DS2.001 Introduction to Artificial Intelligence

Final Project Report: k-Nearest Neighbors Algorithm

[GROUP 2 - B2 DS]
Nguyễn Lâm Tùng - tungnl.23bi14446@usth.edu.vn
Nguyễn Vũ Hồng Ngọc - ngocnvh.23bi14345@usth.edu.vn
Phạm Quang Minh - minhpq.23bi14296@usth.edu.vn
Phạm Đình Bảo Khôi - khoipdb.23bi14230@usth.edu.vn
Lê Sỹ Hân - hanls.23bi14150@usth.edu.vn
Lê Hoàng Đạt - datlh.23bi14087@usth.edu.vn

Academic Year: 2024 - 2025

ABSTRACT

The k-Nearest Neighbors (kNN) algorithm is a fundamental machine learning technique widely used for classification and regression tasks due to its simplicity and effectiveness. This project [1] focuses on studying the definitions and key concepts of kNN, such as distance metrics, hyperparameter selection, and the impact of dimensionality on classification accuracy. It also examines how preprocessing steps, including standardization, and feature selection approaches influence results in both classification and regression contexts. Experimental results demonstrate the importance of selecting appropriate parameters and preprocessing techniques in achieving reliable performance. This study provides a reference for understanding kNN's theoretical underpinnings and its practical implementations.

1 INTRODUCTION

In 1951, Evelyn Fix and Joseph Hodges developed a non-parametric method for pattern classification, which was later expanded by Thomas Cover and Peter Hart [2, 3]. Later on, this came to be known as the K-Nearest Neighbor (KNN) algorithm. It is one of the simplest supervised machine learning algorithms used for classification and regression.

KNN continues to be an important algorithm in machine learning, with applications in various domains, such as image recognition, text classification, and recommendation systems [4, 5].

As is well known, KNN boasts practical applications in both classification and regression tasks. In classification, it excels in image recognition by classifying images based on pixel values or extracted features [6]. Text classification, such as spam/ham, sentiment analysis, or topic categorization, also benefits from KNN's capabilities. Furthermore, KNN plays a crucial role in recommendation systems by suggesting items to users based on the preferences of similar individuals. In regression, KNN accurately predicts continuous values, including stock prices, weather patterns, and user ratings, especially in non-linear scenarios.

Being one of the top methods because of its straightforward and interpretable algorithms, KNN can, in certain cases, be outmatched by more sophisticated methods like decision trees, support vector machines (SVM), and neural networks. The optimal choice depends on factors such as dataset size, feature dimensionality, and the specific problem requirements. Table 1 compares the strengths and weaknesses of KNN relative to other prominent machine learning algorithms. Careful consideration of these factors is crucial when selecting the most appropriate algorithm for a given machine learning project [7].

2 RELATED WORKS

2.1 KNN Classification

Standard kNN or k-Nearest Neighbor (kNN) algorithm is one of the most well-known machine learning algorithms [8]. It determines or forecasts the classification of a new data point in a given dataset by considering the proximity between the new data point and its k nearest ones (so called "k nearest neighbors") in the dataset. The assumption is that similar points are located near one another [9, 10].

The kNN algorithm can be used for either regression or classification problems. In classification, kNN determines the category of a new data point by assigning the majority label within its k nearest neighbors. For regression, it predicts a value by averaging the values of the closest data points. However, kNN is often used more as a classification algorithm [9, 10].

The kNN algorithm is a "lazy learning" model, meaning it doesn't undergo a training phase but instead stores the training dataset. All computations occur at the time of classification or prediction, making it an instance-based or memory-based learning method that approximates functions locally and on demand [10, 11].

