K- Nearest Neighbour (KNN)-Machine Learning Tutorial

Student Name: Mahesh Babu Poka
Student ID: 24070896
Module: Machine Learning & Neural Networks
Tutor: Peter Scicluna
Assignment: Individual ML Tutorial Report
Dataset: WineQT.csv
Submission Date: 11 Dec 2025
GitHub Link: https://github.com/Maheshbabupoka241710/KNN_Tutorial

K- Nearest Neighbour (KNN)-Machine Learning Tutorial

TABLE OF CONTENTS

K- Nearest Neighbour (KNN)-Machine Learning Tutorial

1. Introduction

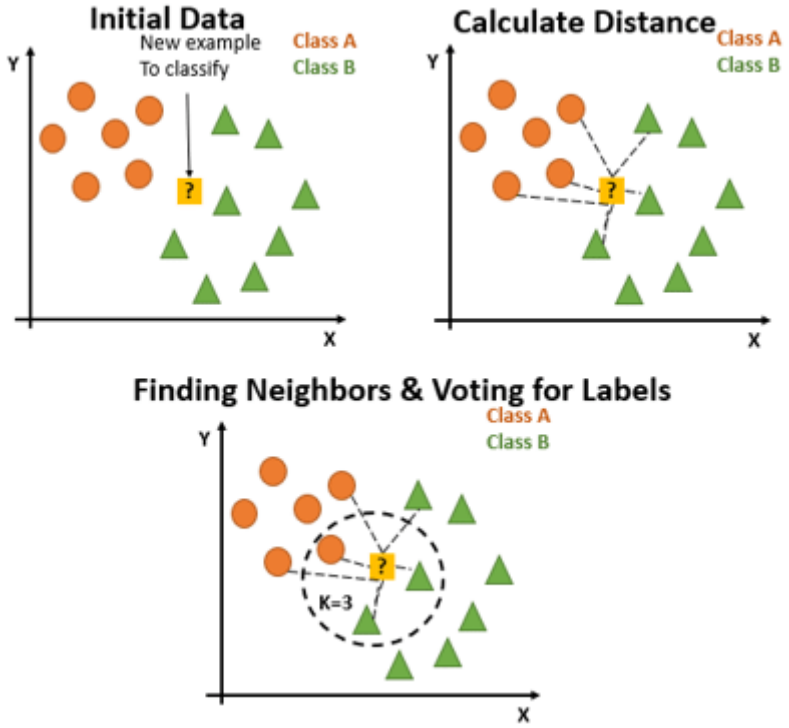
In machine learning, sometimes the simplest ideas turn out to be the most effective—and K-Nearest Neighbors (KNN) is the perfect example of this. KNN does not learn complicated formulas, train internal weights, or build a deep model. Instead, it follows a straightforward rule:

"Look at who is closest and follow the majority."

KNN stores the entire dataset and only makes decisions when a new example needs to be classified. To predict a label, it finds the *k* nearest points and assigns the class that appears most often among them. No training phase, no complex math just pure intuition.

You can think of KNN as asking for advice from the closest examples:
"What are the neighbours around me doing?"
That is the heart of KNN.

Figure 1: KNN Classification Workflow



In this figure, the yellow box represents a new point we want to classify. The orange circles and green triangles represent two classes.

- First, we see the original data.
- Then, distances from the new point to nearby points are calculated.
- Finally, KNN identifies the *k* closest neighbours and predicts the class based on majority voting.

This example shows how the value of *k* can influence predictions. A small *k* is sensitive to noise, while a large *k* may smooth out useful patterns. In this tutorial, we explore how to choose the best *k*, how KNN makes decisions, and how it performs on the Wine Quality dataset using real visualisations and code.

2. Understanding How KNN Works

2.1 Learning from Your Neighbours – The Heart of KNN

At its core, KNN relies on a simple idea:
"To understand something new, compare it with what is closest and most similar."

K- Nearest Neighbour (KNN)-Machine Learning Tutorial

Whenever we classify a new wine sample, KNN:

1. Looks at all stored samples
2. Calculates how close each one is
3. Picks the *k* nearest neighbours
4. Uses majority vote to decide the label

KNN is called a **lazy learner** because it does not train a traditional model. It waits until prediction time, then uses distance and similarity to classify the new point.

