Adaptive Weighted Voting KNN: A Hybrid Clustering Method for Mixed Data

Marina Lin, Clinton Morimoto

Table of contents

01

INTRODUCTION

Primary Problem & Motivation

Problem

Loan Approval

- Impacts individuals' access to funds and institutions' ability to manage risk

Misjudgments → **Significant financial losses for institutions**

Rejecting creditworthy applicants → **Limit economic growth and aspirations**

Previous Approaches

- Previous works experimented with different distance metrics:
  - Manhattan, Minkowski, Mahalanobis, Cosine, etc.
- Mixed weighting approaches with inversely proportional weights
  - Useful for imbalanced class labels

02

DATASET

Dataset Description, Preprocessing, Train/Test Split

Loan Classification Dataset

- Categorical and Numeric Types, 13 features
- Loan Approval Status; **1 = approved, 0 = rejected**

Table 1: Dataset Features with Description and Type

Column	Description	Type
person_age	Age of the person	Float
person_gender	Gender of the person	Categorical
person_education	Highest education level	Categorical
person_income	Annual income	Float
person_emp_exp	Years of employment experience	Integer
person_home_ownership	Home ownership status (e.g., rent, own, mortgage)	Categorical
loan_amnt	Loan amount requested	Float
loan_intent	Purpose of the loan	Categorical
loan_int_rate	Loan interest rate	Float
loan_percent_income	Loan amount as a percentage of annual income	Float
cb_person_cred_hist_length	Length of credit history in years	Float
credit_score	Credit score of the person	Integer
previous_loan_defaults_on_file	Indicator of previous loan defaults	Categorical
loan_status	Loan approval status: 1 = approved; 0 = rejected	Integer

Preprocessing

- 45000 Total Instances
- 10000 Instances that are approved loans, 35000 unapproved

