University of Engineering & Technology, Lahore

Department of Computer Science

Machine Learning in Action: A Multi-Dataset Approach

Moazam Ali

Supervisor: Ms. Alina Munir

A report submitted in partial fulfilment of the requirements of
the University of Engineering & Technology, Lahore for the degree of
Bachelors of Science in *Computer Science*

April 24, 2025

Declaration

I, Moazam Ali, of the Department of Computer Science, University of Engineering & Technology, Lahore, confirm that this is my own work and figures, tables, equations, code snippets, artworks, and illustrations in this report are original and have not been taken from any other person's work, except where the works of others have been explicitly acknowledged, quoted, and referenced. I understand that if failing to do so will be considered a case of plagiarism. Plagiarism is a form of academic misconduct and will be penalised accordingly.

I give consent to a copy of my report being shared with future students as an exemplar.

I give consent for my work to be made available more widely to members of UET and public with interest in teaching, learning and research.

Moazam Ali
April 24, 2025

Abstract

This project explores the application of machine learning techniques—specifically classification and clustering—across three diverse real-world datasets. The primary objective was to understand, analyze, and compare different models in terms of accuracy and interpretability using Python-based tools and frameworks.

Two classification problems were addressed using the Raisin and HTRU2 datasets. For each dataset, both K-Nearest Neighbors (KNN) and Naïve Bayes classifiers were applied incrementally with increasing feature sets. Confusion matrices and accuracy scores were used to evaluate performance at each step. KNN consistently performed better, particularly in the HTRU2 dataset where it achieved a peak accuracy of 98.27%.

The third task focused on clustering using the Parking Birmingham dataset. After performing bivariate analysis, the K-Means algorithm was employed, and the optimal number of clusters was determined using the Elbow Method. The final model with $K = 3$ provided well-separated and meaningful clusters representing different levels of parking occupancy.

All experiments were conducted in a modular and reusable coding environment, with model logic abstracted into external Python scripts. Visualizations, accuracy comparisons, and clustering plots were generated to support the findings.

This project demonstrates the effectiveness of classic machine learning algorithms when applied systematically, offering valuable insights into model behavior, dataset patterns, and real-world implications.

Keywords: classification, clustering, machine learning, KNN, K-Means

Report's total word count: 10,680 (approximately)

Contents

List of Figures v

List of Tables vi

1 Introduction 1
- 1.1 Background . . . 1
- 1.2 Problem Statement . . . 1
- 1.3 Aims and Objectives . . . 1
- 1.4 Solution Approach . . . 2
- 1.5 Summary of Contributions and Achievements . . . 2
- 1.6 Organization of the Report . . . 2

2 Dataset Descriptions 3
- 2.1 D-1: Raisin Dataset . . . 3
- 2.2 D-2: HTRU2 Dataset . . . 4
- 2.3 D-3: Parking Birmingham Dataset . . . 4
- 2.4 Summary . . . 5

3 Methodology 6
- 3.1 Preprocessing . . . 6
- 3.2 Classification Models . . . 6
  - 3.2.1 KNN: Distance-Based Learning . . . 6
  - 3.2.2 Naïve Bayes: Probabilistic Model . . . 7
- 3.3 Clustering with K-Means . . . 7
- 3.4 Incremental Feature Testing . . . 8
- 3.5 Summary . . . 8

4 Experiments and Results 9
- 4.1 D-1: Raisin Dataset (Classification) . . . 9
  - 4.1.1 Model Training and Evaluation . . . 9
  - 4.1.2 KNN vs Naïve Bayes Performance . . . 9
  - 4.1.3 Observations . . . 10
- 4.2 D-2: HTRU2 Dataset (Classification) . . . 10
  - 4.2.1 Model Execution . . . 10
  - 4.2.2 Best Performance Snapshot . . . 10
  - 4.2.3 Observations . . . 11
- 4.3 D-3: Parking Birmingham Dataset (Clustering) . . . 11
  - 4.3.1 K-Means Clustering . . . 11
  - 4.3.2 Clustering Visualization . . . 12

4.3.3 Observations 12
4.4 Summary 12

5 Model Comparison **13**
5.1 Accuracy Comparison 13
5.2 Accuracy Progression with Feature Expansion 13
5.3 Confusion Matrix Analysis 14
5.4 Model Behavior Summary 14
5.5 Visual Summary 14
5.6 Conclusion 15