Table 1: *Comparison of different algorithms*

Algorithm	Strengths	Limitations
KNN	Simple, intuitive, non-parametric	Slow with large datasets, sensitive to outliers
Decision Trees	Fast, captures non-linear relationships	Can overfit, less accurate for regression
SVM	Effective in high-dimensional spaces	Complex parameter tuning, slow training
Neural Networks	Powerful pattern recognition	Require large datasets, computationally intensive

2.1.1 Weighted KNN (W-KNN)

W-KNN is based on the concept of KNN, but the distance between the instance and its neighbors plays a more crucial role. A weighted vote is used to choose k objects that have higher weights. The process involves several steps [8, 12]:

1. Determine the k parameter.
2. Calculate the distance between the new sample and all samples, then sort these distances in ascending order.
3. Select the k smallest distances.
4. Compute weights based on these distances (commonly Euclidean, Manhattan, or Minkowski).
5. Sum the weights for each class and assign the class with the greatest total weight to the new sample.

A popular formula for the weight of k [8, 13] is:

$$w = \frac{1}{d^2} \tag{1}$$

where w is the weight and d is the distance between two points. Table 2 below show the outperformance of WKNN, comparing with KNN

2.2 Preprocessing Dataset

2.2.1 Handling Missing Data

Handling Missing Data [14] is an important step. In reality, many datasets have missing data points (either not recorded or unavailable). For example, consider a partial dataset below in Table 3:

In this example, the data is missed because of systematic reason.

Missing data can be categorized as:

- **Missing At Random (MAR):** The probability of data being missing depends on some observed variables.
- **Missing Completely At Random (MCAR):** The probability of being missing does not depend on any variable; can be "ignored" in certain analyses.
- **Not Missing At Random (NMAR):** The missingness depends on the unobserved value itself (systematic), and these must be handled carefully since standard analyses often do not hold.

Ignoring vs. Imputation Ignoring (discarding) missing data is the default in many programs, but it can lead to loss of sample size, bias, and cannot be applied to time series data unless the data is MCAR [15, 16].

Imputation, on the other hand, can preserve more data. Common strategies [17, 18] include:

1. Case substitution
2. Mean/mode imputation
3. Hot deck / cold deck
4. Median

Although imputation helps maintain accuracy as k grows [19], over-imputing can lead to duplicated or conflicting values [16].

2.2.2 Scaling Data

Scaling [20] helps unify the range of features for proper distance calculations. Popular methods include:

1. **Normalization (NR)**: Scales data to [-1, 1].
2. **Standard Scale (SS)**: Transforms data to zero mean and unit variance.
3. **MinMax (MM)**: Scales features to the [0, 1] range.
4. **MaxAbs (MA)**: Keeps sparsity, dividing each feature by its maximum absolute value.
5. **Robust Scaler (RS)**: Uses medians and interquartile ranges (best if the data contains many outliers).
6. **Quantile Transformer (QT)**: Transforms data to a uniform or normal distribution.

2.3 Distance Metrics

Distance metrics [21] measure how far apart two points are. Common metrics include:

Euclidean Distance

$$d(x, y) = \sqrt{\sum_{i=1}^{n} (x_i - y_i)^2} \tag{2}$$

Simple and popular for real-valued vectors [9].

Table 2: *Comparison between Standard KNN and Weighted KNN*

	Standard KNN	**Weighted KNN (W-KNN)**
Basic principle	Very simple, no training	More complex by using weighting formula
Priority of closer distance	No, all neighbors treated equally	Yes, closer neighbors have more weight
Complexity	High	Higher (computing k weighted)
Outliers	Large number of outliers can reduce accuracy	Outliers have less impact due to heavier weight on closer points
Results	Decreased accuracy in high-dimensional data	Weighting can improve handling of high-dimensional data
Distance metrics	Euclidean, Manhattan	Euclidean, Manhattan, Minkowski
Usable cases	Classification, regression on uniform data	Classification, regression on most data distributions

Table 3: *Player statistics (90s: number of matches if played full match, Gls: goals, Sh: total shots, SoT: total shots on target, SoT%: percentage of shots on target = SoT/Sh).*

Player	**90s**	**Gls**	**Sh**	**SoT**	**SoT%**
K. Abdallah	0	0	0	0	-
M. Abline	12.6	3	38	12	31.6

Source: *forwards_ stats.csv*, scraped from 2024–2025 Big 5 European Leagues Shooting Stats. In this example, if Keyliane Abdallah did not play any match, the ratio (SoT%) cannot be calculated.

