Objective: To implement 2 of the 3 machine learning classification algorithms discussed in class from scratch using either Python or C++ and applying object-oriented programming principles. Students will develop a software package that includes data handling, model training, prediction, and evaluation, without the use of high-level libraries like Pandas in Python or sophisticated libraries in C++.

Goals:
Understand the principles behind Naive Bayes, KNN, and SVM classification.
Practice object-oriented programming and software development skills.
Learn to handle file operations and data processing in Python or C++.

Requirements:
Programming: Basic knowledge of either Python or C++, including classes, functions, and file handling.
Statistics: Understanding of basic probability and statistics as applied to Naive Bayes, KNN, SVM.

Assignment Tasks :

Choose 2 algorithms from the following:
-   Naive Bayes, KNN, SVM
-
For each choice, do the following (let's consider the Naive Bayes algorithm, for example):

Task 1: Data Handler
Implement a class DataHandler:

Methods:
read_csv(filepath): Read a CSV file and store data.
train_test_split(test_size): Split data into training and testing sets.
separate_features_labels(target_column): Separate features from labels.

Task 2: Naive Bayes Classifier
Implement a class NaiveBayesClassifier:

Methods:
__init__: Initialize parameters and structures for the algorithm.
fit(X_train, y_train): Fit the model to the training data.
predict(X_test): Predict labels for the test set.
predict_single(input_features): Predict the label for a single set of input features.
accuracy(y_test, y_pred): Compute the accuracy of the predictions.
classification_report(y_test, y_pred): Generate a report including precision, recall, and F1-score.

Task 4: Main Execution Script
Create a main execution script main.[py/cpp]:
Use the DataHandler to load data and prepare datasets.
Instantiate NaiveBayesClassifier, train it with the training data, and evaluate its performance.
Implement a simple user interface in the console for non-technical users to input data and receive predictions.

Deliverables:
Source code for the project in either Python or C++.

A 2 page report documenting your design decisions, challenges faced, and how you overcame them. Our paper for Project #2 should be single-spaced with a 12-point font.

Evaluation Criteria:
Correctness and completeness of the implementation.
Adherence to OOP principles and best practices in chosen language.
Quality of code, including comments and structure.
Effectiveness of the user interface for making predictions.

Extra Credit:
- Implement all 3 of the algorithms in one language.
- Implement 2 of the 3 algorithms in both Python and C++.
- Implement all 3 of the algorithms in both Python and C++.

Resources:
To learn more about the machine learning algorithms:
- https://en.wikipedia.org/wiki/K-nearest_neighbors_algorithm
- https://en.wikipedia.org/wiki/Naive_Bayes_classifier
- https://en.wikipedia.org/wiki/Support_vector_machine

I have provided a template for the implementation of Naive Bayes in Python. Please see blackboard.