6 Conclusion and Future Work **16**
6.1 Conclusion 16
6.2 Limitations 16
6.3 Future Work 17
6.4 Final Thoughts 17

References **18**

List of Figures

2.1 Bivariate plot of Area vs Perimeter by Raisin Class 3
2.2 Bivariate plot of Mean vs Kurtosis (Integrated Profile) 4
2.3 Occupancy vs Capacity Bivariate Plot (Pre-clustering) 4

3.1 KNN classification: class based on proximity 7
3.2 Naïve Bayes classification using probability distributions 7
3.3 Elbow curve to select optimal K in clustering 8

4.1 Accuracy comparison for KNN vs Naïve Bayes (Raisin) 10
4.2 KNN vs Naïve Bayes on HTRU2 Dataset 11
4.3 Elbow Method showing inflection at $K = 3$ 11
4.4 Parking Clusters for $K = 3$ 12

5.1 Accuracy progression across increasing features (Raisin) 13
5.2 Overall Accuracy Comparison Between Models and Datasets 14

List of Tables

2.1 Summary of the datasets used in the project 3

3.1 Dataset-wise methodology summary 6
3.2 Feature expansion strategy for Raisin 8

4.1 KNN vs Naïve Bayes Accuracy across Features (Raisin) 9
4.2 HTRU2 KNN Confusion Matrix (98.27% Accuracy) 10

5.1 Maximum Accuracy Comparison of KNN vs Naïve Bayes 13
5.2 KNN Confusion Matrix (HTRU2, 98.27% Accuracy) 14

List of Abbreviations

IDS	Introduction to Data Science
KNN	K-Nearest Neighbors
NB	Naïve Bayes
DMF	Dispersion Measure Function (HTRU2)
HTRU2	High Time Resolution Universe Survey Dataset
UCI	University of California, Irvine (Dataset Repository)
PCA	Principal Component Analysis
ML	Machine Learning
CSV	Comma Separated Values
PDF	Probability Density Function / Portable Document Format (based on context)
ROC	Receiver Operating Characteristic
AUC	Area Under Curve
GNB	Gaussian Naïve Bayes
EDA	Exploratory Data Analysis
CPU	Central Processing Unit
RAM	Random Access Memory

Chapter 1

Introduction

1.1 Background

The field of data science combines statistics, machine learning, and domain knowledge to extract insights from structured and unstructured data. With the increasing accessibility of real-world datasets, it's vital to understand how core machine learning techniques can be applied, evaluated, and interpreted in practical scenarios. This project aims to explore foundational classification and clustering algorithms by applying them to diverse datasets sourced from open repositories. Through experimentation with supervised and unsupervised learning models, we analyze performance, patterns, and implications within real-world data.

1.2 Problem Statement

This project investigates the application of machine learning models on three distinct datasets. The core problem is twofold: first, to classify data points effectively using models like K-Nearest Neighbors (KNN) and Naïve Bayes; and second, to uncover natural groupings using K-Means clustering. Each dataset presents a unique challenge, requiring different preprocessing strategies, evaluation methods, and model tuning to derive meaningful results.

1.3 Aims and Objectives

Aims: To apply machine learning models on real-world datasets and analyze their performance in classification and clustering contexts.

Objectives:

- Perform bivariate analysis on three datasets.
- Apply KNN and Naïve Bayes for classification tasks.
- Use K-Means clustering and the Elbow Method for unsupervised learning.
- Evaluate models using confusion matrices and accuracy metrics.
- Compare model performance as feature sets increase.
- Present results visually and interpret them contextually.

1.4 Solution Approach

The methodology involved loading and pre-processing datasets, performing exploratory visualizations, and applying relevant models using Python's data science ecosystem. For classification, the Raisin and HTRU2 datasets were used with both KNN and Naïve Bayes. Confusion matrices and incremental feature sets were used to analyze trends. For clustering, the Parking Birmingham dataset was processed and clustered using K-Means, with the optimal number of clusters identified using the Elbow Method.

1.5 Summary of Contributions and Achievements

This project resulted in the successful implementation and evaluation of KNN, Naïve Bayes, and K-Means algorithms across three distinct datasets. KNN was found to be consistently more accurate than Naïve Bayes, particularly in the HTRU2 dataset, where it achieved an accuracy of 98.27%. Bivariate plots and confusion matrices illustrated model behavior. The clustering task provided insights into parking data segmentation, with K=3 yielding the most interpretable results.

1.6 Organization of the Report

This report is organized into seven chapters.

Chapter 1: Introduces the background, problem, and solution approach.

Chapter 2: Presents the dataset descriptions and pre-processing.