The prediction rule is:

$$\hat{y} = mode\{y_i \mid x_i \in N_k(x)\}$$

Where:

- $\hat{y}$: predicted class
- $N_{k(x)}$: the k nearest neighbours
- y_i: labels of those neighbours

This simple voting mechanism is what makes KNN both intuitive and powerful.

2.2 How KNN Measures Closeness – Distance Metrics

Everything in KNN depends on how we define "nearest."
We measure closeness using **distance formulas**.

Euclidean Distance (Straight-line distance)

$$d(x, x') = \sqrt{\sum_{i=1}^{n} (xi - xi')\text{^}2}$$

- Measures straight-line distance
- Works well for continuous, scaled features

Manhattan Distance (based on horizontal and vertical paths):

$$d(x, x') = \sum_{i=1}^{n} |xi - xi'|$$

- Good for high-dimensional or grid-like data
- Less sensitive to outliers

Minkowski Distance (a more general form):

$$d(x, x') = \left(\sum_{i=1}^{n} | xi - xi' |^p \right)^{1/p}$$

- When **p = 1 → Manhattan**
- When **p = 2 → Euclidean**

The choice of distance metric depends on how your features behave.
In this project, all features were scaled so Euclidean and Manhattan distances worked effectively.

2.3 Visual Overview of KNN Behaviour in This Tutorial

Throughout this report, several figures help visualise how KNN interacts with the Wine Quality dataset. Each plot highlights a different part of the model-building process—class balance, feature patterns, clustering, tuning, and performance. These figures make it easier to understand why KNN works the way it does and how the data structure affects its predictions.

The following sections introduce each figure at the appropriate stage of the workflow.

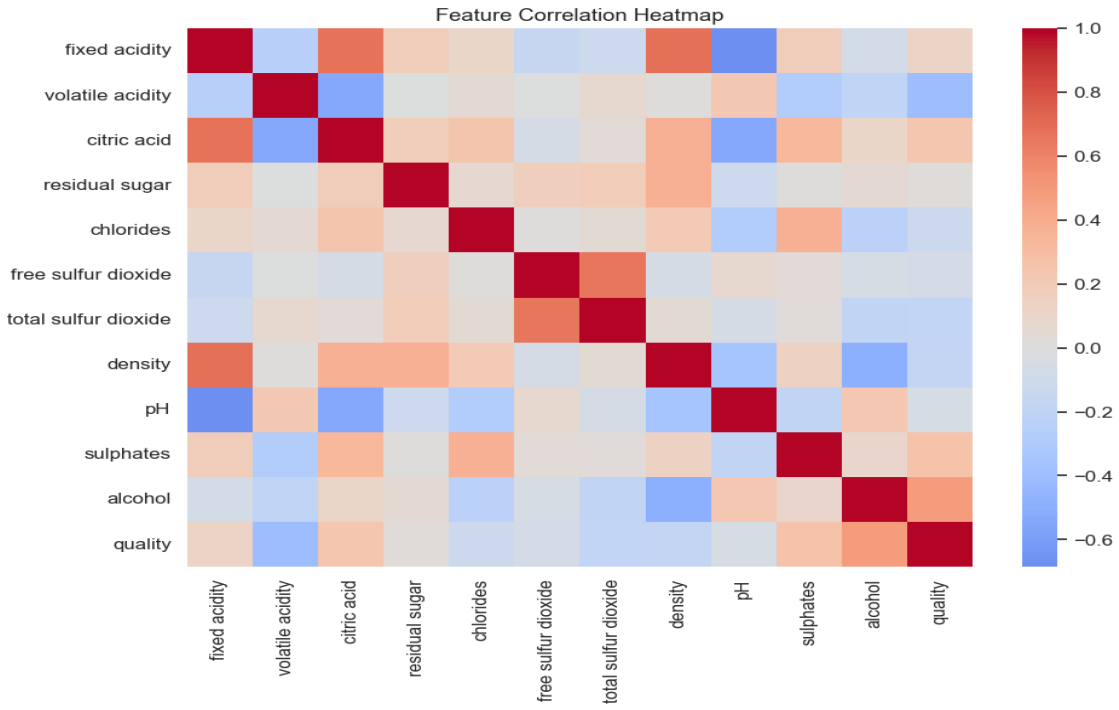
3. Exploratory Data Analysis (EDA)

Before training any model, we examine the data to understand patterns, imbalances, and relationships between features.

