Assignment 01:
Machine Learning & NN
(ICT 06135231)

Presented By:
Md. Shahidul Alam
ID: 2023822017, Batch-6th

Tasks on KNN and Naive Bayes

Datasets: Bank_Personal_Loan_Modelling.csv
Ensure proper (handle missing values, encoding, normalization, etc.).
=>Classification → Accuracy, Precision, Recall, F1-score, Confusion Matrix.
=>Regression → MSE, RMSE, R^2, confusion matrix, etc.

Task 01:

1. Implement KNN with scikit-learn.
2. Preprocess data, implement, compute, and use.
3. Train KNN on a dataset with categorical labels. Evaluate using .
4. Experiment with different and compare with another classifier ().
5. Train KNN on a dataset with continuous target values. Compare with using .
6. Experiment with and observe performance changes.
7. Optimize using hyperparameter tuning. Compare configurations and report finding for different k values.

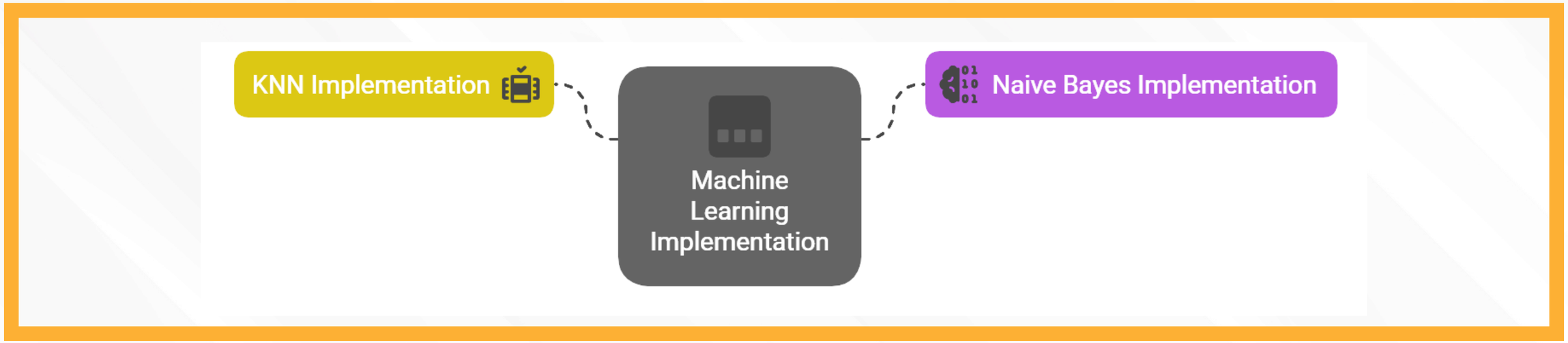
Task 02:

1. Implement the Naive Bayes classifier with scikit-learn. Preprocess data, compute , and classify using .
2. Train Naive Bayes on a dataset with categorical labels. Evaluate using .
3. Compare with another classifier ().
4. Apply Naive Bayes to a dataset with continuous target values. Compare with using .
5. Implement for (e.g., spam detection).

Summary of the Project:

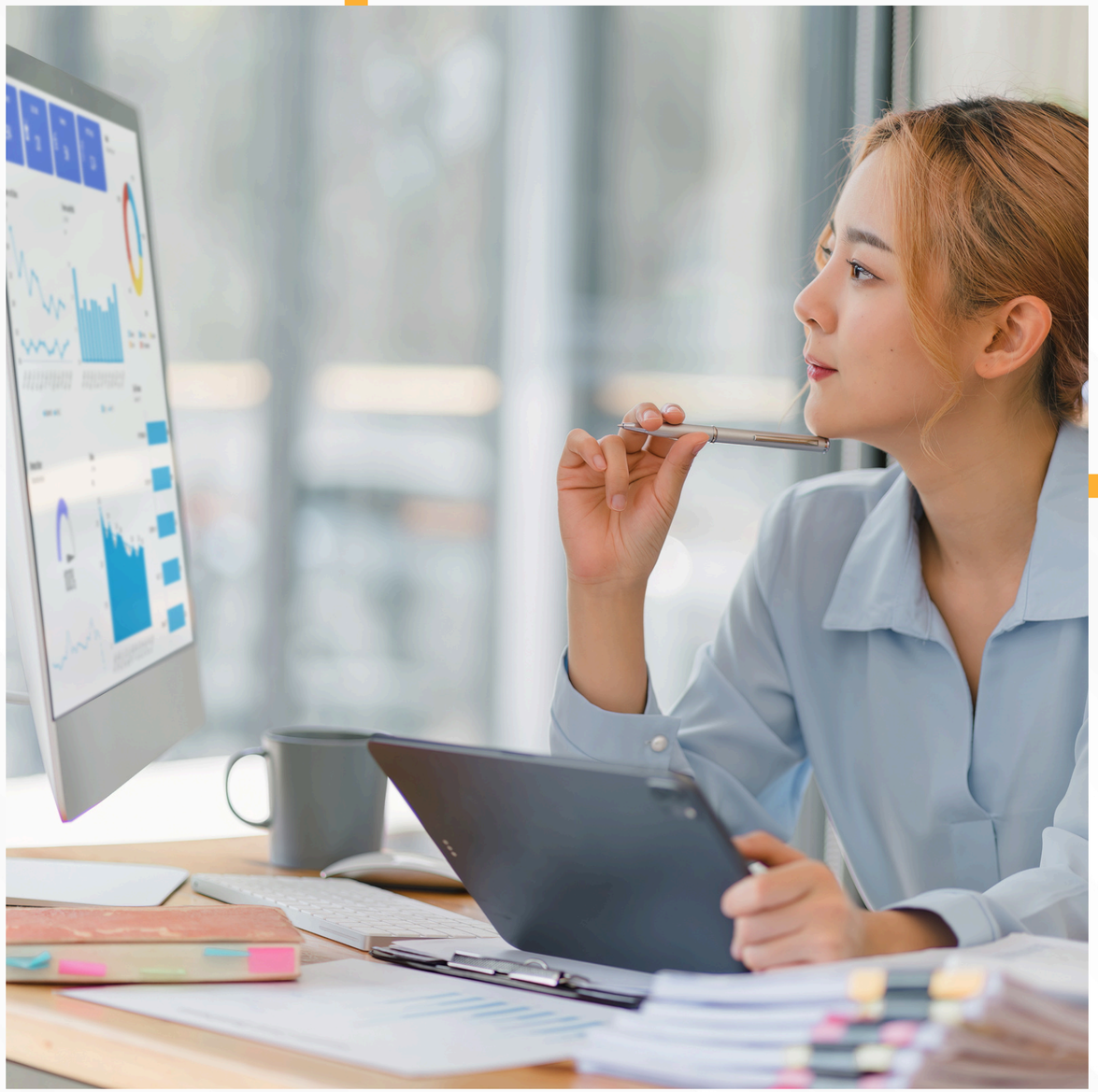
This project focuses on implementing and evaluating two fundamental machine learning algorithms, K-Nearest Neighbors (KNN) and Naive Bayes. The dataset used is the Bank Personal Loan Modelling dataset, which contains information about customers and whether they accepted a personal loan offer. The goal is to classify whether a customer will accept a loan and evaluate the performance of the models using metrics such as accuracy, precision, recall, F1-score and confusion matrices. Additionally, the project explores regression tasks with MSE, RMSE and R^2 for continuous target values.

Project repository



Data Preprocessing

- The dataset was preprocessed to ensure it was suitable for machine learning. Missing values were handled by filling them with the mean of the respective columns. Categorical variables (If any) were encoded using label encoding, and the data was normalized using StandardScaler to ensure all features were on the same scale. The dataset was then split into training and testing sets (70% training, 30% testing) to evaluate the models.