After preprocessing with normalization and type conversion

- 9000 Total Instances
- 2022 Instances that are approved loans, 6978 unapproved

Train/Test Split: 80/20

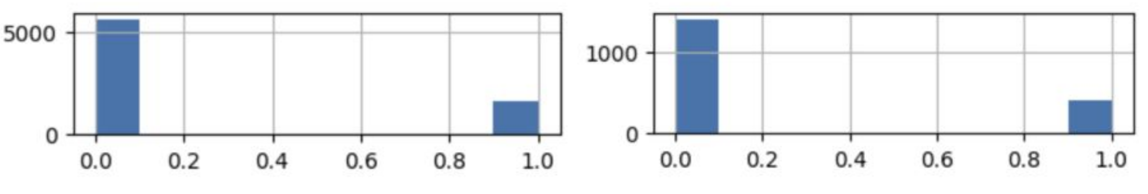


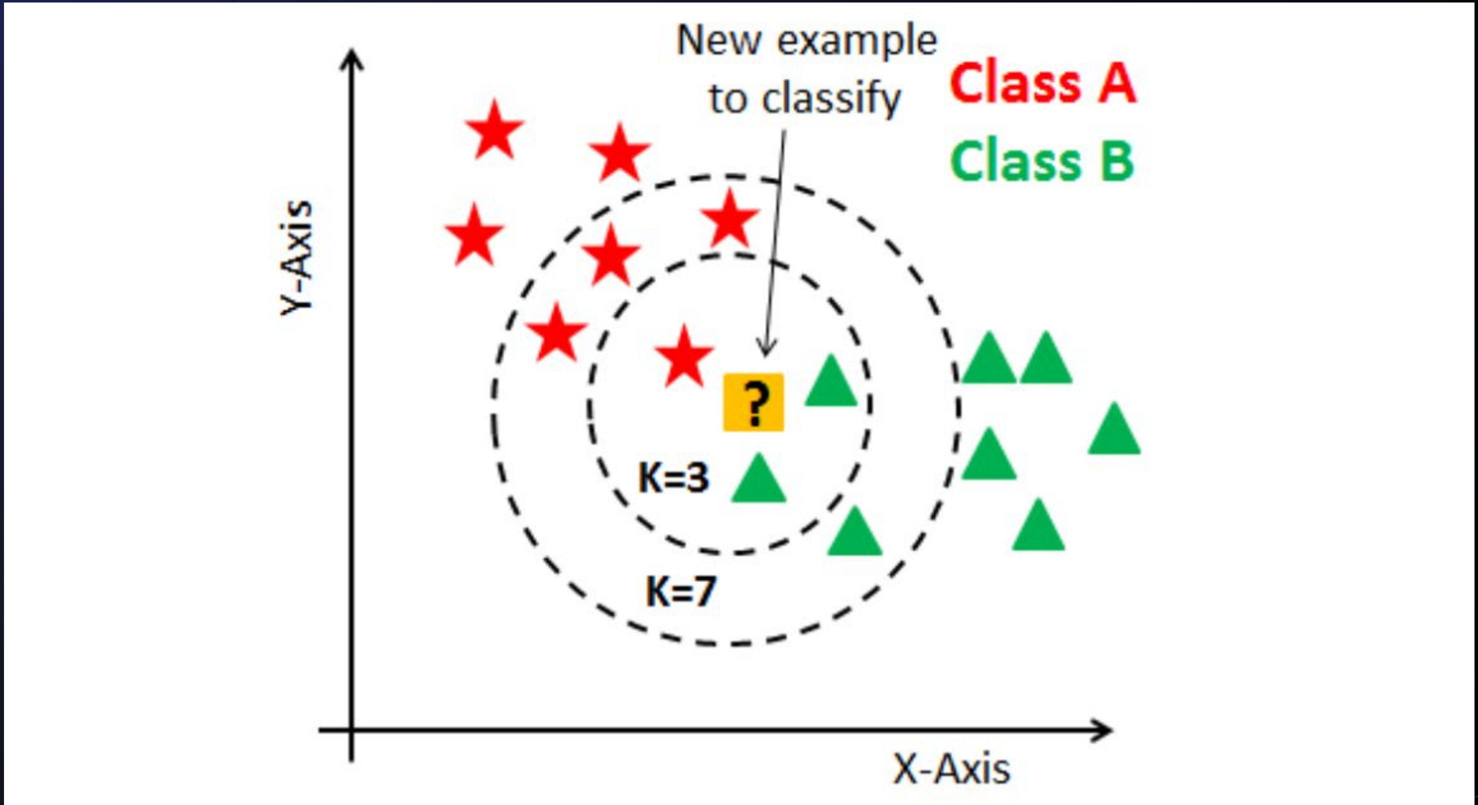
