Python and WEKA project

Project Overview

This project involves collecting data via web scraping, preprocessing it, and applying various data mining techniques in both **Python** and **WEKA**. The goal is to compare the performance of different algorithms under varying preprocessing and dimensionality reduction approaches.

Python Tasks

1. Web Scraping

- **Objective**: Scrape the provided website to collect a dataset of **500+ instances**.
- **Implementation**:
  - Use libraries like BeautifulSoup or Scrapy to extract structured data.
  - Store the collected data in a structured format (e.g., CSV or Pandas DataFrame).

2. Data Preprocessing

- **Objective**: Clean and prepare the dataset for analysis.
- **Steps**:
  - **Handle missing data**: Impute or remove null values.
  - **Feature rescaling**: Normalize/standardize numerical features.

- **Outlier treatment**: Detect and manage anomalies using IQR or Z-score.
- **Categorical encoding**: Convert text-based features into numerical representations.

3. Model Training & Comparison

- **Objective**: Apply and evaluate **at least three** machine learning algorithms.
- **Approach**:
  - Train models such as **Random Forest, SVM, Logistic Regression, or XGBoost**.
  - Compare performance using metrics like **accuracy, precision, recall, and F1-score**.

4. Feature Selection via Correlation Matrix

- **Objective**: Reduce dimensionality by eliminating highly correlated features.
- **Steps**:
  - Compute a correlation matrix and remove redundant features.
  - Re-train and evaluate models to assess performance changes.

5. Dimensionality Reduction via PCA

- **Objective**: Further reduce features using Principal Component Analysis (PCA).
- **Steps**:
  - Apply PCA and select the optimal number of components.
  - Compare model performance before and after PCA.

6. Comparative Analysis

- **Objective**: Summarize findings from different preprocessing and modeling approaches.
- **Key Questions**:
  - Which algorithm performed best?
  - Did feature selection (correlation/PCA) improve results?

WEKA Tasks

7. Preprocessing with WEKA Filters

- **Objective**: Apply different preprocessing techniques in WEKA.
- **Filters to Test**:
  - **Normalization/Standardization** (e.g., Normalize, Standardize).
  - **Missing value handling** (e.g., ReplaceMissingValues).
  - **Feature selection** (e.g., CorrelationAttributeEval).

8. Model Training & Hyperparameter Tuning

- **Objective**: Test multiple classifiers in WEKA and optimize their parameters.
- **Approach**:
  - Run algorithms such as **J48 (Decision Tree), Naive Bayes, Random Forest, and SMO (SVM)**.
  - Adjust hyperparameters (e.g., tree depth, kernel type) for better performance.

9. Results Conclusion

- **Objective**: Compare WEKA's results with Python's findings.
- **Key Insights**:
  - Which preprocessing method worked best?
  - Did WEKA or Python yield better model performance?

Video Recording Guidelines

- **Duration**: **10-15 minutes** (concise and structured).
- **Content**:
  - **Python Section**: Walk through code, explain preprocessing, and model comparisons.
  - **WEKA Section**: Demonstrate filter applications, classifier runs, and result analysis.
- **Focus**: Clarity, logical flow, and visual demonstration of key steps.

Grading Criteria

1. **Code Quality**
   - Well-structured, modular code with clear comments and markdown explanations.
   - Proper use of functions and libraries.
2. **Methodological Soundness**
   - Justification for preprocessing and algorithm choices.

- Correct application of dimensionality reduction techniques.

3. **Comparative Analysis**
   - Clear comparison of model performances under different conditions.

Deliverables

1. **Python Notebook** (.ipynb) with:
   - Clean, executable code.
   - Explanations for each major step.
2. **WEKA Results (PDF)**
   - Screenshots of key outputs, model evaluations, and settings.
3. **Recorded Video**
   - Professional, well-paced walkthrough of the project.

Final Notes

- **Best Practices**:
  - Use cross-validation to ensure reliable model evaluation.
  - Document any challenges faced and how they were resolved.
- **Innovation Bonus**:
  - Experiment with advanced techniques (e.g., ensemble methods, neural networks) if time permits.