HILCOE

Artificial Intelligence (CS488)

GROUP ASSIGNMENT 1

Title: Solve Machine Learning Problems

GROUP MEMBERS

1. LEUL YARED	ID-YN8503
2. MERYEM ABDELLA	ID-BN7468
3. MICHAEL DEMISSEW	ID-CB2419
4. MICHAEL GETU	ID-ML2612
5. MICHAEL WAGAYE	ID-YG6812

SECTION: DRB2102-B
SUBMITTED TO: Mr. Fantahun
SUBMITTING DATE: 17/08/2024

Acknowledgment

We would like to thank our instructor, Mr. Fantahun, for his incredible and interactive teaching methods. We would also like to thank him for giving us this assignment, where we learned several critical concepts related to machine learning, including:

- Understanding the impact of tree depth on overfitting and underfitting in decision trees.
- Exploring how the choice of k and distance metric affects the performance of the K-Nearest Neighbors (KNN) classifier.
- Gaining insight into the workings and limitations of the perceptron learning algorithm.
- Evaluating the strengths and weaknesses of various classifiers and how they perform on the same dataset.
- Understanding the mechanics of k-means clustering and how to assess clustering performance.
- Comparing the differences between k-means and k-medoids clustering, and evaluating their performance on the same dataset.

This assignment has made us greatly realize all these critical topics, through an opportunity to use these concepts in practice.

- Group members

Introduction

An algorithm refers to a well-defined procedure or set of rules to solve a particular problem.

- **Learning Algorithm:** is an algorithm which, over time, performs better after experiencing or training with data. It is used to establish patterns from data and make predictions or decisions on new inputs.
  It can be broadly categorized into supervised, unsupervised, semi-supervised, and reinforcement learning based on the nature of the input data and learning process.

  - *Supervised Learning:* model gets trained on a labeled dataset wherein each example provided for training is associated with an output label. The model learns a mapping from inputs to the desired outputs.
  - *Unsupervised learning:* model gets trained on data without explicit labels. The aim is to find underlying structure in a dataset; often this is done through clustering or association.

Overfitting: An error occurring during modeling when a learning algorithm fits noise in a training dataset rather than underlying patterns in data; generally, it generalizes poorly to new data.
Underfitting: A situation wherein a learning algorithm fails to capture the underlying pattern in a data set, usually due to a too-simple model. It results in poor performance on both the training and test data.

Decision Tree

A decision tree is one of the supervised learning algorithms used in both classification and regression tasks. It works by modeling decisions and their possible results in a tree-like structure, where each internal node represents an attribute test, every branch is the result of the test performed, and every leaf node class label or output.

K-Nearest Neighbors (KNN)

K-Nearest Neighbors is among the most simplistic, non-parametric, supervised learning algorithms applied to classification and regression. Here, the algorithm assigns the majority class for the data point among its k nearest neighbors in the feature space, where k is user-defined.

Euclidean Distance is the straight-line distance between two points in a Euclidean space, calculated using the Pythagorean theorem. It's the most direct path between two points.

Manhattan Distance (or Taxicab Distance) is the distance between two points measured along axes at right angles. It's like how a taxi would drive in a city grid, only moving horizontally or vertically.

Perceptron Learning Algorithm

Perceptron is the simplest neural network, linear binary classifier, and hence forms the basis for all other neural networks. It is an example of a supervised learning algorithm, in that it learns by updating its weights iteratively based on misclassification of training samples

Clustering

Clustering is another unsupervised learning technique used to partition a set of points into clusters so that points in one cluster are more similar to each other than to points in another cluster.

- *K-Means Clustering:* perhaps the most popular partition-based clustering algorithm, which has the aim of dividing a dataset into k clusters, where every data point belongs to the cluster having the nearest mean (centroid).
- *K-Medoids Clustering:* is a variation of k-means clustering that is more resistant to noise and outliers. Instead of taking the mean to represent each cluster, k-medoids use actual data points, called medoids, as the center of clusters.

Performance Metrics

Performance metrics are measurements used to evaluate the effectiveness of a machine learning model, particularly in classification tasks.

Accuracy: ratio of correctly predicted instances to total number of instances.

