Lelio GUALINO

Youri HALMAERT

Report on Binary and Multi-Class Classification: Linear Regression vs. K-Nearest Neighbors (KNN)

Task 1: Binary Classification Using Linear Regression

In this task, we are required to develop a binary classifier using linear regression, while ignoring the fact that the target variable is binary. For this, we used the **Iris dataset**, which contains three classes: **Setosa**, **Versicolor**, and **Virginica**. Our goal was to fit a linear regression model and apply a decision rule to classify the data points into two categories (e.g., Versicolor and Virginica).

Data Preprocessing

We first filtered the dataset to include only two classes which we can choose with an input, thus simplifying the problem into a binary classification task. Each data point in the dataset consists of the following features:

- Sepal length
- Sepal width
- Petal length
- Petal width

We transformed the 'variety' column (representing the class) into binary labels: 0 for the first flower and 1 for the second one.

Model Training

For the binary classification, we used Linear Regression despite its intended purpose for regression tasks. The linear regression model works by minimizing the sum of squared differences between the predicted values and actual values, thereby learning the best-fitting line.

After splitting the dataset into training and testing sets (70% for training and 30% for testing **for each flower**), we trained the model on the training data. The model was then used to predict the class labels of the test data.

Decision Rule

Since the model outputs continuous values (real numbers), we applied a simple decision rule:

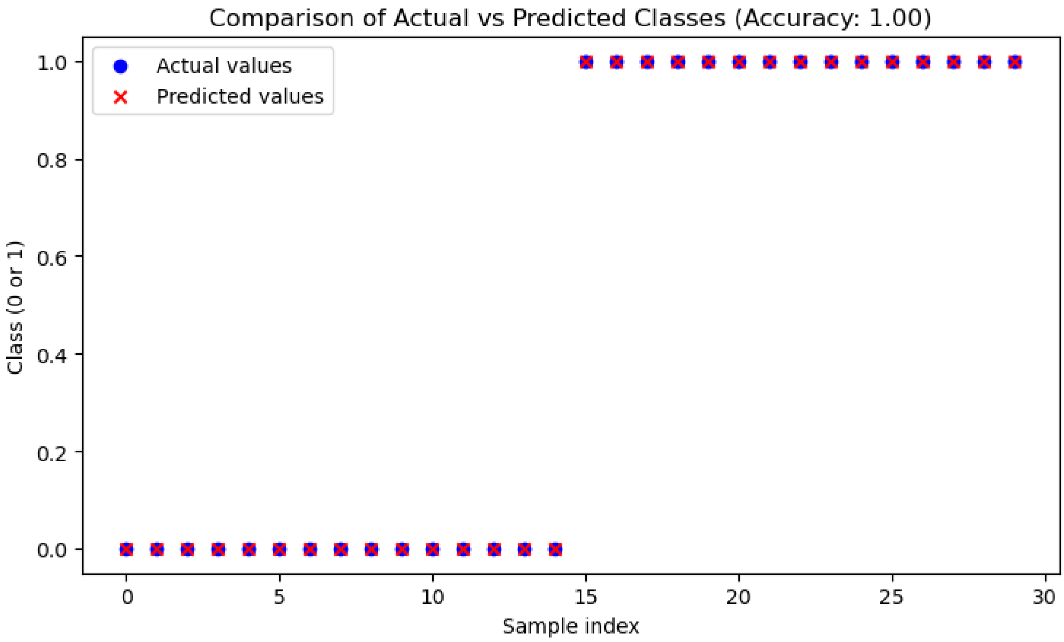
- If the predicted value is greater than or equal to 0.5, we classify the sample as class **1** (Virginica).
- If the predicted value is less than 0.5, we classify the sample as class **0** (Versicolor).

Lelio GUALINO

Youri HALMAERT

Results and Analysis

The model worked perfectly, but linear regression does not directly suit classification tasks. The output is continuous, and predictions may fall outside the [0, 1] range, leading to potential misclassifications.



Task 2: Multi-class Classification Using K-Nearest Neighbors (KNN)

In this task, we extended the classification problem to multi-class classification by keeping all three classes: Setosa, Versicolor, and Virginica.