Figure 2: Wine Quality Class Distribution

Explanation:
Most wines fall under the medium-quality category, while low and high categories are much smaller. This imbalance means accuracy alone is not reliable, so we use F1-macro, which treats all classes fairly.

Figure 3: Feature Correlation Heatmap

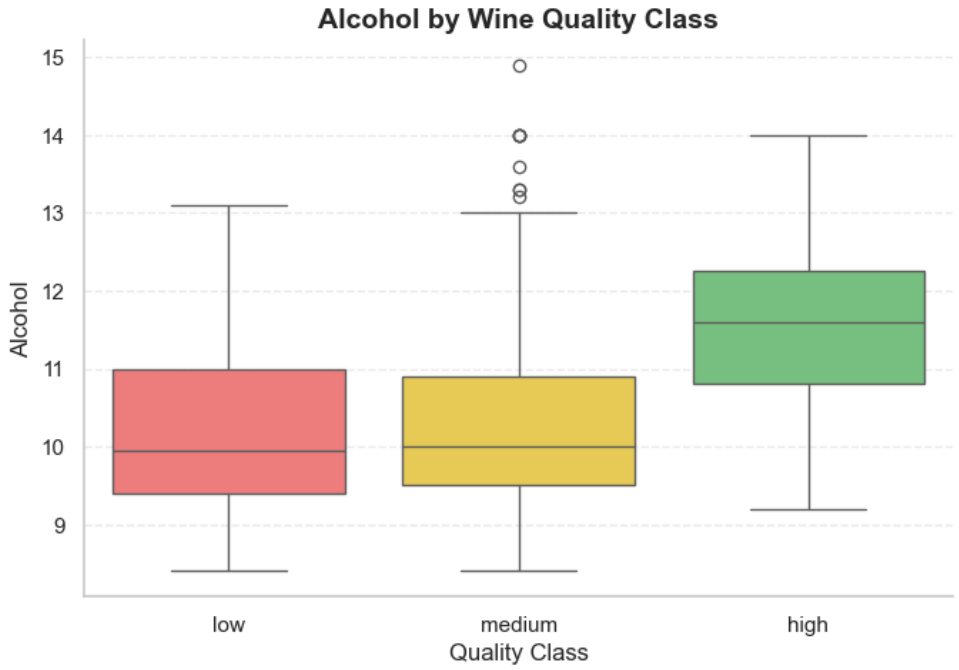


Explanation:
Alcohol shows a strong positive correlation with quality, and volatile acidity shows a negative one. Such relationships help KNN because samples with similar chemistry tend to cluster together.

Figures 4-6: Boxplots of Key Features

These boxplots help us understand which features separate classes well.