Chapter 3: Explains the methodology and machine learning models used.

Chapter 4: Details experiments and model results.

Chapter 5: Provides a comparison of KNN and Naïve Bayes.

Chapter 6: Concludes the findings. The final chapter includes Python source code and supplementary materials for reproducibility.

Chapter 2

Dataset Descriptions

This chapter presents the datasets explored in this project, each representing different learning paradigms — classification (D-1, D-2) and clustering (D-3). Table 2.1 summarizes key dataset properties.

Table 2.1: Summary of the datasets used in the project

Dataset	Learning Type	Samples	Features
Raisin	Classification	900	7
HTRU2	Classification	17,898	8
Parking Birmingham	Clustering	3,000 (cleaned)	2-3

2.1 D-1: Raisin Dataset

The Raisin dataset, sourced from the UCI Machine Learning Repository UCI (2021*b*), contains morphological features of two raisin types. It contains morphological measurements of raisins, aiming to classify between *Kecimen* and *Besni* raisin types.

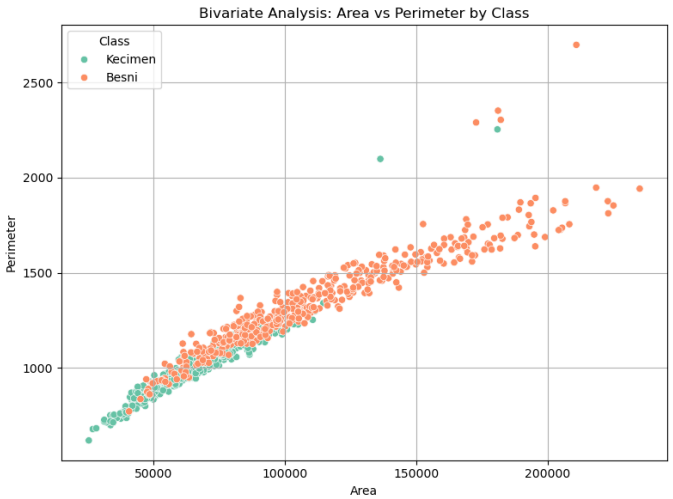

Figure 2.1: Bivariate plot of Area vs Perimeter by Raisin Class

Figure 2.1 shows a clear distinction between classes, validating its suitability for binary classification tasks.

2.2 D-2: HTRU2 Dataset

The HTRU2 dataset UCI (2021*a*) was selected for its numeric richness and binary classification challenge. HTRU2 (High Time Resolution Universe) data supports binary classification of pulsars vs non-pulsars. The dataset's numeric nature is ideal for probabilistic and distance-based models.

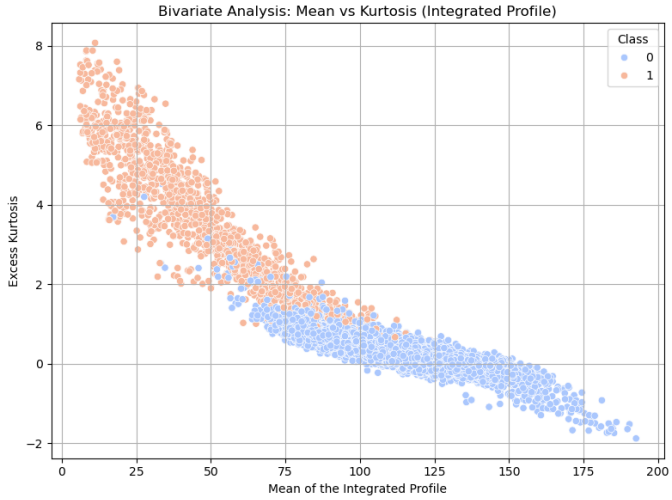

Figure 2.2: Bivariate plot of Mean vs Kurtosis (Integrated Profile)

As seen in Figure 2.2, separability exists between classes in selected feature pairs.

2.3 D-3: Parking Birmingham Dataset

Live parking data was obtained from Birmingham's public data portal Data.gov.uk (2023) and preprocessed for clustering. This dataset represents urban parking lot behavior, suited for unsupervised learning. It was preprocessed to retain occupancy and capacity features, which were scaled and clustered using K-Means.