Figure 1: Class Label Distribution for Train (left) and Test (right).

03

METHODOLOGY

KNN, Mixed Data, Our extension

KNN



KNN Algorithm for Mixed Data

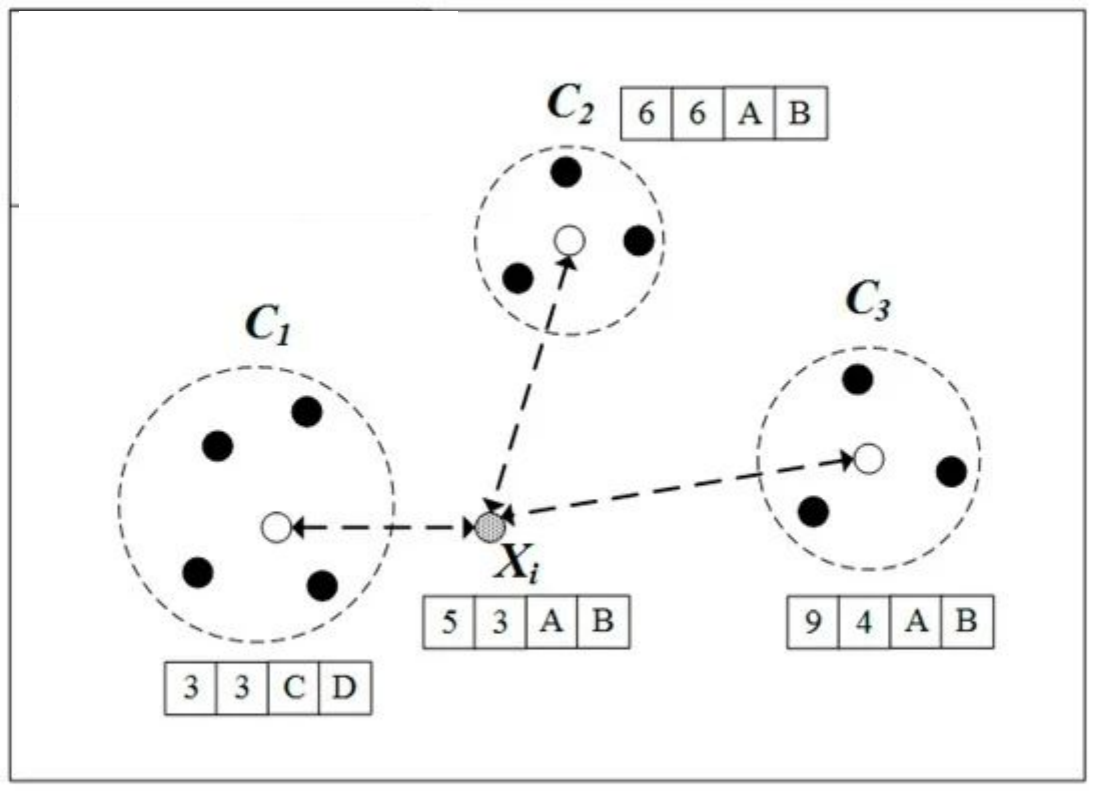
Three classes, $C=\{C_1, C_2, C_3\}$.