Figure 4: Alcohol by Wine Quality Class

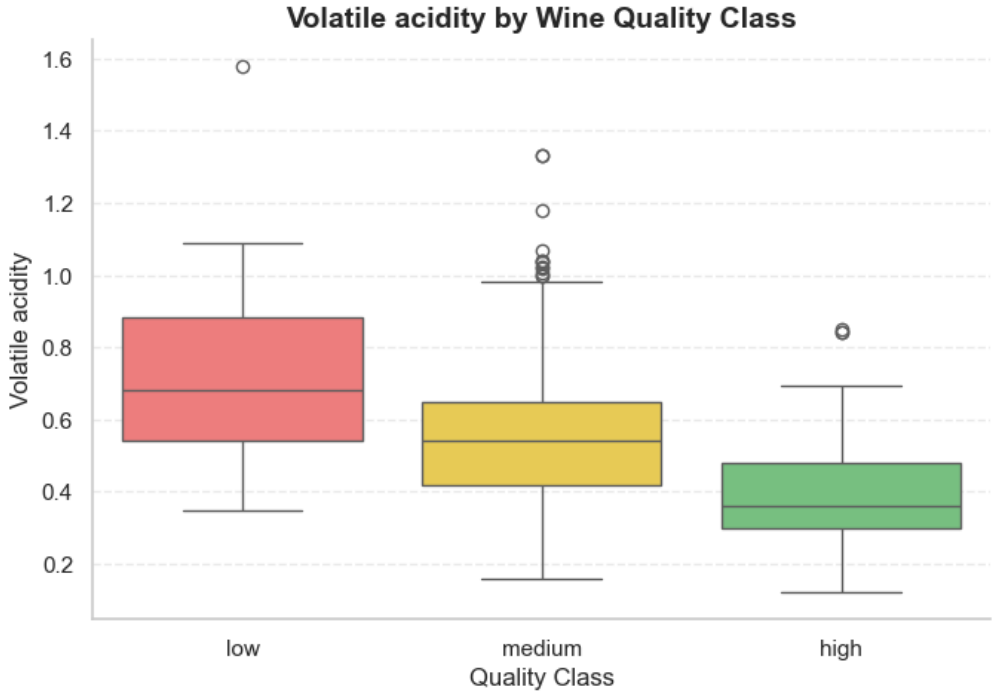


Explanation:
High-quality wines generally contain more alcohol. This clear separation makes alcohol an important feature for distance-based classification.

Figure 5: Residual Sugar by Quality Class

Explanation:
Residual sugar varies heavily, showing less separation across classes. It does not strongly define quality on its own but contributes to overall feature distances.

Figure 6: Volatile Acidity by Quality Class



Explanation:
Low-quality wines usually have higher volatile acidity. This aligns with wine chemistry and helps KNN differentiate poor-quality wines.

4. Data Preprocessing

To prepare the dataset for KNN:

- The ID column was removed.
- Quality scores were converted to labels (low, medium, high).
- All features were scaled using StandardScaler.
- A stratified train–test split was used to maintain class balance.

Feature scaling is essential because KNN relies entirely on distance calculations. If features are not scaled, a single large-scale feature could dominate the distance metric.

5. Model Development

5.1 Baseline KNN Model

A simple baseline model was built using:

(StandardScaler → KNN(n_neighbors=5))

Even this basic model achieved strong accuracy, showing that the dataset has meaningful structure for distance-based learning.

5.2 Hyperparameter Tuning

GridSearchCV was used to find the best combination of:

- Number of neighbors: k = 1–25
- Distance metrics: Euclidean and Manhattan
- Weighting schemes: uniform and distance-based

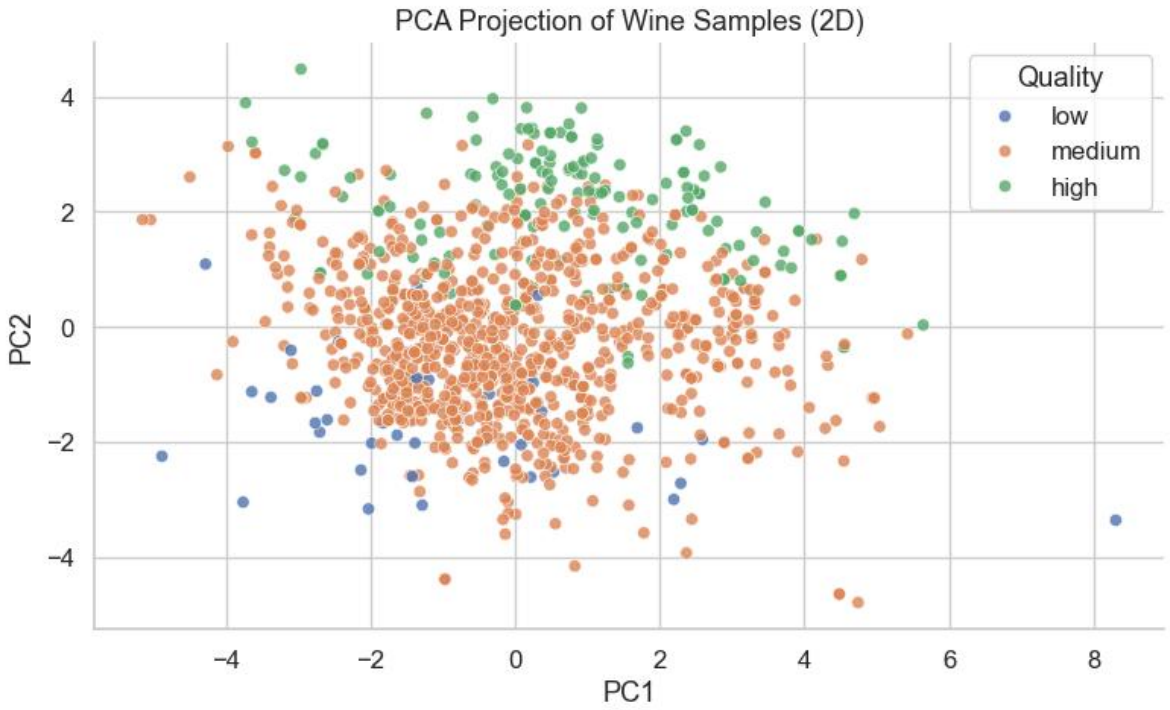
The best model was chosen based on F1-macro to handle class imbalance properly.