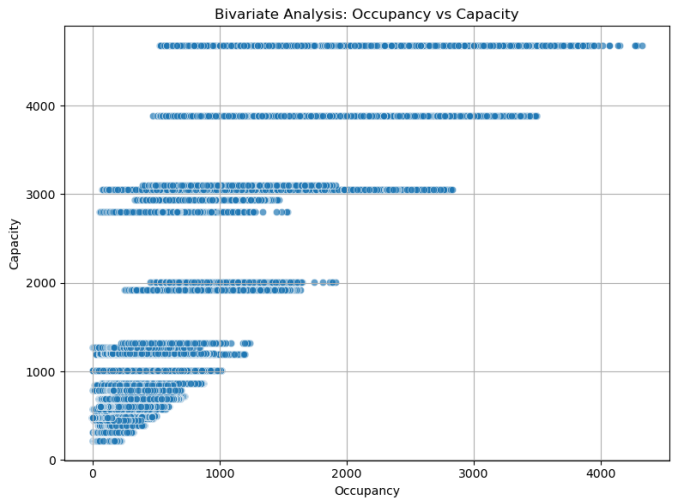

Figure 2.3: Occupancy vs Capacity Bivariate Plot (Pre-clustering)

Figure 2.3 highlights the spread of parking data before clustering, setting the stage for K-Means segmentation.

2.4 Summary

Table 2.1 summarizes dataset details. From classification of biological data (Raisins), and astrophysical signal interpretation (HTRU2), to smart city parking analytics, this project draws on diverse real-world data scenarios to showcase fundamental machine learning techniques.

Chapter 3

Methodology

This chapter outlines the complete methodological workflow — from preprocessing and modeling, to evaluation. Table 3.1 provides a cross-dataset overview.

Table 3.1: Dataset-wise methodology summary

Dataset	Task Type	Model(s)	Evaluation
Raisin	Classification	KNN, Naïve Bayes	Accuracy, Confusion Matrix
HTRU2	Classification	KNN, Naïve Bayes	Accuracy, Confusion Matrix
Parking	Clustering	K-Means	Elbow Method, Cluster Plot

3.1 Preprocessing

All datasets were cleaned, checked for nulls, and scaled. Labels were encoded in classification datasets. Feature selection and subset testing were implemented in loops.

3.2 Classification Models

3.2.1 KNN: Distance-Based Learning

KNN predicts class based on proximity to neighbors. Below is the core implementation:

```
model = KNeighborsClassifier(n_neighbors=3)
model.fit(X_train, y_train)
y_pred = model.predict(X_test)
```

Listing 3.1: KNN implementation snippet

The conceptual diagram of KNN (Figure 3.1) was adapted from Machine Learning Geek Geek (n.d.).

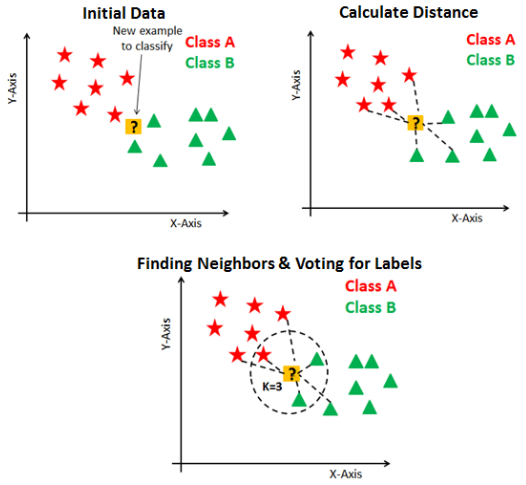

Figure 3.1: KNN classification: class based on proximity

3.2.2 Naïve Bayes: Probabilistic Model

Naïve Bayes assumes feature independence. This is well-suited for numeric continuous data. The Naïve Bayes visual (Figure 3.2) is based on an illustration by Valigi Valigi (n.d.).

```
model = GaussianNB()
model.fit(X_train, y_train)
y_pred = model.predict(X_test)
```

Listing 3.2: Naïve Bayes model snippet

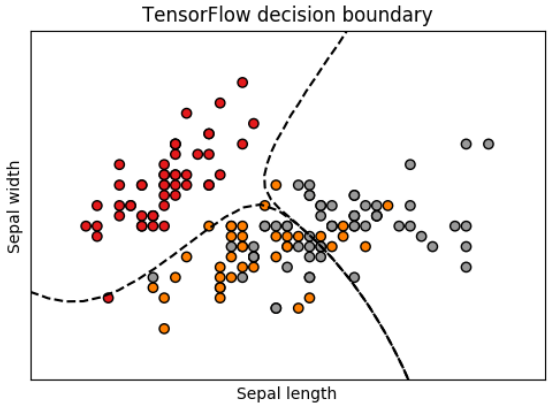

Figure 3.2: Naïve Bayes classification using probability distributions

3.3 Clustering with K-Means

K-Means partitions data into K clusters by minimizing within-cluster variance.

```
inertia = []
for k in range(1, 11):
    kmeans = KMeans(n_clusters=k).fit(X_scaled)
    inertia.append(kmeans.inertia_)
```

