Project Report: Using Linear Regression for Classification on the Iris Dataset

---

Team Members:

- Member 1: Herman Østengen
- Member 2: Isabel Wolden

---

1. Introduction

In this project, linear regression for classification of the Iris dataset will be investigated. The first part implements a binary classifier by reducing the dataset to two classes, then applying a linear regression model with a decision rule for classification. The second task extends to a multi-class scenario, thus requiring the development and evaluation of a multi-class classifier. While linear regression is traditionally designed for regression tasks, this report explores its utility for classification, highlighting its advantages, limitations, and experimental results.

---

2. Task 1: Binary Classification

2.1 Objective

In the first task, we implemented a binary classifier by dropping one class from the complete Iris dataset. We then trained sklearn's linear regression model to predict a continuous output, which is mapped to the binary classes using a decision rule.

2.2 Experimental Setup

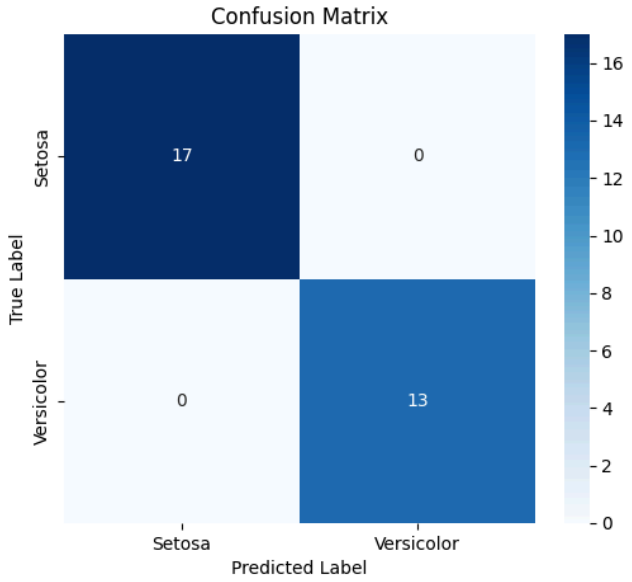
1. **Data Preparation**:
    - Two classes, Setose and Versicolor, were selected, dropping Virginica.
    - Features used: all four features of the Iris dataset.
    - Dataset split: 70% training, 30% testing.

2. **Model and Decision Rule**:
    - A linear regression model was trained on the binary target values. 0 for Setosa and 1 for Versicolor).
    - A decision rule was applied: if the predicted output is greater than or equal to 0.5, classify as Versicolor; otherwise, classify as Setosa.

3. **Evaluation Metrics**:
    - Accuracy and the confusion matrix were used to evaluate performance.

2.3 Results

- **Accuracy**: The binary classifier achieved perfect accuracy on the test set (100%)
- **Confusion Matrix**:



- **Figure 1:** We see from the confusion matrix that the model classified all cases in the testing dataset as true positives, giving it a 1.

Model visualization:

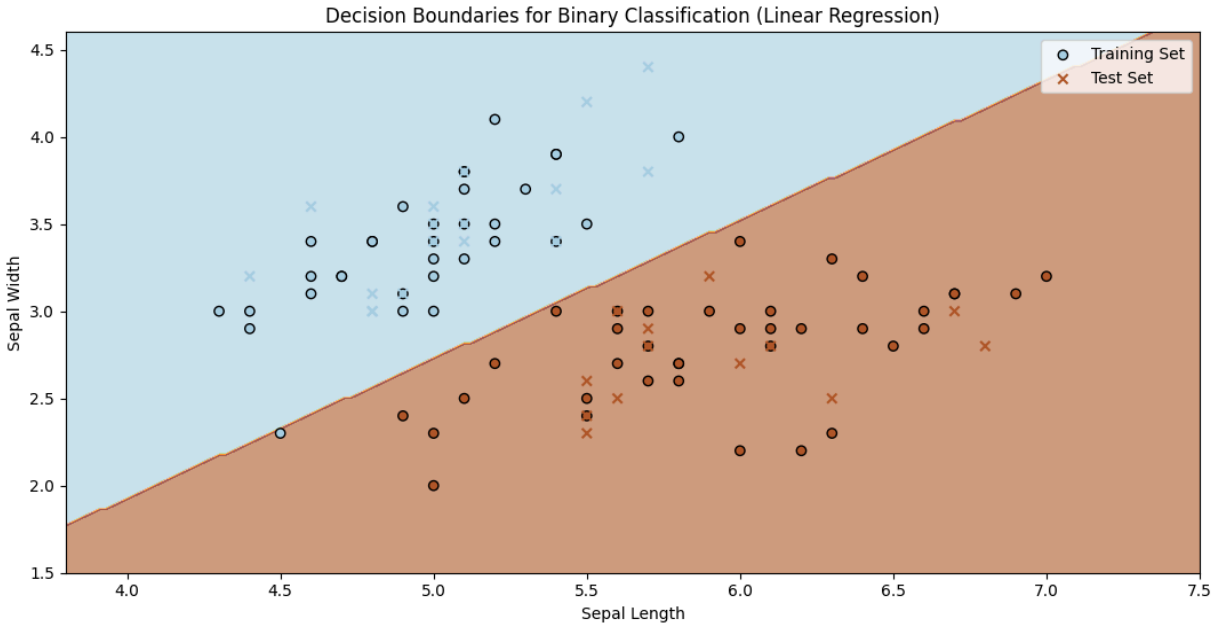


Figure 2: The linear graph separating the training and test sets of classes is produced by the model. This visualization only takes two out of four features into consideration, making it less precise in the interest of displaying the graph in 2 dimensions instead of 4.

2.4 Discussion

Does It Always Work?

Through the usage of linear regression for classification, it is possible to observe that the approach worked well for both Setosa and Versicolor. As their spaces exhibit clear separability, it is clear that the method proved efficient. The method, however, struggles with datasets or class combinations which are not linearly separable.

Advantages

The method is easy to implement and requires minimal computational resources, making it accessible for smaller datasets or scenarios where computational efficiency is essential. In addition to this, a wide range of libraries and tools are supported by linear regression. This allows for rapid prototyping and testing of classification tasks in simple cases.

Disadvantages

A disadvantage is that linear regression for classification assumes linear separability. This limits its applicability to more complex datasets. In the real world, raw data often exhibits nonlinear patterns, where linear regression fails to provide accurate predictions. Moreover, the continuous outputs in linear regression lack a probabilistic interpretation. This makes it less suitable for tasks requiring confidence estimates of probabilistic reasoning.

---

3. Task 2: Multi-Class Classification

3.1 Objective

The second task extends the binary classification approach to the three-class problem (Setosa, Versicolor, Virginica) using a One-vs-All (OvA) strategy.

3.2 Experimental Setup

1. **Data Preparation**:
    - All classes from the Iris dataset were included.
    - Features used: (a) Sepal length and width, (b) Petal length and width, evaluated separately.
    - Dataset split: 70% training, 30% testing.
2. **Model and Decision Rule**:
    - Three linear regression models were trained, one for each class, with binary target values (1 for the current class, 0 for others).
    - For prediction, the class with the highest predicted probability was selected.
3. **Evaluation Metrics**:
    - Accuracy and a confusion matrix were used to assess performance.

3.3 Results

- **Accuracy**:
    - Accuracy turned out to 82.22% on the test set due to overlapping features for Versicolor and Virginica.

Confusion Matrix:

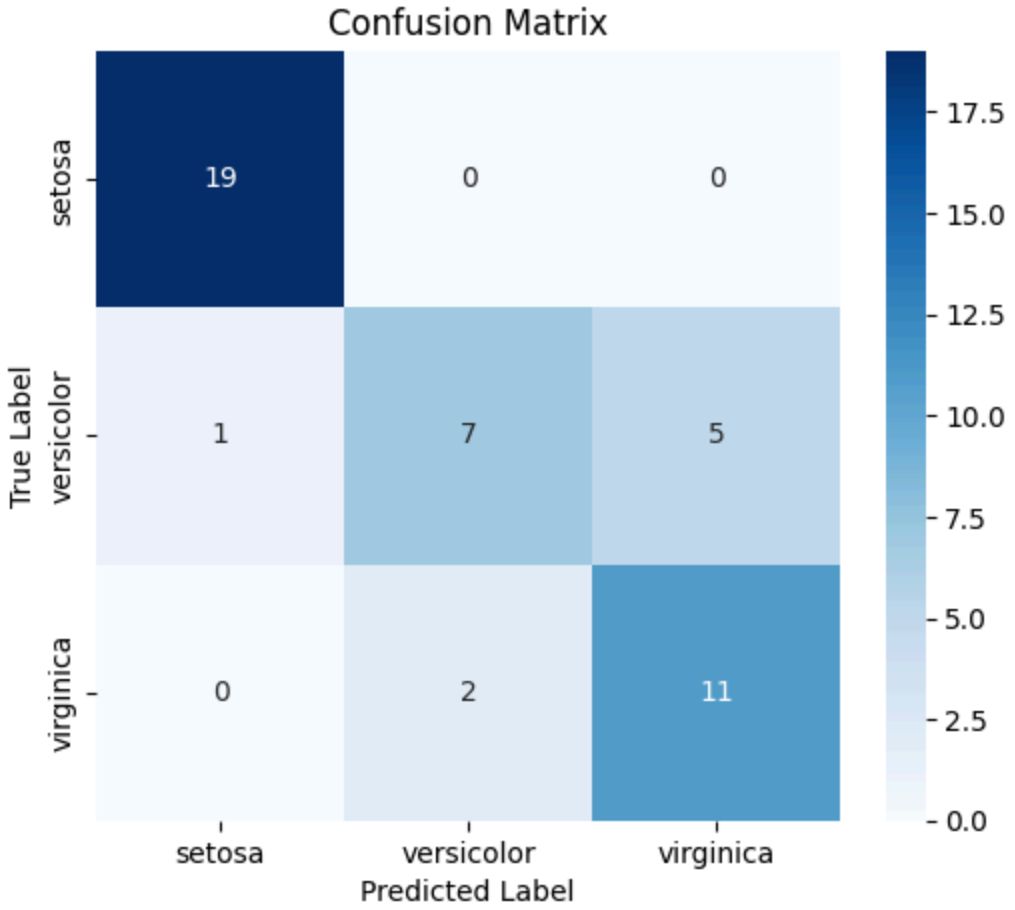
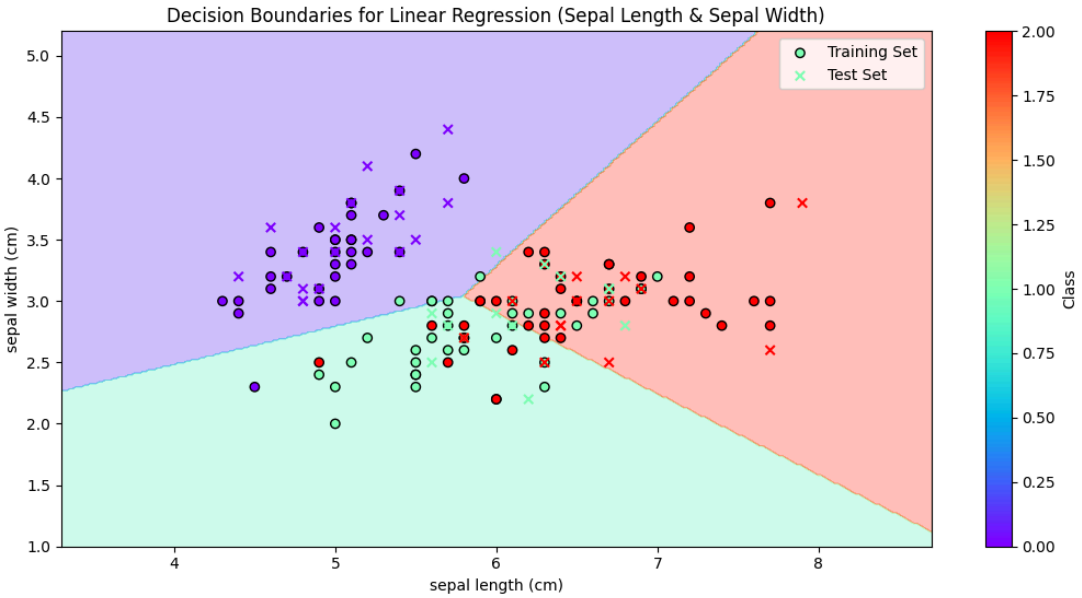


Figure 3: We see from the confusion matrix that the multi-class model misidentifies some flowers due to closely related features, especially between Versicolor and Virginica.

Model visualization:



- **Figure 2:** We can see that the three linear models separate the dataset into sections where the different classes have a higher likelihood of being placed. Once again, the visualization only considers two features, making it look like the predictions are of very low precision.

- **Decision Boundaries**:
    - Linear decision boundaries for Sepal length and Sepal width were less effective compared to Petal length and Petal width.

3.4 Discussion

- **Does It Work?**
    - The OvA strategy successfully extended linear regression to a multi-class problem but struggled with overlapping classes (Versicolor and Virginica).
- **Problems Identified**:
    - Classes with overlapping features lead to misclassifications.
    - Lack of a probabilistic interpretation for predictions makes the model less robust.
- **Advantages**:
    - Simple to implement and interpret.
    - Works well for linearly separable data.
- **Disadvantages**:
    - Poor performance for non-linear class boundaries.
    - Sensitive to feature scaling and data distribution.

---

4. Conclusion

Linear regression can be adapted for classification tasks using decision rules and the One-vs-All strategy. While the method is simple and effective for linearly separable datasets, it faces significant challenges in non-linear or overlapping class scenarios. For future work we could explore alternative models like logistic regression or Support Vector Machines (SVMs) for better performance on non-linear class boundaries.

---

5. Member contribution

Isabel: Task 1 code and writing report.
Herman: Task 2 code and writing report.