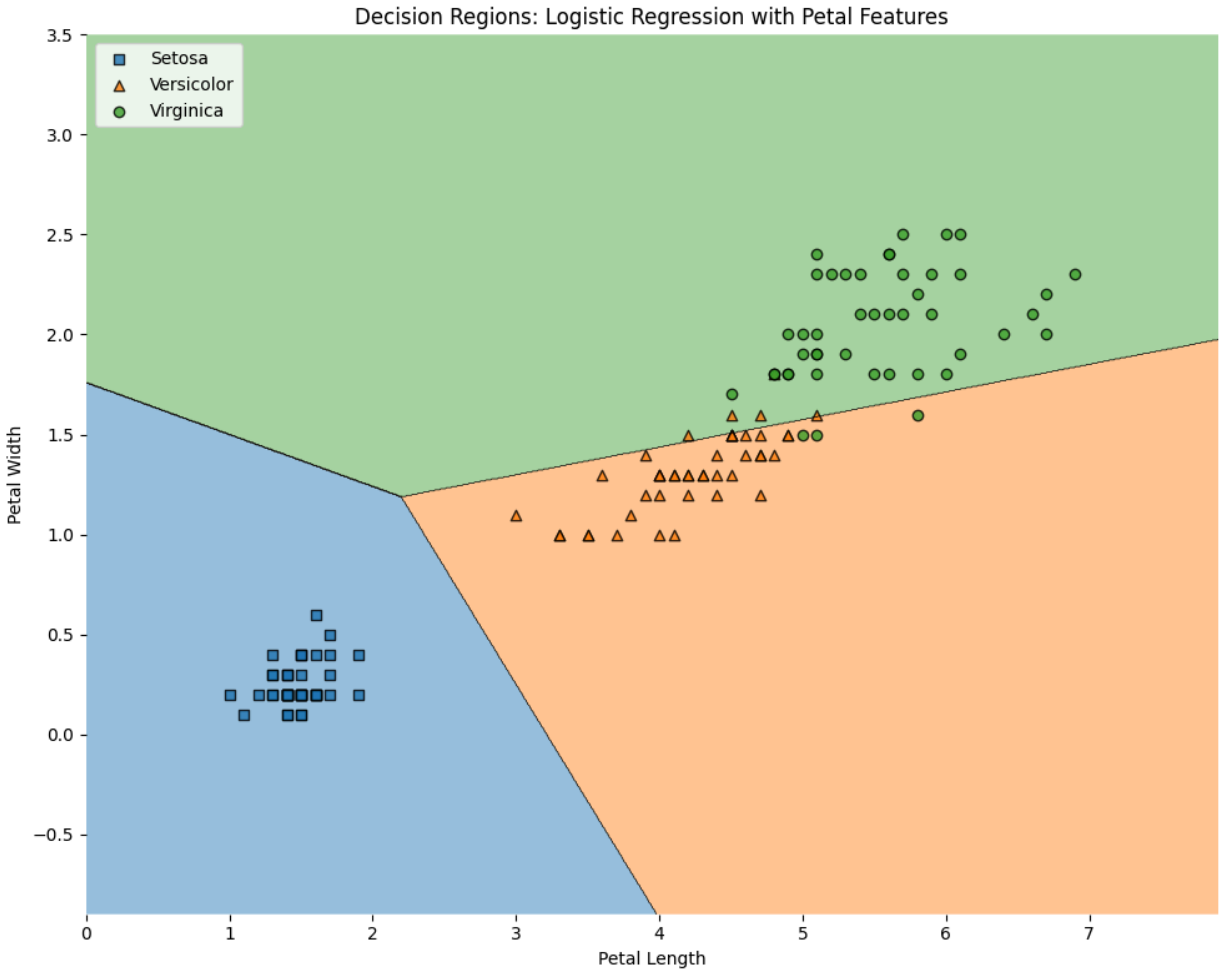
Sepal Length & Sepal Width

```
Learning Rate = 0.05
Batch Size = 32
Max Epochs = 200
Patience = 10
L2 regularization factor = 0
```

```
Training accuracy: 0.9407
```

```
Test Accuracy: 0.8667

Classification Report:
              precision    recall  f1-score   support

      setosa       1.00      1.00      1.00         5
  versicolor       0.80      0.80      0.80         5
   virginica       0.80      0.80      0.80         5

    accuracy                           0.87        15
```

```
   macro avg       0.87      0.87      0.87        15
weighted avg       0.87      0.87      0.87        15


Confusion Matrix:
[[5 0 0]
 [0 4 1]
 [0 1 4]]
```



