Datamining Project

1. Project Overview

This project aims to apply various data mining algorithms to a dataset containing 2000 records and 10 columns. The primary objectives are to:

- **Apply Preprocessing Techniques**: Clean and prepare the data for analysis.
- **Feature Selection**: Identify the most relevant features for each algorithm.
- **Algorithm Implementation**: Implement and evaluate the performance of Apriori, Naïve Bayes, ID3, and K-Means.
- **Evaluation**: Evaluate the effectiveness and accuracy of the algorithms through appropriate metrics.

2. Dataset Requirements

- **Dataset Size**: 2000 records and 10 columns.
- **Type of Data**: The dataset should have a mixture of numerical and categorical data.
  - At least **two categorical features** for association rule mining with Apriori.
  - At least **one continuous feature** for Naïve Bayes and K-Means clustering.
- **Data Format**: The dataset should be in CSV or Excel format.
- **Dataset Examples**:
  - **Market Basket Data** (for Apriori).
  - **Customer Data** with demographic information (for Naïve Bayes and ID3).
  - **Multivariate Data** (for K-Means clustering).

3. Preprocessing Requirements

The dataset will undergo preprocessing to handle missing values, outliers, and noise.

- **Handling Missing Values**:
  - If any records have missing values, decide on an imputation strategy (mean , most mention , imputation, deletion, or other).
- **Normalization/Standardization**:
  - For algorithms that require numerical data (like K-Means and Naïve Bayes), ensure that all numeric features are normalized or standardized to a common scale.
- **Encoding Categorical Data**:
  - For categorical features, apply one-hot encoding or label encoding as necessary, depending on the algorithm.

- **Outlier Detection**:
    - Identify and handle outliers in the dataset, especially for clustering algorithms like K-Means.

4. Feature Selection

Feature selection aims to reduce the dimensionality of the dataset by selecting only the most relevant features for each algorithm.

5. Algorithm Requirements

The following algorithms will be applied to the dataset:

a) Apriori Algorithm (Association Rule Mining)

- **Objective**: Discover frequent itemsets and generate association rules.
- **Parameters**:
    - **Support Threshold**: Choose a support threshold that makes sense for the dataset.
    - **Confidence Threshold**: Set a minimum confidence threshold to filter meaningful rules.
    - **Lift**: Compute lift as an additional evaluation metric for the rules.
- **Implementation Details**:
    - Use libraries like mlxtend in Python for easy implementation of the Apriori algorithm.
    - Visualize the association rules (e.g., with network graphs or rule plots).

b) Naïve Bayes (Classification)

- **Objective**: Build a probabilistic model to classify data into predefined classes.
- **Assumptions**:
    - The Naïve Bayes classifier assumes feature independence, which will be checked via correlation analysis.
- **Implementation Details**:
    - Split the dataset into training and testing sets (e.g., 70/30 split).
    - Evaluate performance using accuracy, precision, recall, and F1-score.

c) ID3 Algorithm (Decision Trees)

- **Objective**: Create decision trees based on information gain.
- **Parameters**:
    - **Tree Depth**: Limit the depth of the tree to avoid overfitting.
    - **Pruning**: Implement post-pruning or pre-pruning techniques to reduce tree size.

- **Implementation Details**:
  - Use entropy and information gain to build the decision tree.
  - Visualize the tree structure to understand the classification logic.
  - Evaluate accuracy using cross-validation or a holdout dataset.

d) K-Means Algorithm (Clustering)

- **Objective**: Partition the data into clusters based on similarity.
- **Parameters**:
  - **Number of Clusters (K)**: Use the elbow method or silhouette score to determine the optimal K value.
  - **Initialization**: Use the k-means++ initialization to improve convergence speed and cluster quality.
- **Implementation Details**:
  - Standardize the data before applying K-Means.
  - Visualize the clusters and their centroids.
  - Evaluate clustering quality using metrics like Silhouette Score and Inertia.

6. Model Evaluation

Each model needs to be evaluated on a set of performance metrics:

a) Classification (Naïve Bayes and ID3)

- **Accuracy**: Percentage of correct predictions.
- **Precision, Recall, F1-Score**: These metrics will be used for evaluating the classification models.
- **Confusion Matrix**: To understand misclassifications.
- **Cross-Validation**: Perform k-fold cross-validation to assess model stability.

b) Association Rules (Apriori)

- **Support, Confidence, and Lift**: These metrics will assess the quality of the generated rules.
- **Visualization**: Use visualization tools like graph plots or heatmaps to understand rule relationships.

c) Clustering (K-Means)

- **Silhouette Score**: Measures how well each point is clustered.
- **Inertia**: Measures the sum of squared distances from points to their assigned cluster center.
- **Cluster Visualization**: Use PCA or t-SNE for 2D or 3D visualizations of clusters.

7. Documentation Requirements(important)

Provide clear documentation for all previous requitements and print it as hard copy for:

- **Data Preprocessing Steps**: A detailed report on how the dataset was cleaned and preprocessed.
- **Algorithm Implementation**: A step-by-step guide to the algorithms used, including any parameter tuning.
- **Evaluation Metrics**: Detailed explanation of how each model was evaluated.
- **Results and Insights**: Visualizations, performance scores, and any key insights gained from applying the algorithms.

8. Technology Stack

- **Programming Language**: Python
- **Libraries**:
  - **Data Manipulation**: Pandas, NumPy.
  - **Visualization**: Matplotlib, Seaborn, Plotly.
  - **Machine Learning/Mining**: Scikit-learn, mlxtend (for Apriori), and any relevant libraries for decision trees and clustering.

9. Additional Considerations

- **Scalability**: Ensure that the algorithms are scalable for larger datasets.
- **Reproducibility**: Provide a well-documented and reproducible environment (e.g., using Jupyter Notebooks, Weka..ect).

All Best

Eng.Ibrahim Altharhi