Points from each class are:

$C_1=(3,3,C,D)$

$C_2=(6,6,A,B)$

$C_3=(9,4,A,B)$



KNN Algorithm for Mixed Data

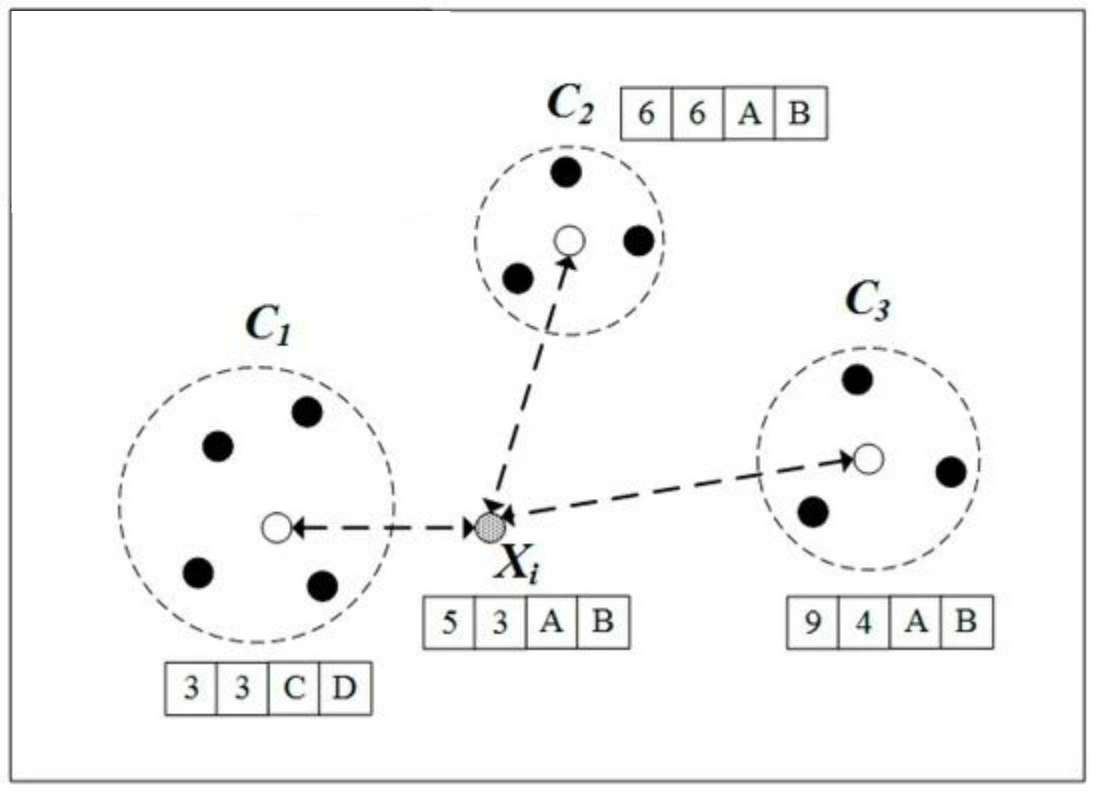
Calculates distance with each point to find the class to which X_i = (5, 3, A, B) is assigned.

The distance about the numerical attribute of X_i and C_1, C_2, C_3 is

(3−5)^2+(3−3)^2=4

(6−5)^2+(6−3)^2=10

(9−5)^2+(4−3)^2=17.