KNN Implementation

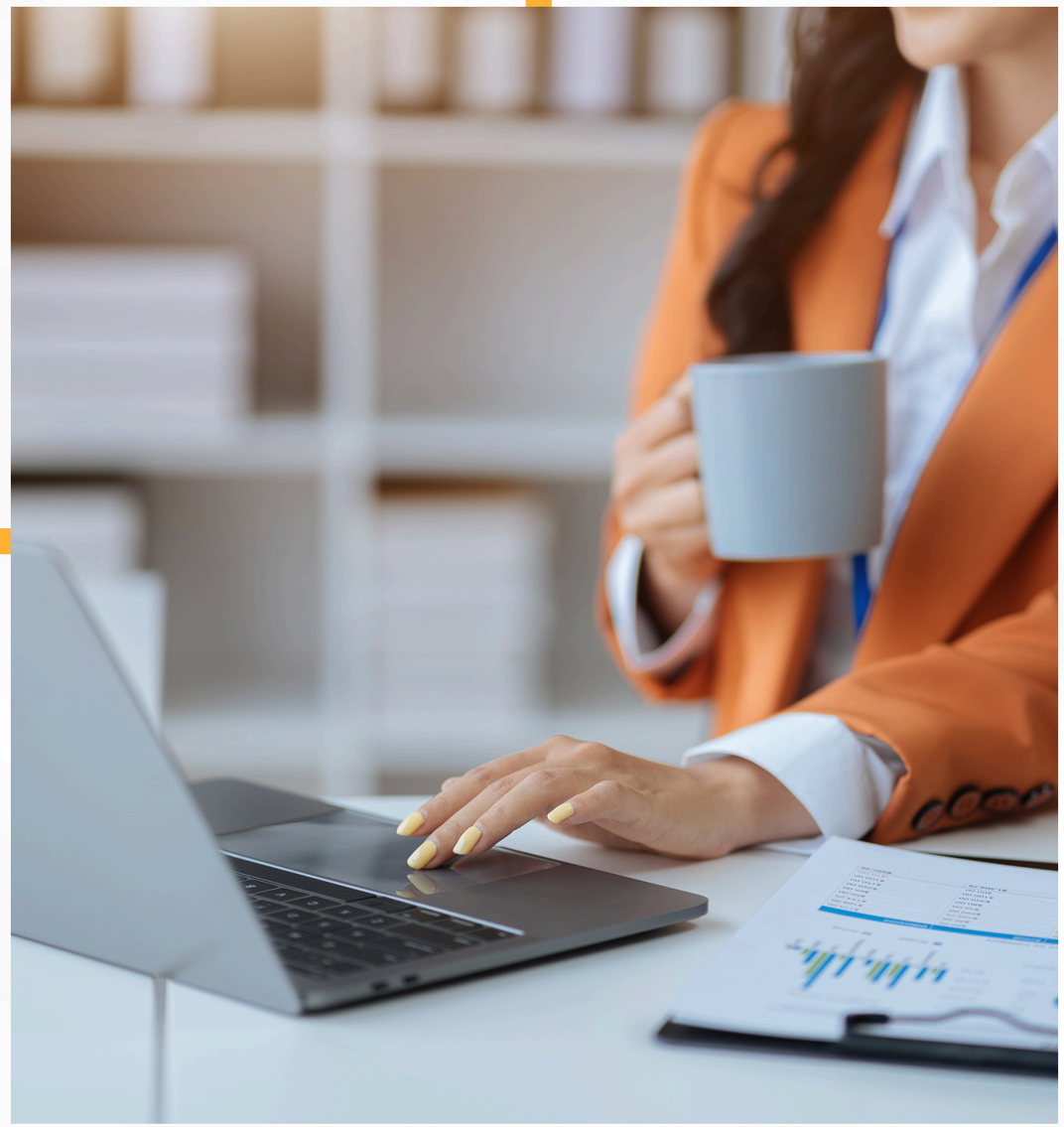
- The KNN algorithm was implemented with scikit-learn. The algorithm predicts the class of a test instance by finding the k-nearest neighbors in the training set and selecting the majority class among them. The model was evaluated using accuracy, precision, recall, F1-score, and a confusion matrix. Different values of k (3, 5, 7, 9) were experimented with to observe their impact on performance. Hyperparameter tuning using GridSearchCV was also performed to find the optimal k value.

Project repository for KNN

Naive Bayes Implementation

- The Naive Bayes classifier was implemented with scikit-learn. The algorithm calculates the prior probabilities of each class and the likelihood of features given the class. Predictions are made using Bayes' theorem, and the model was evaluated using the same metrics as KNN. The Naive Bayes classifier was also applied to a regression task (predicting continuous target values like income) using Gaussian Naive Bayes, and performance was measured using Mean Squared Error (MSE), Root Mean Squared Error (RMSE), and R^2 score.

[Project repository for Naive Bayes]



Comparison and Results

- The performance of KNN and Naive Bayes was compared. KNN achieved higher accuracy and F1-score compared to Naive Bayes for the classification task, but Naive Bayes was computationally more efficient. For regression, Gaussian Naive Bayes provided reasonable predictions, though it is generally less suited for regression tasks compared to classification.

Thank You

That is summary of the project