Task: Implementing Machine Learning Models from Scratch in Python

Description:

Your task is to implement various machine-learning models from scratch using Python. The goal of this task is to test your mathematical understanding of the algorithms and your coding skills. You will be working on two datasets provided in the following files:

- data/ds1_train.csv
- data/ds1_test.csv
- data/ds2_train.csv
- data/ds2_test.csv

Link to Dataset:
https://drive.google.com/drive/folders/1uTIVgqxMDpbuY935zBHE2eRyuxwAXNOO?usp=sharing

Dataset Description:

Each dataset contains multiple examples, with each example represented by a set of features and a corresponding label. The dataset files are in CSV format, where each row represents an example, and the last column represents the label. The features can be of any data type (numeric, categorical, etc.).

Choose at least two different machine learning models from the following options:

Option 1: Logistic Regression
Option 2: Naive Bayes
Option 3: K Means Clustering
Option 4: Gaussian discriminant analysis (GDA)
Option 5: Neural Networks

For each chosen model, perform the following sub-tasks:

Subtask 1: Mathematical Understanding (20 points)
Provide a detailed mathematical explanation of the chosen model. Describe the underlying principles, assumptions, and equations involved. Explain how the model learns from the data and makes predictions.

Subtask 2: Training and Prediction (30 points - Coding)
Implement the chosen model from scratch in Python. You should write your own functions to handle training and prediction steps. Use the provided training dataset to train the model and

the test dataset for evaluation. Print the accuracy of the model on both the training and test datasets.

Note: Use of object-oriented programming is preferred, but not mandatory.

Subtask 3: Hyperparameter Tuning (20 points - Coding)

Select at least one hyperparameter of the chosen model that affects its performance. Implement a method to tune this hyperparameter. This can be done using techniques like grid search or random search. Use suitable evaluation metrics to assess the performance of different hyperparameter values. Print the best hyperparameter value and the corresponding evaluation metric score.

Subtask 4: Comparison with Scikit-Learn (30 points - Coding)

Utilize the scikit-learn library to train and evaluate the chosen model on the same datasets. Compare the performance of your implemented model with the scikit-learn version. Print the accuracy of both models on the training and test datasets.

Note:

- Ensure that you provide comments and explanations in your code to demonstrate your understanding of the algorithms.
- You are allowed to use basic libraries for data manipulation and visualization, but the core logic of the machine-learning models should be implemented from scratch.
- Organize your code into separate files for each subtask (e.g.,logistic_regression.py) to maintain modularity.
- Use appropriate evaluation metrics for classification or regression tasks (e.g., accuracy, precision, recall, F1-score, mean squared error, etc.) depending on the problem type.
- You may choose additional machine learning models not listed above, but make sure they are significantly different from each other.
- Ensure that you submit the necessary files and documents as instructed in the task description.

Resources link:
https://docs.google.com/document/d/11fmdBz7NYf9OUDOwtdG7lzTPQ5O6aLfNYqyxDMV-zf4/edit?usp=sharing