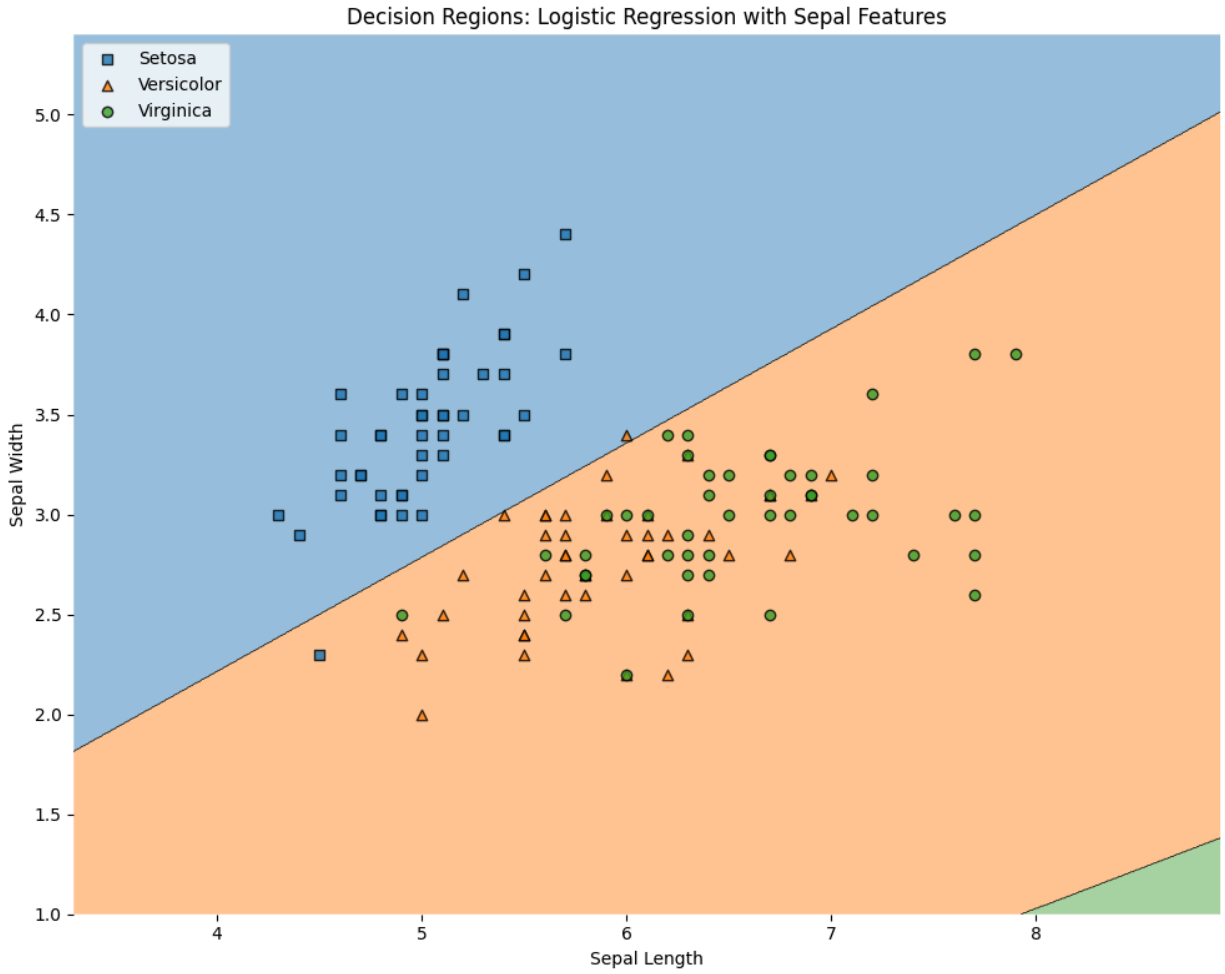
The classifier 1 performed really well.