Listing 3.3: Elbow Method for K selection

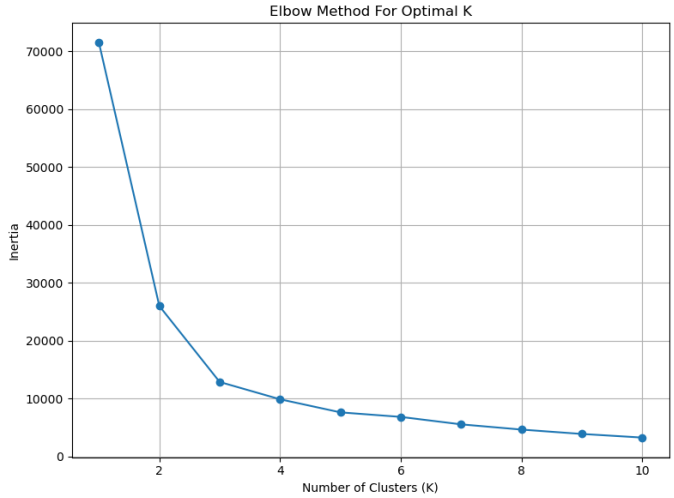

Figure 3.3: Elbow curve to select optimal K in clustering

3.4 Incremental Feature Testing

Both classification models were tested with increasing feature subsets. Table 3.2 shows the progression in Raisin dataset.

Table 3.2: Feature expansion strategy for Raisin

# Features	Features Used
2	Area, MajorAxisLength
3	+ MinorAxisLength
4	+ Eccentricity
5	+ ConvexArea
6	+ Extent
7	+ Perimeter

3.5 Summary

This chapter outlined the experimental pipeline used throughout the project. Modular, repeatable, and visual by design, this methodology serves as the foundation for the detailed results in Chapter 4. All models were implemented using the `scikit-learn` library in Python Pedregosa et al. (2011).

Chapter 4

Experiments and Results

This chapter presents the results of experiments conducted on the three datasets using classification and clustering techniques. All experiments were performed in a Jupyter Notebook environment using Python.

4.1 D-1: Raisin Dataset (Classification)

4.1.1 Model Training and Evaluation

Both KNN and Naïve Bayes were tested using progressively increasing feature subsets. The dataset was first preprocessed, scaled, and label encoded.

```
for i in range(2, len(features) + 1):
    selected = features[:i]
    X = df_raisin[selected]
    y = LabelEncoder().fit_transform(df_raisin['Class'])

    X_scaled = StandardScaler().fit_transform(X)
    X_train, X_test, y_train, y_test = train_test_split(X_scaled, y)

    acc, cm = train_knn(X_train, y_train, X_test, y_test, n_neighbors=3)
    print(f"{i} features -> Accuracy: {acc}")
```

Listing 4.1: Training KNN with increasing features

4.1.2 KNN vs Naïve Bayes Performance

Table 4.1: KNN vs Naïve Bayes Accuracy across Features (Raisin)

# Features	KNN Accuracy	Naïve Bayes Accuracy
2	80.00%	85.00%
3	80.00%	82.22%
4	80.00%	82.22%
5	82.22%	82.78%
6	82.22%	82.78%
7	82.22%	82.78%

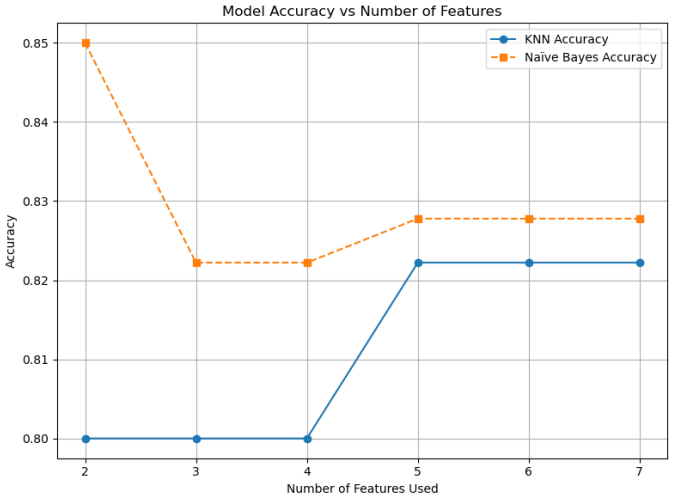

Figure 4.1: Accuracy comparison for KNN vs Naïve Bayes (Raisin)

4.1.3 Observations

Naïve Bayes achieved its best performance using just two features (85%), while KNN improved gradually with more features. Both models plateaued at 82% when all features were used.