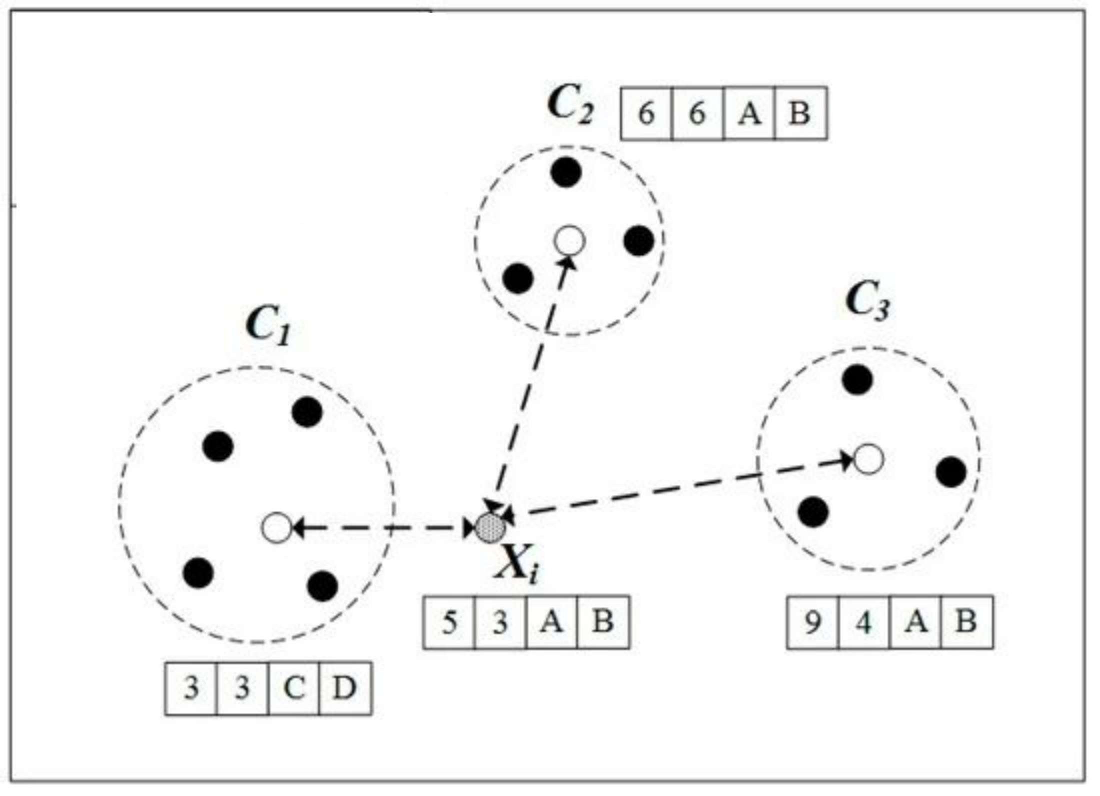
KNN Algorithm for Mixed Data

The distance about the categorical attribute of X_i and C_1, C_2, C_3 is

1 +1=2 ∵C≠A, D≠B

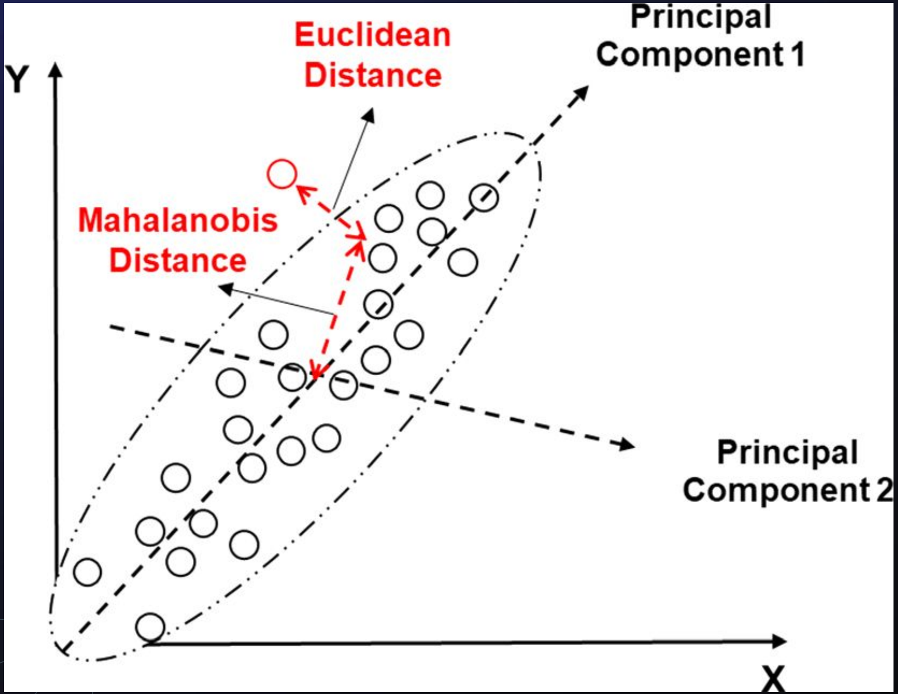
0 +0=0 ∵A=A, B=B,

The total distance of X_i to C_1, C_2, C_3 is 6, 10, and 17, respectively.



Extension

Mahalanobis distance



Weighted Voting

Weight Minority Label Higher

1. Accumulate weights for each class c among the k-nearest neighbors

2. Assign the class with the highest accumulated weight

Pseudocode

Algorithm 1 Weighted Mixed K-Nearest Neighbors (WM-KNN)

Input: Training dataset with numerical and categorical features, test dataset, k neighbors
Output: Predicted labels for all test data points
Step 1: Calculate weights for each class in the training dataset:

- Determine class counts using the training labels.
- Assign a higher weight to the minority class ($w = 2.0$) and default weight to other classes ($w = 1.0$).

Step 2: For each test data point x:

- Compute distances to all training points.
- Identify the k-nearest neighbors based on the computed distances.

Step 3: Perform weighted voting to determine the predicted label:

- Accumulate weights for each class c among the k-nearest neighbors:
- Assign the class with the highest accumulated weight.

Step 4: Repeat Steps 2–3 for all test data points.
Step 5: Evaluate the performance of the algorithm using multiple metrics.

04

RESULTS

Performance of our proposed algorithm.