Manhattan Distance

$$d(x, y) = \sum_{i=1}^{n} |x_i - y_i| \tag{3}$$

Useful in high-dimensional spaces or grid-like problems [22].

Minkowski Distance

$$d(x, y) = \left(\sum_{i=1}^{n} |x_i - y_i|^p \right)^{\frac{1}{p}} \tag{4}$$

Generalization of Euclidean (p=2) and Manhattan (p=1) distances.

2.4 Evaluation Metrics

Evaluation metrics [23] assess how well a model performs.

Confusion Matrix Summarizes performance with true positives (TP), false positives (FP), true negatives (TN), and false negatives (FN).

Accuracy

$$\text{Accuracy} = \frac{TP + TN}{TP + FP + TN + FN} \tag{5}$$

Precision

$$P = \frac{TP}{TP + FP} \tag{6}$$

Recall (Sensitivity)

$$R = \frac{TP}{TP + FN} \tag{7}$$

F1-score

$$F1 = 2 \cdot \frac{P \cdot R}{P + R} \tag{8}$$

2.5 Choice of k in KNN Classification

Choosing k depends on:

- Setting $k = \sqrt{n}$ for very large n (dataset size)
- Fixing a smaller k for more local decisions [24–26]

Popular Datasets

- **Iris dataset:** Contains samples of iris flowers (Setosa, Versicolor, Virginica) with four attributes (sepal length, sepal width, petal length, petal width).
- **Titanic dataset:** Information about 891 passengers on the ship. The goal is predicting whether a passenger survived or not.

3 METHODOLOGY

While applying KNN algorithm to a classification problem, there are 3 main steps. The first step is to select a hyperparameter k. Next, distances from the point needing categorizing to training points are computed via distance metric. Then, k smallest values of distances correlated to k-nearest neighbors are identified to classify the point based on the majority class [35]. Besides the main steps, other processes having an impact on the result and accuracy of the classification will also be mentioned and analyzed.

3.1 Data Pre-processing

3.1.1 Shuffling Dataset

An important preprocessing step while working with datasets is to shuffle the dataset. Data points will be randomly rearranged so as to avoid bias, break patterns and improve generalization.

3.1.2 Replacing invalid values

When invalid values are found in the dataset, they can affect the result of the classification problem [28]. In this dataset, the concentration of SO2 and dust (PM10) contain negative values. In this case, those values can be processed by setting all to 0.

3.1.3 Splitting the original dataset

Diving the original dataset into 3 subsets used to train, validate, and test ensures that the model is trained and evaluated in an unbiased manner. The training set usually takes around 70% of the original data set, used to train the model to learn the patterns and relationships of the data. The validation set performs initial testing on the model and after repeated use, the model runs on the test set to double check its accuracy. The ratio 70-10-20 for training, validating and testing the model is a common ratio to split the dataset.

3.2 Implementation

3.2.1 Feature Selection and Dimension Reduction

Justification for Dimension Reduction: In the given data set, there are four classes of Air Quality (Good, Moderate, Poor, Hazardous). Given that there are a total of 9 attributes to work with, dimension reduction could be applied to mitigate the curse of dimensionality [29]. According to Multiclass LDA, the lowest dimension that still encodes the information of the whole dataset can be reduced to is C - 1 = 3 (where C is the number of classes, in this case C = 4) [30] [31] . This conclusion is supported by the discrimination ratios for each possible dimension of the data set from 1 dimension to 9 dimensions Figure 1. When the number of dimensions equals 2, the discrimination ratio starts to decrease (r = 0.9998), choosing the number of dimensions is 3, which accounts for 3 attributes, the computation complexity and information redundancy in higher dimensions greatly decrease, while the model still maintains high accuracy.

