CMPT 459 - Fall 2024

Course Project

Objective

In this project, student groups will explore a dataset of their choice, performing multiple data mining tasks including **data preprocessing**, **exploratory data analysis (EDA)**, **clustering**, **outlier detection**, **feature selection**, **classification**, and **model evaluation**. The goal is to develop a comprehensive understanding of the data mining process by applying these techniques and critically analyzing the results. For this project, you are required to use the Python programming language. You may utilize any Python libraries that you find beneficial to achieve the desired outcomes.

Project Requirements

1. **Dataset Selection**
    - Students must select a **real-world dataset** with at least **1000 samples** and **10 or more features** from public repositories like Kaggle, UCI Machine Learning Repository, etc.
    - The dataset should contain a mix of **numerical** and **categorical** features, as well as at least one **target variable** for classification.
    - For the first milestone, provide a 1-2 paragraph **problem definition** that describes the dataset and the problem domain (e.g., customer segmentation, fraud detection, or medical diagnosis).

2. **Data Preprocessing**
    - Handle **missing values** appropriately (e.g., imputation, removal).
    - **Normalize/standardize** numerical features, and perform **encoding** for categorical variables (one-hot encoding or label encoding).
    - Perform **data augmentation** if applicable (e.g., generating synthetic samples for imbalanced datasets).
    - Use **dimensionality reduction** techniques like **PCA** or **t-SNE** if the dataset has high dimensionality.

3. **Exploratory Data Analysis (EDA)**
    - Perform basic EDA to understand the structure and distribution of the dataset.
    - Plot distributions of key features using **histograms**, box plots, etc.

- Visualize relationships between features and identify correlations using **heatmaps**.
- Discuss key insights drawn from EDA and potential challenges with the dataset (e.g., class imbalance, highly correlated features).

4. **Clustering**
   - Apply **at least two different clustering algorithms** (e.g., K-Means, DBSCAN, Hierarchical Clustering).
   - Visualize clustering results using methods like **PCA** or **t-SNE** for dimensionality reduction, followed by 2D or 3D scatter plots.
   - Evaluate the clustering performance using metrics such as **Silhouette Score**, **Calinski-Harabasz Index**, or **Davies-Bouldin Index**.
   - Discuss the appropriateness of the clustering algorithms for your dataset and compare their performances.

5. **Outlier Detection**
   - Perform **outlier detection** using techniques like **Isolation Forest**, **Local Outlier Factor (LOF)**, or **Elliptic Envelope**.
   - Visualize the outliers using plots (e.g., scatter plots with outliers marked).
   - Analyze the outliers: Are they noise, or do they contain important information? Decide whether to keep or remove them for further analysis.

6. **Feature Selection**
   - Apply **feature selection** techniques such as **Recursive Feature Elimination (RFE)**, **Lasso Regression**, or **mutual information** to reduce dimensionality.
   - Discuss the importance of selected features and their impact on the classification task.
   - Evaluate the model with and without feature selection to compare performance and computational efficiency.

7. **Classification**
   - Use **at least three classification algorithms**, such as **Random Forest**, **Support Vector Machines (SVM)**, **k-NN**.
   - Split the dataset into training and testing sets (e.g., 80% training, 20% testing) and perform **cross-validation** (e.g., 5-fold or 10-fold) to evaluate model consistency.
   - Evaluate the models using various metrics, including:
     - **Accuracy**

    - **Precision**
    - **Recall**
    - **F1-score**
    - **AUC-ROC** for binary classification problems.
- Visualize results using a **confusion matrix** and **ROC curves**.

8. **Hyperparameter Tuning**:
    - Perform **hyperparameter tuning** for at least one classifier using **Grid Search** or **Random Search**.
    - Compare the performance of the model before and after tuning. Discuss the impact of tuning on model performance.

9. **Conclusion**
    - Discuss the insights that you learned about the domain of the dataset (e.g., for a rental dataset it could be people's preference and general taste for renting).
    - Discuss the lessons learned about data mining methodology.

Deliverables (100%)

1. **Project Proposal (10%)**:
    - Dataset description, problem definition: 5%
    - Dataset Selection & Justification: 5%

2. **Final Github Report (40%)**:
    - A detailed, step-by-step explanation of each task, including data preprocessing, EDA, clustering, outlier detection, feature selection, classification, and model interpretation.
    - Visualizations and metric results for all stages of the project.
    - Comparisons of different models and their performances.
    - Discussions of challenges, limitations, and potential future work.
3. **Code Submission (20%)**:
    - Clean, well-documented Python code in a **Github repository**.
    - Code should be modular, with clear sections for each task.
    - The README file in your repository will serve as the primary report. It should clearly explain the results of each task, with references to the relevant sections of the code.
4. **Presentation (30%)**:

- A poster presentation summarizing the project, dataset, key methods, results, and **insights**.
- Include visualizations of the clustering, outlier detection, classification.