Petal Length & Petal Width

```
Learning Rate = 0.05
Batch Size = 32
Max Epochs = 200
Patience = 10
L2 regularization factor = 0
```

```
Training accuracy: 0.6519
```

```
Test Accuracy: 0.6667

              precision    recall  f1-score   support

      setosa       1.00      1.00      1.00         5
  versicolor       0.50      1.00      0.67         5
   virginica       0.00      0.00      0.00         5

    accuracy                           0.67        15
   macro avg       0.50      0.67      0.56        15
weighted avg       0.50      0.67      0.56        15


Confusion Matrix:
[[5 0 0]
 [0 5 0]
 [0 5 0]]
```



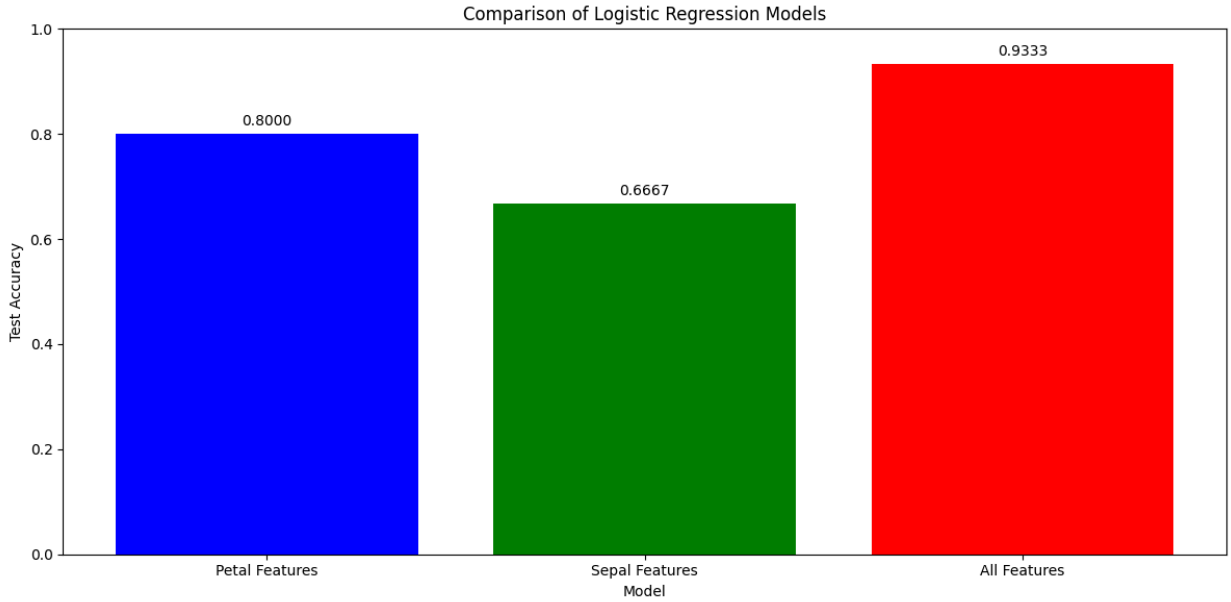
Classifier 2 did not perform that well and completely misclassified `virginica.`

All Features

```
Learning Rate = 0.05
Batch Size = 32
Max Epochs = 200
Patience = 10
L2 regularization factor = 0
```

```
Training accuracy: 0.9556
```

```
Test Accuracy: 1.0000

Classification Report:
              precision    recall  f1-score   support

      setosa       1.00      1.00      1.00         5
  versicolor       1.00      1.00      1.00         5
   virginica       1.00      1.00      1.00         5

    accuracy                           1.00        15
   macro avg       1.00      1.00      1.00        15
weighted avg       1.00      1.00      1.00        15


Confusion Matrix:
[[5 0 0]
 [0 5 0]
 [0 0 5]]
```



Classifier 3 performed the Best