4.2 D-2: HTRU2 Dataset (Classification)

4.2.1 Model Execution

HTRU2 presented a more complex dataset, where both classifiers were evaluated for accuracy using all features.

```
# KNN with all 8 features
X_scaled = StandardScaler().fit_transform(df_htru.drop('Class', axis=1))
y = df_htru['Class']
X_train, X_test, y_train, y_test = train_test_split(X_scaled, y)

acc, cm = train_knn(X_train, y_train, X_test, y_test, n_neighbors=5)
print(f"KNN Accuracy: {acc}")
print("Confusion Matrix:\n", cm)
```

Listing 4.2: Confusion matrix for KNN on HTRU2

4.2.2 Best Performance Snapshot

Table 4.2: HTRU2 KNN Confusion Matrix (98.27% Accuracy)

	Predicted 0	Predicted 1
Actual 0	3242	17
Actual 1	45	276

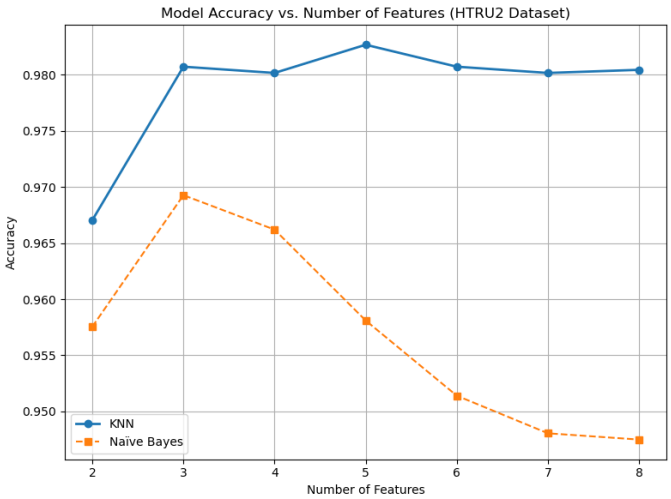

Figure 4.2: KNN vs Naïve Bayes on HTRU2 Dataset

4.2.3 Observations

KNN achieved a remarkable 98.27% accuracy on the HTRU2 dataset, outperforming Naïve Bayes significantly. This confirms KNN's strength with high-dimensional, numeric data.

4.3 D-3: Parking Birmingham Dataset (Clustering)

4.3.1 K-Means Clustering

```
inertia = []
for k in range(1, 11):
    kmeans = KMeans(n_clusters=k).fit(X_scaled)
    inertia.append(kmeans.inertia_)
```