Accuracy = (True Positives + True Negatives) / Total Instances

It gives an overall measure of how often the model is correct, but it can be misleading if the data is imbalanced.

Precision: ratio of correctly predicted positive instances to total predicted positive instances.

Precision = (True Positives / (True Positives + False Positives)

It answers the question *"Of all the positive predictions, how many were actually correct?"*

Recall (Sensitivity or True Positive Rate): ratio of correctly predicted positive instances to all instances that were actually positive.

Recall = True Positives / (True Positives + False Negatives)

It answers the question *"Of all the actual positives, how many were correctly predicted?"*

F1-Score: The harmonic mean of Precision and Recall, providing a balance between the two.

F1-Score = 2 × ((Precision × Recall) / (Precision + Recall))

It’s useful when you need a balance between Precision and Recall, especially with imbalanced datasets.

Confusion Matrix: A table that shows how well a classification model performs by comparing predicted vs. actual values, summarizing true positives, true negatives, false positives, and false negatives.

Silhouette Score: A measure for evaluating clustering quality, indicating how well-separated and cohesive the clusters are. Scores range from -1 (bad clustering) to 1 (good clustering).

This report provides an in-depth exploration of the topics discussed above, accompanied by a practical implementation using a real dataset.

Table of Content

Problems

Problem #1: Decision Tree Depth Optimization

Task: Train a decision tree classifier on the Iris dataset and optimize its performance by adjusting the maximum depth of the tree.

Instructions:

1. Split the dataset into training and testing sets.
2. Train a decision tree with various maximum depths.
3. Evaluate the performance of each model using accuracy, precision, recall, and F1-score.
4. Visualize the decision boundaries of the best-performing model.
5. Plot the performance metrics against the tree depth to analyze the impact.

Problem #2: K-Nearest Neighbors Hyperparameter Tuning

Task: Implement a k-nearest neighbors (KNN) classifier on the Iris dataset and optimize the number of neighbors (k) and distance metrics (e.g. Euclidean, Manhattan).

Instructions:

1. Split the dataset into training and testing sets.
2. Train multiple KNN classifiers with different values of k and distance metrics.
3. Evaluate the models using accuracy, precision, recall, and F1-score.
4. Visualize the decision boundaries for different k values.
5. Plot performance metrics against different k values and distance metrics.

Problem #3: Perceptron Learning Algorithm

Task: Train a perceptron classifier on the Iris dataset and analyze its performance.

Instructions:

1. Split the dataset into training and testing sets.
2. Train a perceptron classifier on the training set.
3. Evaluate the model using accuracy, precision, recall, and F1-score.
4. Visualize the decision boundaries of the perceptron.
5. Plot the convergence of the perceptron learning algorithm over iterations.

Problem #4: Comparing Decision Tree, KNN, and Perceptron

Task: Compare the performance of decision tree, KNN, and perceptron classifiers on the Iris dataset.

Instructions:

1. Split the dataset into training and testing sets.
2. Train a decision tree, KNN, & perceptron classifier on training set.
3. Evaluate each model using accuracy, precision, recall, and F1-score.
4. Visualize the decision boundaries for all three classifiers.
5. Plot the performance metrics for each classifier.

Problem #5: K-Means Clustering and Visualization
Task: Apply k-means clustering to the Iris dataset and visualize the results.
Instructions:
1. Normalize the Iris dataset.
2. Apply k-means clustering with k=3.
3. Visualize the cluster assignments and the centroids in 2D and 3D plots.
4. Compare the cluster assignments with the actual class labels.
5. Plot the silhouette scores to evaluate the quality of the clusters.

Problem #6: K-Medoids Clustering and Comparison with K-Means
Task: Apply k-medoids clustering to the Iris dataset and compare it with k-means clustering.
Instructions:
1. Normalize the Iris dataset.
2. Apply k-medoids clustering with k=3.
3. Visualize cluster assignments and the medoids in 2D and 3D plots.
4. Compare the cluster assignments with the actual class labels.
5. Compare the performance of k-means and k-medoids using silhouette scores and other clustering metrics.

https://tinyurl.com/AI-Problems6