CS60050_Machine Learning_Programming Assignment_3

Total Marks: 100

Part A: Support Vector Machines (SVMs) and Kernel Methods - HIGGS Dataset (50 Marks)

Problem Statement:

You are tasked with building a Support Vector Machine (SVM) classifier to predict whether a particle collision event is classified as a signal (Higgs boson) or background. The dataset is large-scale and high-dimensional, requiring efficient data handling, advanced feature selection, and model tuning.

Dataset:

- **Dataset Name:** HIGGS Dataset
- **Download Link:** [HIGGS Dataset (UCI)]
- **Features:** 28 physics-derived features from particle collision events
- **Target:** Binary classification (Signal vs. Background)

Tasks:

1. Data Preprocessing and Exploration (5 Marks)

- **Exploratory Data Analysis (EDA):** Analyze the dataset, visualize feature distributions, and identify outliers or anomalies.
- **Data Normalization/Standardization:** Apply normalization or standardization to the features for better model performance.
- : **Feature Engineering (2 Marks)**
    - Perform feature engineering (e.g., polynomial features, interaction terms, or transformations) to create new features that might improve model performance.
- : **Feature Selection (2 Marks)**
    - Use methods like Recursive Feature Elimination (RFE) or SelectKBest to identify the most important features for classification, reducing dimensionality.

2. Linear SVM Implementation (10 Marks)

- Implement an SVM with a linear kernel and evaluate the model using cross-validation.

- Report key classification metrics: accuracy, precision, recall, F1-score, and AUC (Area Under the ROC Curve).
- **Scalability and Efficiency (3 Marks)**
    - Discuss and implement strategies to handle the large-scale dataset efficiently (e.g., using Stochastic Gradient Descent or mini-batch learning for SVM).

3. SVM with Polynomial, RBF, and Custom Kernels (15 Marks)

- Implement SVMs with the following kernels:
    - **Polynomial Kernel:** Experiment with degrees (2, 3, 4) and compare the results.
    - **RBF Kernel:** Tune the gamma parameter and observe the effect on performance.
    - **Custom Kernel:** Implement and evaluate at least one custom kernel (e.g., a sigmoid kernel or a hybrid kernel combining RBF and linear).
- Tune the regularization parameter **C** for each kernel using Grid Search or Random Search.
- Compare the performance of each kernel based on classification metrics (accuracy, precision, recall, F1-score, AUC) and computational complexity.

Time Complexity Analysis (3 Marks)

- Evaluate and report the computational cost (time complexity) of each kernel during training and prediction.

4. Hyperparameter Tuning (10 Marks)

- Perform hyperparameter tuning for the chosen kernel to optimize performance.
- Use advanced methods such as **Bayesian Optimization** or **Random Search** for tuning.
- Report the optimal values of the regularization parameter **C** and other kernel-specific parameters (degree for polynomial, gamma for RBF, etc.).

Hyperparameter Sensitivity Analysis (3 Marks)

- Analyze the sensitivity of the SVM performance to different hyperparameters (e.g., changes in C, gamma, or kernel degree), and visualize the results using heatmaps or line plots.

5. Analysis and Report (10 Marks)

- Summarize the results from all kernel methods and hyperparameter variations.
- Compare the performance of each kernel and provide insights on which one is most suitable for the HIGGS dataset based on classification metrics and computational efficiency.
- **Explainability and Interpretability (3 Marks)**

- Use tools like SHAP (SHapley Additive exPlanations) or LIME (Local Interpretable Model-Agnostic Explanations) to explain the model's predictions and assess the importance of the most influential features.

Grading Rubric (Out of 50 Marks):

1. **Code**: Submit well-documented Python code (preferably as PartA_your_name.ipynb) with comments explaining each step.

- **Data Preprocessing and Exploration:** 7 Marks (including feature engineering and selection)
- **Linear SVM Implementation:** 10 Marks (including scalability discussion)
- **SVM with Polynomial, RBF, and Custom Kernels:** 15 Marks (including time complexity analysis)
- **Hyperparameter Tuning:** 10 Marks (including hyperparameter sensitivity analysis)
- **Analysis and Report:** 10 Marks (including explainability and interpretability)

Part B: K-Means Clustering - Anuran Calls Dataset (MFCCs) (50 Marks)

Problem Statement:

You are provided with a dataset of frog species based on their sound frequencies (MFCCs). Your task is to apply advanced clustering techniques, starting with K-Means, to group the frogs into clusters based on their acoustic features and explore clustering performance using additional evaluation methods.

Dataset:

Dataset Name: Anuran Calls Dataset (MFCCs)
Download Link: [Anuran Calls Dataset (Kaggle)]
Features: 22 MFCC coefficients for frog calls

Tasks:

1. Data Preprocessing and Exploration (7 Marks)

- **Exploratory Data Analysis (EDA):** Analyze the dataset by checking for missing values, feature distributions, and outliers.
- **Data Scaling:** Apply feature scaling using normalization or standardization.
- **Feature Engineering:** Try to derive new features from the existing MFCCs (e.g., polynomial features or interaction terms) to potentially improve clustering performance.

Feature Correlation Analysis (2 Marks)

- Investigate correlations between features and remove highly correlated features to avoid redundancy and improve clustering results.

2. K-Means Clustering (15 Marks)

- **Elbow Method:** Implement the Elbow Method to determine the optimal number of clusters.
- **Silhouette Score Evaluation:** After finding the optimal number of clusters, evaluate the clustering quality using the silhouette score.
- **Cluster Implementation:** Implement K-Means clustering based on the optimal number of clusters.

Cluster Initialization (2 Marks)

- Compare different initialization methods for K-Means (e.g., random initialization vs. k-means++).

3.Cluster Visualization (10 Marks)

- **Dimensionality Reduction:** If needed, apply PCA or t-SNE to reduce dimensions for visualization purposes.
- **Cluster Plots:** Visualize the clusters using 2D scatter plots.

Feature Contribution to Clustering (3 Marks)

- Analyze which features (MFCCs) contribute the most to cluster separation and visualize these contributions.

4.Cluster Evaluation Metrics (10 Marks)

Evaluation Using Multiple Metrics

- Calculate additional metrics like the **Davies-Bouldin Index** and **Calinski-Harabasz Index** to assess cluster quality.
- Compare these metrics across different numbers of clusters to validate the Elbow Method and silhouette score results.

5. Comparison with Other Clustering Algorithms (8 Marks)

Algorithm Comparison

- Apply **Agglomerative Hierarchical Clustering** or **DBSCAN** and compare the clustering results with K-Means.
- Analyze the strengths and weaknesses of each algorithm, particularly in the context of this dataset.

6. Analysis and Report (5 Marks)

- Summarize the overall clustering process, including the optimal number of clusters, insights from the visualizations, and an analysis of the chosen evaluation metrics.
- Discuss the limitations of K-Means and other clustering algorithms in terms of their applicability to this dataset.

Submission Requirements & Grading Rubric:

Submission Requirements:

2. **Code**: Submit well-documented Python code (preferably as PartB_your_name.ipynb) with comments explaining each step.
3. **Report:** Provide a detailed report including:

- Visualizations (e.g., Elbow Method, scatter plots, PCA).
- Clustering performance metrics and a comparison between algorithms.
- Key insights and conclusions.

Grading Rubric (Out of 50 Marks):

- **Data Preprocessing and Exploration:** 7 Marks
- **K-Means Clustering:** 15 Marks
- **Cluster Visualization:** 10 Marks
- **Cluster Evaluation Metrics:** 10 Marks
- **Comparison with Other Algorithms:** 8 Marks