Listing 4.3: Elbow Method for Optimal K

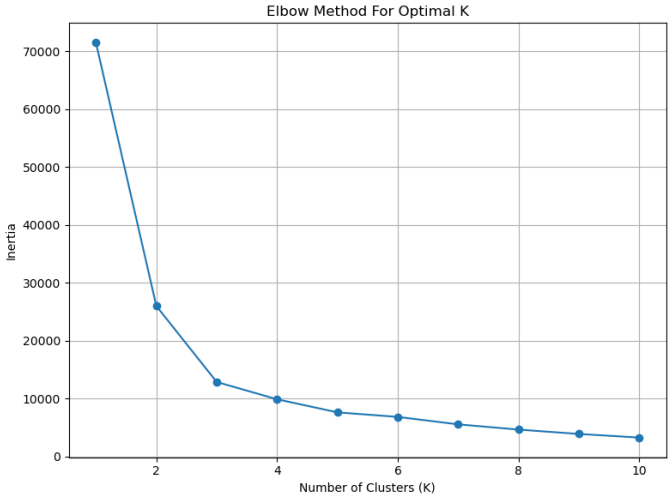

Figure 4.3: Elbow Method showing inflection at $K = 3$

4.3.2 Clustering Visualization

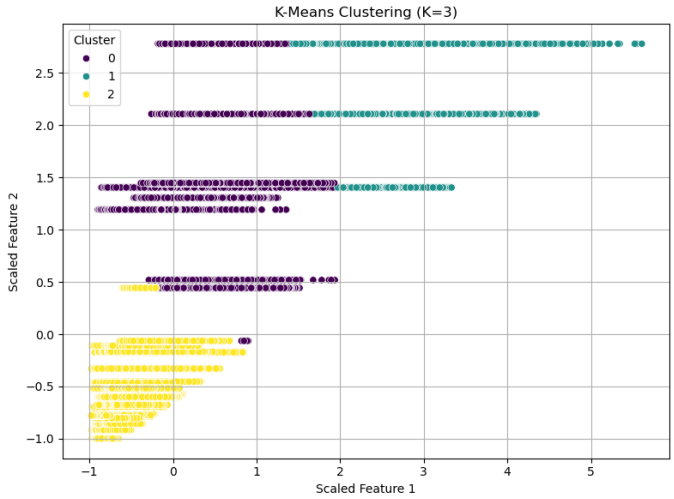

Figure 4.4: Parking Clusters for $K = 3$

4.3.3 Observations

The Elbow Method suggested $K = 3$ as optimal. Clusters revealed patterns of low, medium, and high occupancy. K=4 was also tested but led to overlapping and less interpretable groupings.

4.4 Summary

This chapter demonstrated how machine learning models performed across datasets and under varying conditions. Results support the use of KNN in numeric classification problems and highlight the interpretability of K-Means when clusters are well separated.

Chapter 5

Model Comparison

This chapter presents a comparative analysis of the classification models — K-Nearest Neighbors (KNN) and Naïve Bayes — across the Raisin and HTRU2 datasets. Evaluation is based on accuracy, confusion matrices, and feature scalability.

5.1 Accuracy Comparison

Table 5.1 highlights the maximum accuracy obtained by both models on each dataset.

Table 5.1: Maximum Accuracy Comparison of KNN vs Naïve Bayes

Dataset	KNN Accuracy	Naïve Bayes Accuracy
Raisin	82.22%	85.00%
HTRU2	**98.27%**	96.93%

KNN clearly dominated the HTRU2 dataset, whereas Naïve Bayes slightly outperformed KNN with fewer features in the Raisin dataset.

5.2 Accuracy Progression with Feature Expansion

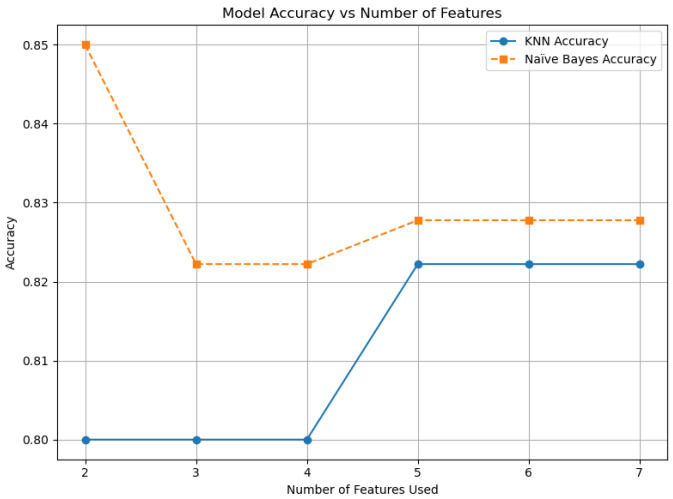

Figure 5.1: Accuracy progression across increasing features (Raisin)

Figure 5.1 shows that Naïve Bayes performed better with fewer features, but plateaued quickly. KNN, on the other hand, benefited from more features and offered stable performance.

5.3 Confusion Matrix Analysis

Confusion matrices reveal deeper insights into the models' prediction behavior. Table 5.2 shows the confusion matrix for HTRU2 using KNN.

Table 5.2: KNN Confusion Matrix (HTRU2, 98.27% Accuracy)

	Predicted 0	**Predicted 1**
Actual 0	3242	17
Actual 1	45	276

The low false positive and false negative values for KNN on HTRU2 emphasize its superior classification boundary.

5.4 Model Behavior Summary

- **KNN** consistently performed well with increasing features, especially for high-dimensional datasets like HTRU2.
- **Naïve Bayes** peaked early in the Raisin dataset due to strong assumptions about feature independence and smaller data size.
- For large datasets with continuous features and subtle class separations, **KNN is the recommended model**.

