Machine Learning Lab Report

Title

Visualization of Decision Boundary Using Two Features

Aim of the Experiment

The aim of this experiment is to understand how a machine learning classification algorithm separates data points belonging to different classes by learning a **decision boundary** using only two input features. The experiment also focuses on visualizing this boundary and interpreting the classifier's behavior.

Objectives

- To train a classification model using two features
- To visualize the decision boundary learned by the model
- To understand how feature space is divided into different class regions
- To analyze the performance and behavior of the classifier visually

---

Tools and Technologies Used

- **Programming Language:** Python
- **Platform:** Jupyter Notebook / Google Colab
- **Libraries Used:**
- NumPy (for numerical operations)
- Matplotlib (for plotting graphs and decision boundaries)
- Scikit-learn (for dataset generation and model training)

Theory

Classification

Classification is a supervised machine learning technique where the model learns from labeled data and predicts the class of unseen data points.

Decision Boundary

A **decision boundary** is a line or surface that separates different classes in the feature space. For two features, this boundary can be visualized in a 2D plane.

- Points on one side of the boundary belong to one class
- Points on the other side belong to another class

The shape of the decision boundary depends on: - The type of classifier used - The distribution of the data - The number of features

---

Dataset Description

- The dataset consists of **two numerical features**.
- Each data point belongs to one of **two classes**.
- Data points are represented using different colors for easy visualization.

Methodology / Procedure

1. Import required Python libraries
2. Generate or load a dataset with two features
3. Split the dataset into input features and labels
4. Train a classification model on the dataset
5. Create a mesh grid over the feature space
6. Predict class labels for each point in the mesh grid
7. Plot the decision boundary using contour plots
8. Overlay the original data points on the decision boundary

Observations

- The classifier divides the feature space into two distinct regions.
- Each region corresponds to a predicted class.
- The boundary is **non-linear**, indicating that the classifier can model complex patterns.
- Some points near the boundary are closely mixed, showing regions of uncertainty.

Result

The trained classification model successfully learned a decision boundary that separates the two classes using only two features. The visualization clearly shows how the model classifies different regions of the feature space.

Output Visualization

Decision Boundary Plot

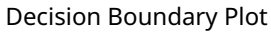

Figure: Decision Boundary learned by the classifier using two features. The background color indicates predicted class regions, while the dots represent actual data points.

Inference

- Decision boundary visualization helps in understanding model behavior.
- Non-linear boundaries suggest the use of advanced classifiers.
- Points close to the boundary are more prone to misclassification.
- Using only two features makes interpretation simple and intuitive.

Applications

- Pattern recognition
- Medical diagnosis
- Spam detection
- Image and signal classification

Advantages

- Easy to interpret for two features
- Helps debug and understand model decisions
- Useful for teaching and learning machine learning concepts

Limitations

- Visualization is not possible for more than three features
- Performance may degrade with noisy data
- Overfitting may occur with complex boundaries

Conclusion

This experiment demonstrates how a machine learning classifier learns and visualizes a decision boundary using two features. Such visualizations are extremely useful for understanding the internal working of classification models and their decision-making process.

References

- Scikit-learn Documentation
- Introduction to Machine Learning by Tom Mitchell
- Lecture Notes on Supervised Learning