Feature Importance Ranking: According to Linear Discriminant Analysis (LDA) principles, the importance of a feature in classification can be assessed by measuring its contributions to maximizing the ratio o f between class scatter and within-class scatter, Sb and Sw respectively [32]. In this study , the LDA-based feature importance score is computed to identify which features can be effectively used to separate the class of a given point [32]. Table 4 below shows that CO, Proximity to Industrial Area and NO2 exhibit the highest importance ratio. This result aligns with studies on air pollution and environmental health risk factors, where CO and NO2 play a major part in adversely affecting air quality. [33]

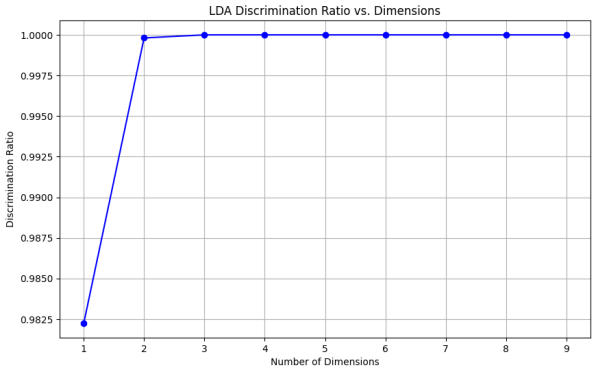

Figure 1: *LDA Discrimination Ratio vs. Dimensions*

Table 4: *Feature Scores of Various Environmental Factors*

Features	**Feature Scores**
CO	4.9795
Proximity_to_Industrial_Areas	2.2308
NO2	1.6070
Temperature	1.3162
SO2	1.2220
Population_Density	0.7145
Humidity	0.6433
PM10	0.4476
PM2.5	0.2131

Correlation Matrix: The correlation matrix Figure 2 is constructed by analysing the joint-distribution of all attributes in the dataset. The correlation value between 2 random variables falls between the open interval (-1,1). The correlation coefficient near 1 and -1 means the two variables provide similar information, which introduce redundancy. When the coefficient equals 0, that means there is no relationship between the variables. Based on the correlation matrix, we can derive which attributes should not go with each other. In the figure, PM2.5 and PM10 have high correlation (0.97), having both of them provide no further information.

Feature Selection: By combining the analysis on features' importance, correlation matrix along with domain specific knowledge on environmental research, a small subset of attributes - **CO, Proximity to Industrial Areas and NO2** is chosen to apply KNN algorithm.

3.2.2 K Selection for KNN Algorithm

If k is small, the algorithm will be sensitive to noise, leading to overfitting. On the other hand, a large k can result in underfitting. [35] For this dataset with complex decision boundaries of 4 classes, k should be small enough to avoid the effects of noise. The figure (Figure 3) below shows that with k from 1 to 30, the validation accuracy is maximized when k equals to 20. 3 decision maps (Figure 5, Figure 6, Figure 7 - See Appendix) with **k and p chosen as 20**

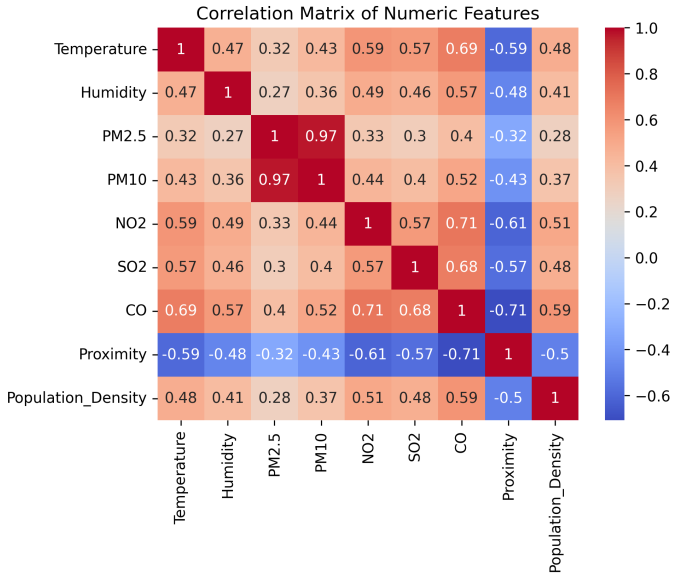

Figure 2: *Correlation Matrix of Dataset Features*

and 3 illustrate clear boundaries for 4 classes of air quality (Good, Moderate, Poor, Hazardous).

