Project Title

Glass Classification Using Machine Learning

Project Overview

This project involves developing a glass classification system that can predict the type of glass based on its chemical composition. The project aims to assist in forensic science, especially in identifying glass types from crime scenes. Students will use a machine learning model trained on the UCI Glass Identification dataset to achieve this task and will build an interactive user interface to enhance usability.

Objective

The main objective of this project is to create a system that can classify different types of glass based on its chemical features. Students will build a complete pipeline, including data preprocessing, model training, and user interaction through a UI that allows users to input the chemical composition and receive a classification result. It is upto the student to either do complete classification or binary classification (Window glass/Non Window glass).

Tools and Libraries

- Python
- Scikit-learn (for model building and training)
- Pandas and NumPy (for data handling)
- Matplotlib or Seaborn (for data visualization)
- tkinter or Flask (for building the user interface)

Dataset

- **Dataset**: Glass Classification Dataset
- **Description**: The dataset contains 214 instances with 10 different attributes representing the chemical composition of glass. The target variable represents the type of glass, including categories such as building windows, vehicle windows, and containers.

Classes

The dataset includes the following types of glass:

- Building windows float processed
- Building windows non-float processed
- Vehicle windows float processed
- Containers
- Tableware
- Headlamps

•

Project Components

1. **Data Preprocessing**:
    - Load the glass dataset from Kaggle.
    - Handle missing values (if any), normalize features, and split the dataset into training and test sets.
2. **Classification**:
    - Train a classification model using algorithms such as K-Nearest Neighbors (KNN), Support Vector Machine (SVM), or a Neural Network.
    - Evaluate the model's performance using metrics like accuracy, precision, recall, and F1-score.
    - Perform hyperparameter tuning to improve model performance.
3. **User Interaction**:
    - Develop a user-friendly interface using tkinter or Flask.
    - Allow users to input chemical compositions and display the classification result along with confidence scores.
4. **Documentation for Project**:
    - Include detailed documentation explaining the project overview, methodology, results, and future work.

UML: class diagram
build a workflow out of it
separate methods, public, priv

Submission Guidelines

1. **Executable Code**:
    - Submit the entire project as executable code. no notebooks,
2. **Documentation**:
    - Provide a comprehensive documentation of the project
3. **Code Structure**:
    - Ensure the code is well-structured, with proper comments and modular functions.
4. **GUI**:
    - The GUI should be fully functional and demonstrate all key features.

Abstract
Methodology
metrics

ex knn why chose k
nn
kernel trick... why

confusion matrix:
ex: accuracy 70%,
-> 30% only comes from 1 class -> look into that specific