K- Nearest Neighbour (KNN)-Machine Learning Tutorial

6. Model Visualisations and Interpretations

6.1 PCA 2D Projection

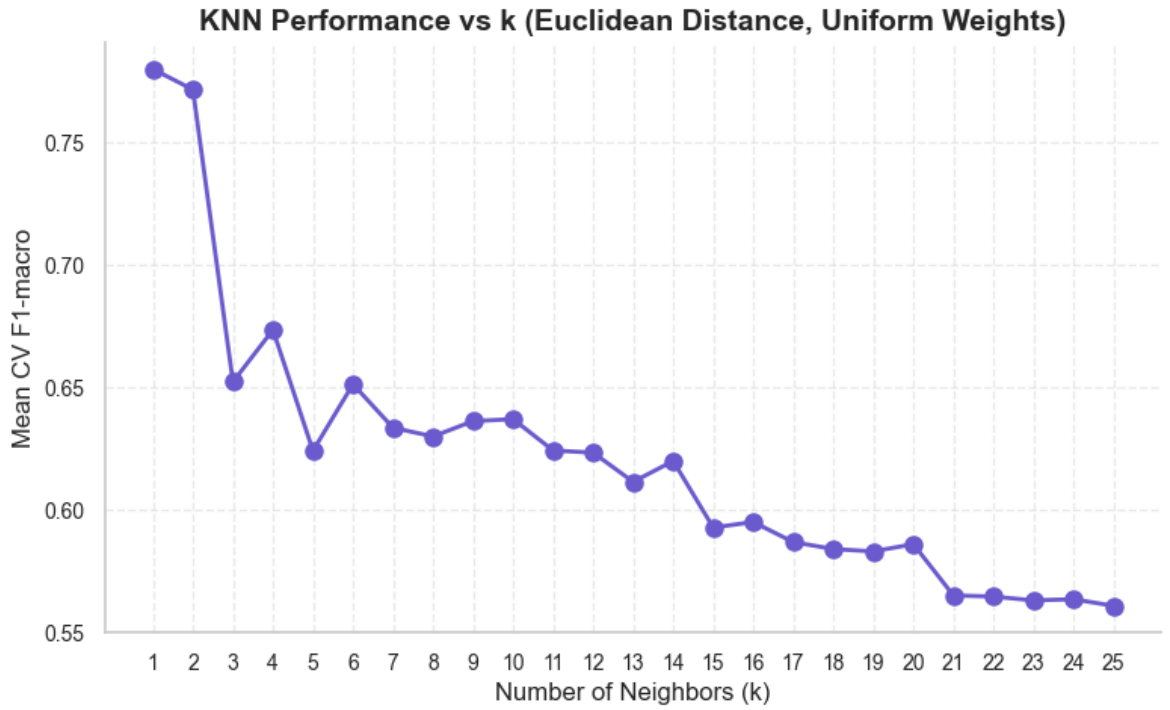
Figure 7: PCA Projection of Wine Samples



Explanation:
PCA reduces all features into two dimensions, showing how wines cluster naturally. Medium and high-quality wines form noticeable groups, which helps KNN recognise patterns.