Approach

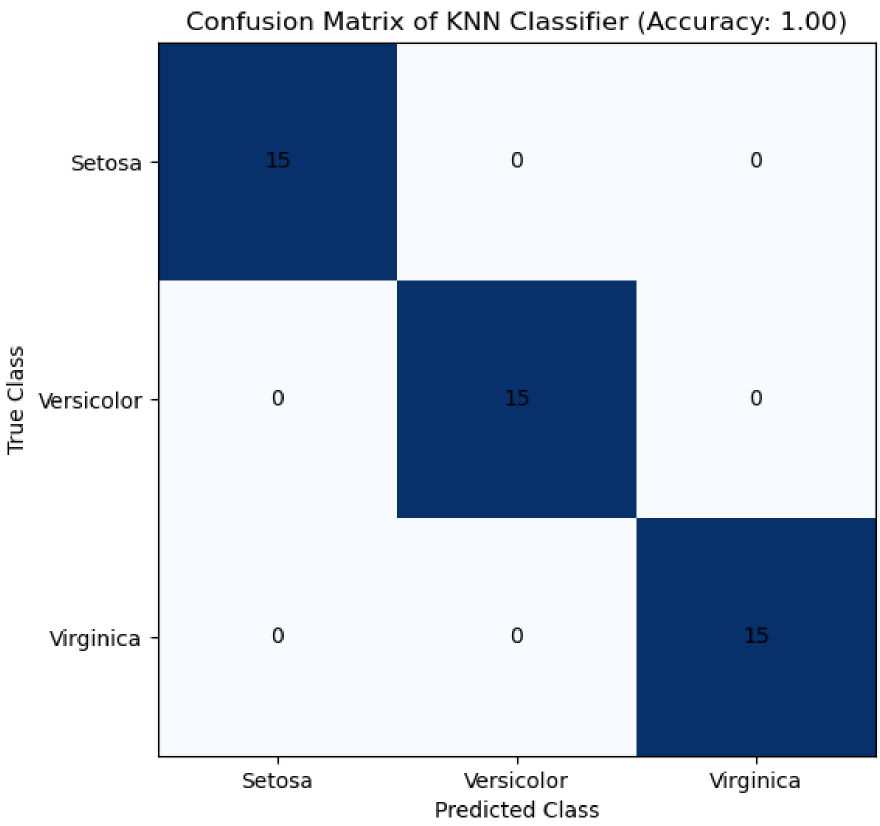
Since Linear Regression is not ideal for multi-class classification, we implemented the K-Nearest Neighbors (KNN) algorithm, which is more suitable for multi-class classification tasks. KNN works by finding the k-nearest neighbors of a data point and predicting its class based on a majority vote among those neighbors. In this case we used KNN with k=5 after different tries, meaning the class of each test point was determined by the 5 nearest neighbors in the training set.

Results and Analysis

The KNN model performed well, achieving a good classification accuracy on the Iris dataset, outperforming the linear regression approach. KNN, unlike linear regression, does not suffer from issues related to continuous outputs and linear decision boundaries.

Lelio GUALINO

Youri HALMAERT



Discussion: Advantages and Disadvantages of Using Linear Regression and KNN for Classification

Linear Regression

Advantages:

- **Simplicity**: Linear regression is easy to understand and quick to implement.
- **Speed**: Training and predicting are fast, especially on smaller datasets.

Disadvantages:

- **Not Designed for Classification**: Linear regression was not created for classification. It does not inherently model probabilities or output class probabilities in the range [0, 1].
- **Linear Decision Boundaries**: The model produces a linear decision boundary, which might not be effective for more complex datasets.
- **Out-of-Range Predictions**: Linear regression can produce predictions outside the expected range [0, 1], leading to potential misclassifications.

Lelio GUALINO

Youri HALMAERT

K-Nearest Neighbors (KNN)

Advantages:

- **Flexible Decision Boundaries**: KNN can handle complex decision boundaries, making it more suitable for datasets with non-linear relationships.
- **No Need for Model Assumptions**: KNN is a non-parametric method and does not require assumptions about the underlying data distribution.
- **Effective for Multi-Class Problems**: KNN naturally extends to multi-class classification without the need for special modifications like One-vs-All.

Disadvantages:

- **Computationally Expensive**: KNN requires calculating the distance between a test point and every point in the training set, which can be slow for large datasets.
- **Sensitivity to Noisy Data**: KNN is sensitive to noisy data and irrelevant features, which can impact its performance.
- **Choice of KNN**: The performance of KNN heavily depends on the choice of KNN. A small KNN can lead to overfitting, while a large KNN can smooth out the decision boundaries too much.

Conclusion

- **Linear Regression**: While linear regression can be used for binary classification with a threshold-based decision rule, it is not ideal for classification tasks. It has limitations, such as out-of-bound predictions and linear decision boundaries, which make it unsuitable for more complex datasets or multi-class classification.
- **K-Nearest Neighbors (KNN)**: KNN proved to be more effective for multi-class classification. It naturally handles multiple classes and provides more flexible decision boundaries, making it a better choice for the Iris dataset and other similar classification problems.

In summary, while linear regression can work in some classification tasks, KNN is a more robust and effective solution for multi-class classification problems, offering more flexibility and better performance on this dataset.