Performance Metrics

k = 85

- **Accuracy:**

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}}$$

- **Precision:**

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}}$$

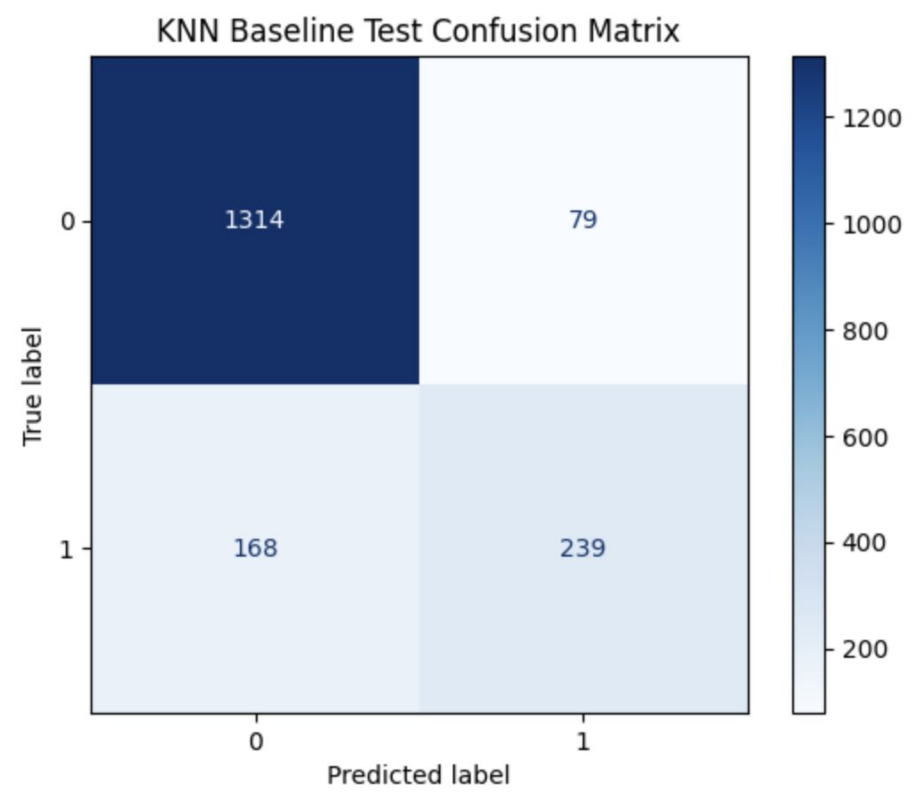
- **Recall:**

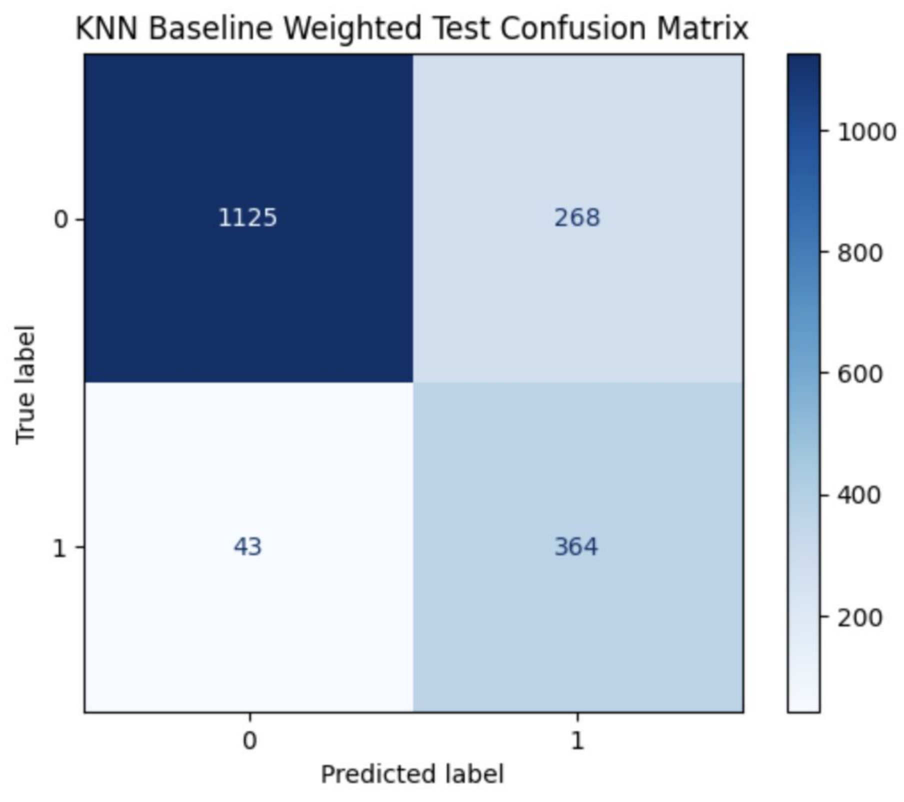
$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}}$$

- **F1-Score:**

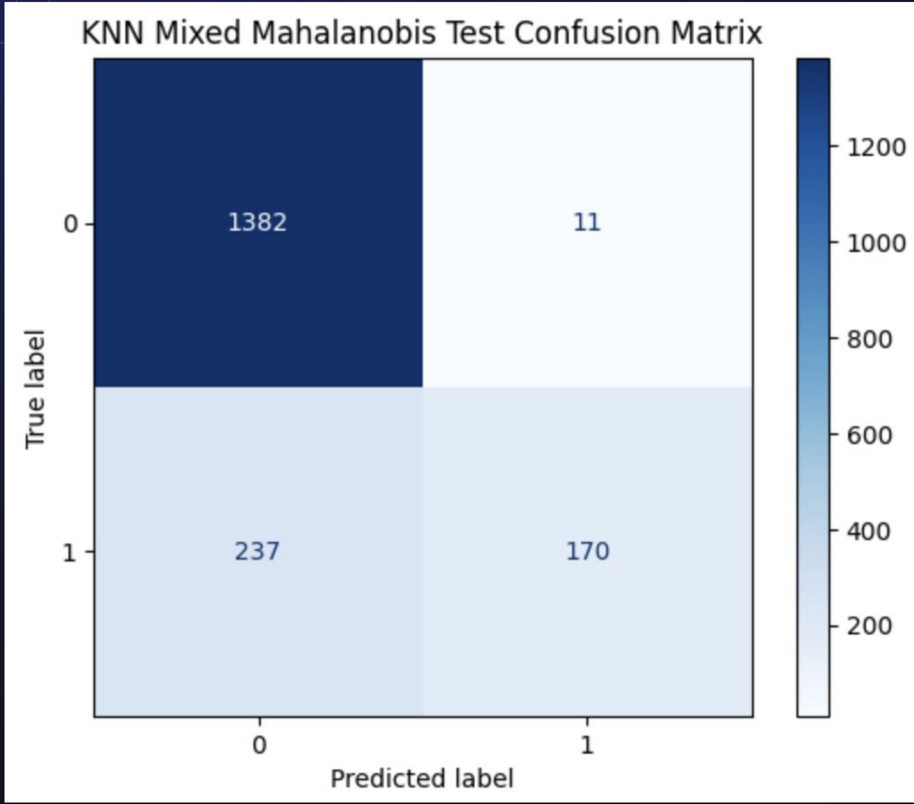
$$\text{F1-Score} = 2 \cdot \frac{\text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}}$$