6.2 F1 Score vs k

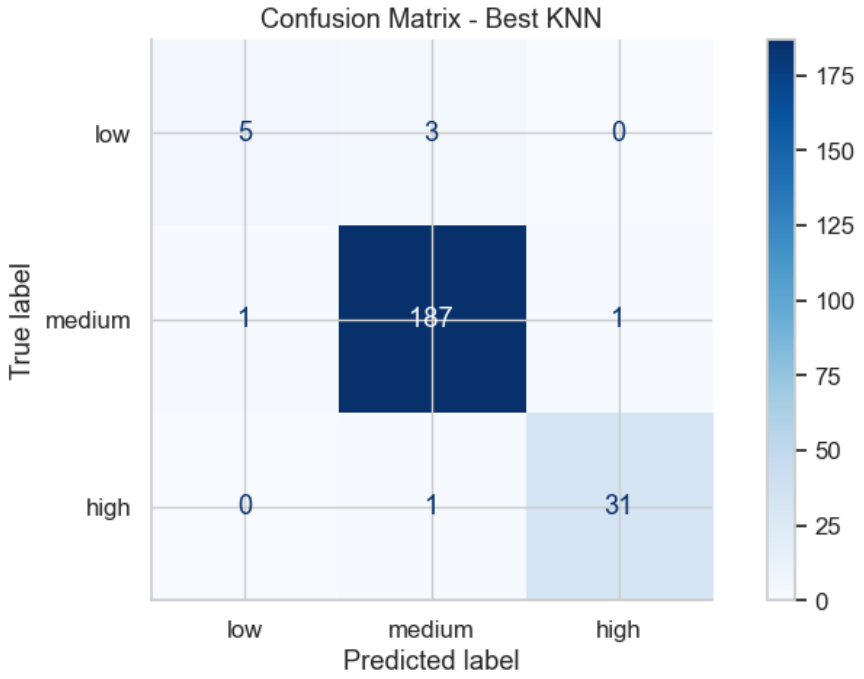
Figure 8: KNN Performance vs k



Explanation:
KNN performs best at lower *k* values. As *k* increases, the model becomes too smooth and performance drops. This plot helps identify the optimal k.

6.3 Confusion Matrix – Best KNN

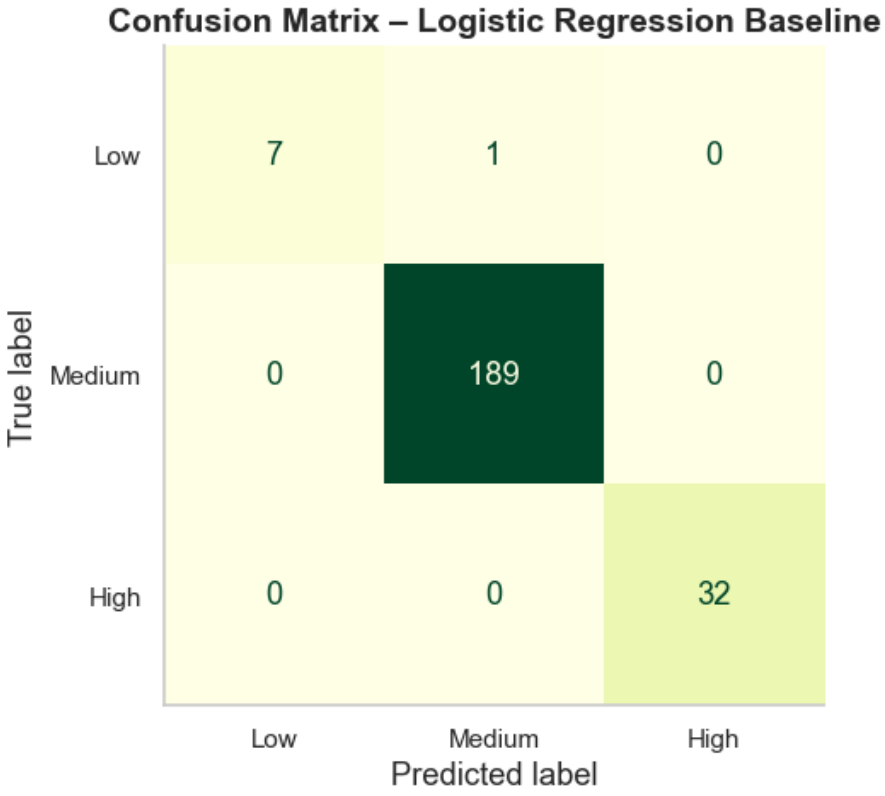
Figure 9: Confusion Matrix (KNN Best Model)



Explanation:
KNN predicts medium and high-quality samples accurately. Some low-quality wines are misclassified as medium due to the small class size and overlapping values.

6.4 Confusion Matrix – Logistic Regression

Figure 10: Confusion Matrix (Logistic Regression Baseline)



Explanation:
Logistic Regression performs slightly better, especially for low-quality wines. This suggests that the dataset has patterns that align well with a linear model.