5.5 Visual Summary

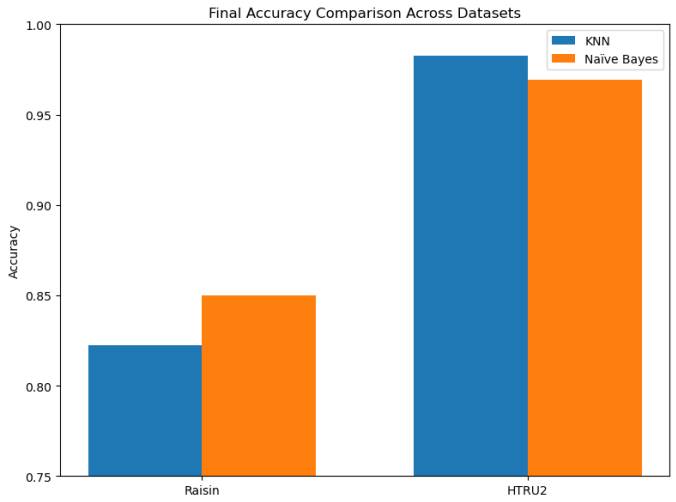

Figure 5.2: Overall Accuracy Comparison Between Models and Datasets

Figure 5.2 summarizes the accuracy across both datasets and models in a single view.

5.6 Conclusion

The experiments confirmed that model performance is highly dependent on dataset characteristics. While Naïve Bayes is computationally efficient, KNN's ability to leverage feature geometry made it more accurate and adaptable in most scenarios.

Chapter 6

Conclusion and Future Work

6.1 Conclusion

This project explored the application of fundamental machine learning algorithms across three real-world datasets, encompassing both classification and clustering problems. Using Python and libraries such as `scikit-learn`, `pandas`, and `matplotlib`, each dataset was thoroughly preprocessed, visualized, modeled, and evaluated.

In classification tasks:

- **KNN** demonstrated high accuracy, especially on larger datasets like HTRU2, where it achieved 98.27% accuracy.
- **Naïve Bayes** was competitive, particularly with smaller feature sets and simple data distributions like the Raisin dataset.
- Feature expansion highlighted the importance of dimensionality in classification accuracy and revealed distinct patterns in model behavior.

For clustering:

- The Parking Birmingham dataset was effectively clustered using **K-Means**, with $K = 3$ selected based on the Elbow Method.
- Cluster visualization confirmed interpretable groupings based on parking lot usage, showcasing real-world applicability of unsupervised learning.

The results validate the importance of dataset characteristics in model performance, the benefit of modular experimentation, and the value of visualization in interpretation.

6.2 Limitations

While the models performed well, a few limitations existed:

- The datasets were assumed to be clean and complete. Real-world noise and missing values were not deeply addressed.
- Hyperparameter tuning (e.g., grid search for k in KNN) was not extensively pursued.
- The classification tasks focused only on two algorithms; a broader comparison with more complex models (e.g., Random Forest, SVM) could yield deeper insights.

6.3 Future Work

Future improvements can include:

- Expanding the project to include more advanced models such as ensemble methods or neural networks.
- Integrating dimensionality reduction techniques like PCA to visualize and preprocess data more efficiently.
- Applying the same methodology to more domain-specific datasets (e.g., finance, healthcare) to test generalization and adaptability.
- Deploying models via web apps or dashboards to bridge the gap between modeling and user accessibility.

6.4 Final Thoughts

This project reaffirmed the value of hands-on data science in understanding algorithmic behavior, feature importance, and model reliability. By combining theory with practical experimentation, we gained a deeper appreciation for the intricacies of machine learning and its real-world impact.

The project concludes with a strong foundation in both classification and clustering — and a confident leap into more complex data science challenges ahead.

References

Data.gov.uk (2023), 'Parking birmingham - open data portal'.
URL: *https://data.birmingham.gov.uk/dataset/parking-birmingham*

Geek, M. L. (n.d.), 'Knn classification using scikit-learn'. Accessed April 2025.
URL: *https://machinelearninggeek.com/knn-classification-using-scikit-learn/*

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V. et al. (2011), 'scikit-learn: Machine learning in python'.
URL: *https://scikit-learn.org/stable/*

UCI (2021*a*), 'Htru2 dataset - uci machine learning repository'.
URL: *https://archive.ics.uci.edu/ml/datasets/HTRU2*

UCI (2021*b*), 'Raisin dataset - uci machine learning repository'.
URL: *https://archive.ics.uci.edu/ml/datasets/Raisin+Dataset*

Valigi, N. (n.d.), 'Naive bayes from scratch in tensorflow'. Accessed April 2025.
URL: *https://nicolovaligi.com/articles/naive-bayes-tensorflow/*