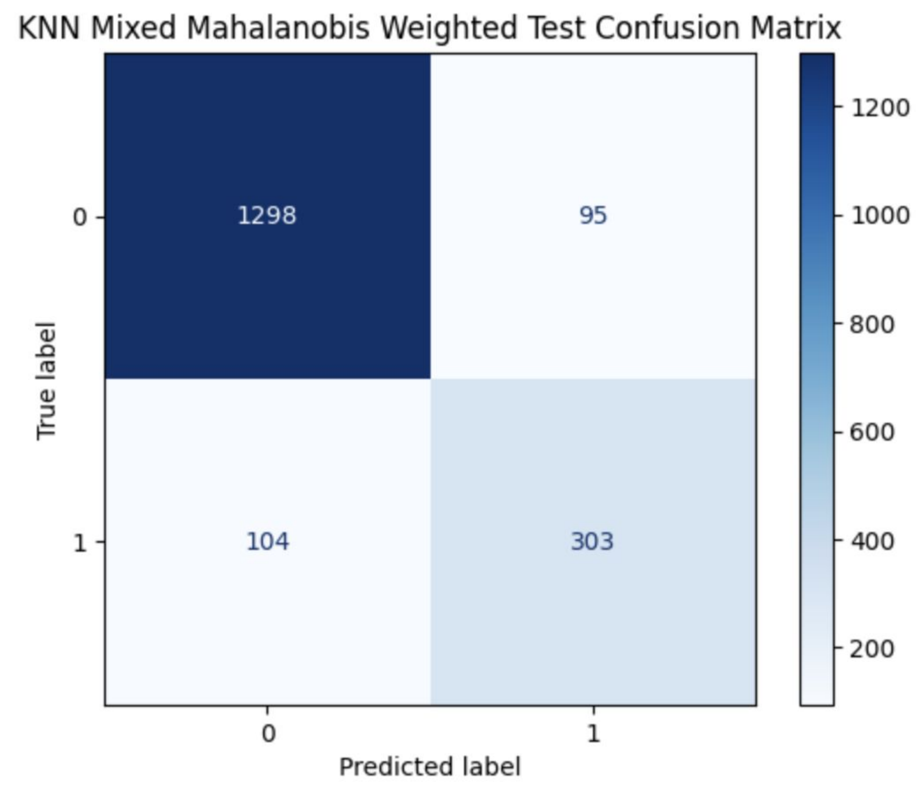
Mixed KNN w/Euclidean





Mixed KNN w/Mahalanobis





Algorithm Comparison

Table 2: Comparison of Different KNN Variants on Testing Data

Metric	Euclidean	Euclidean Weighted	Mahalanobis	Mahalanobis Weighted
Accuracy (Test %)	86.28	82.72	86.22	**88.94**
Precision (Test, 0)	94.33	80.76	99.21	93.18
Precision (Test, 1)	58.72	89.43	41.77	74.44
Recall (Test, 0)	88.66	96.32	85.36	92.58
Recall (Test, 1)	75.16	57.59	93.92	76.13
F1-Score (Test, 0)	91.41	87.32	91.77	**92.88**
F1-Score (Test, 1)	65.93	69.62	57.82	**75.28**

1 = approved, 0 = rejected

Conclusion

- Developed a new Hybrid KNN Clustering Method for Mixed Data

Mixing Two Techniques for improved performance

1. Adaptive Weighted Voting
2. Mahalanobis Distance Metric

Improvement in Accuracy, F1-Score, and achieved good balance for other metrics

Thank you!