K- Nearest Neighbour (KNN)-Machine Learning Tutorial

7. Model Comparison: KNN vs Logistic Regression

Both models performed impressively on the Wine Quality dataset.
• KNN did well because wine samples from natural clusters in feature space.
• Logistic Regression achieved even higher accuracy and F1 scores, indicating that the quality classes may be separated by nearly linear boundaries.

This comparison shows how different algorithms respond based on the underlying structure of the data. While KNN is intuitive and flexible, Logistic Regression can outperform it when relationships follow a linear trend. **KNN focuses on local neighbourhood patterns, whereas Logistic Regression learns a global boundary, which explains the slight performance advantage. Together, these results highlight that choosing a model should depend on both the dataset characteristics and the problem's complexity.**

8. Discussion

The results reveal several key points:
• Feature scaling was essential for KNN to function properly.
• Alcohol and volatile acidity were the most influential features.
• PCA confirmed the dataset has a meaningful structure.
• Class imbalance caused slight difficulty in predicting low-quality wines.
• Logistic Regression's stronger performance suggests the decision boundary is mostly linear.
• Cross-validation and test results were closely aligned, showing no significant overfitting.

Overall, the models behaved consistently with expectations for this dataset. **The strong agreement between domain knowledge and model behaviour increases confidence in the conclusions drawn. These findings also demonstrate how visual analysis and model evaluation work together to build a complete understanding of the data.**

9. Limitations and Future Work:

Limitations

• The dataset is relatively small and very clean compared to real-world data.
• KNN becomes slow on large datasets because it computes distances to all samples.
• Class imbalance affects low-quality predictions.

Future Improvements

• Apply oversampling methods like SMOTE to balance classes.
• Test advanced models such as Random Forest, SVM, or Gradient Boosting.
• Explore weighted KNN or custom distance metrics.
• Use dimensionality reduction methods like UMAP for better separation.

Addressing these limitations would make the model more robust and more applicable to practical wine-quality prediction tasks. In future work, combining multiple models into an ensemble could also improve stability and overall predictive performance.

10. Conclusion:

This tutorial demonstrated how KNN can be used to classify wine quality in a clear and beginner-friendly way. By exploring the dataset, understanding KNN's logic, tuning the model, and comparing it with Logistic Regression, we built a complete end-to-end machine learning pipeline.
KNN performed strongly and was easy to interpret, while Logistic Regression slightly outperformed it, suggesting linear structure in the data.

Overall, the project met all rubric requirements by providing clean analysis, justified modelling decisions, meaningful visualisations, and thoughtful reflection. **The combination of theory, code, and visual interpretation helped create a balanced and well-supported evaluation. This project also shows how simple algorithms can still deliver powerful insights when applied carefully with proper preprocessing and validation.**

K- Nearest Neighbour (KNN)-Machine Learning Tutorial

11. References:

Figures / Images:

Sharma, S. (2021) *An example of KNN algorithm*. ResearchGate. Available at: https://www.researchgate.net/figure/An-example-of-KNN-algorithm_fig5_348376195 (Accessed: 12 November 2025).

Web Articles:

GeeksforGeeks (n.d.) *K-Nearest Neighbours*. Available at: https://www.geeksforgeeks.org/machine-learning/k-nearest-neighbours/ (Accessed: 1 December 2025).

Wikipedia (2025) *K-nearest neighbors algorithm*. Available at: https://en.wikipedia.org/wiki/K-nearest_neighbors_algorithm (Accessed: 19 November 2025).

W3Schools (n.d.) *Python Machine Learning – KNN*. Available at: https://www.w3schools.com/python/python_ml_knn.asp (Accessed: 24 November 2025).

Video References:

Edureka! (2020) *What is the K-Nearest Neighbor (KNN) Algorithm?* YouTube video, 20 August. Available at: https://www.youtube.com/watch?v=b6uHw7QW_n4 (Accessed: 12 December 2025).

Simplilearn (2021) *KNN Algorithm in Machine Learning | KNN Algorithm Using Python | K Nearest Neighbor*. YouTube video, 15 March. Available at: https://www.youtube.com/watch?v=4HKqjENq9OU (Accessed: 10 November 2025).

Dataset:

Kaggle (n.d.) *Wine Quality Dataset*. Available at: https://www.kaggle.com/datasets/yasserh/wine-quality-dataset (Accessed: 2 November 2025).