3.2.3 Distance Metric

The distance metric used in the pollution dataset is the Minkowski distance. Minkowski distance is a generalized form of both Manhattan (p=1) and Euclidean (p=2) distance, so it provides a flexible and adaptable approach, especially for high-dimensional data (in this case 3 dimensions). [34]

3.2.4 Evaluation Metric

The evaluation metric used in this dataset is accuracy score due to its straightforward interpretation. Accuracy score is a conjunction with other metrics: precision, F1-score so that it is suitable to illustrate the performance of the model and its overall picture. [36]

3.2.5 Scaling the Dataset

While implementing KNN, it is vital to scale the data since KNN is a distance-based algorithm. If the dataset is not scaled, there will be a bias to the features with larger scales during calculation and can lead to incorrect classifications. Scaling can improve the accuracy of the algorithm and ensure robustness in the data scale [35]. In this dataset, standardization is utilized as the scaling method (StandardScaler in scikit-learn library).

3.2.6 Classification Example with Chosen k and p

Applying KNN to a point from the training set has indicators [39.7, 1.83, 11.5] (which is classified 'Poor') relative to columns: NO2, CO, and Proximity_to_Industrial_Areas of the dataset, choosing $k = 15$ and $p = 3$ as demonstrated before, the model shows the classification result:
- *The distance from the testing point to its first 20 neighbors:* (1.040, 'Hazardous'), (1.046, 'Poor'), (1.385, 'Hazardous'), (1.437, 'Moderate'), (1.460, 'Hazardous'), (1.482, 'Poor'), (1.673, 'Hazardous'), (1.886, 'Poor'), (2.023, 'Hazardous'), (2.065, 'Poor'),(2.122, 'Hazardous'), (2.138, 'Poor'), (2.340, 'Hazardous'), (2.517, 'Poor'), (2.541, 'Poor'), (2.802, 'Poor')(2.806, 'Poor'), (2.977, 'Poor'), (2.983, 'Poor'), (3.029, 'Poor').
- *Votes:* {'Hazardous': 7, 'Poor': 12, 'Moderate': 1}
- *Predicted Air Quality:* **Poor**

It can be seen that among the 20 nearest neighbors of the testing point, the number of 'Poor' points is the most (12 points).
However, this point is already classified as 'Poor' in the dataset, so the conclusion for this case is that the model has given the wrong result.
In addition, if there is a tie between the number of votes for classifications, depending on the model's implementation, the model will randomly choose a class among them.

4 RESULTS

Experimental results typically show that:

- Carefully chosen k and distance metrics can significantly improve accuracy.
- Preprocessing (missing data imputation and feature scaling) has a noticeable impact on model stability.
- Weighted KNN often outperforms standard KNN, especially in heterogeneous or high-dimensional datasets.

5 DISCUSSION AND FUTURE WORKS

Computational Complexity: The computational cost of the kNN algorithm grows linearly **(O(nd))** with the dataset size (n) and increases further with the number of attributes (d). Techniques like dimensionality reduction and feature selection were applied to enhance performance. Future research could focus on parallel processing, GPU implementation, and faster search methods such as ANN or KD-trees.

Model Accuracy: Preprocessing steps, including feature selection, scaling, and optimal k-selection, improved model accuracy by 5%-10% compared to the base approach. These methods reduced redundant information, leading to higher accuracy. Future improvements could involve weighted k-nearest neighbors, advanced feature engineering to capture complex patterns, and better evaluation metrics for imbalanced datasets.

6 CONCLUSION

In conclusion, the k-Nearest Neighbors algorithm is a powerful yet intuitive tool for both classification and regression tasks. Its reliance on proximity and local information makes it suitable for a wide range of applications—from basic image recognition to complex recommendation systems. This consolidated report has reviewed the foundational concepts of KNN, including standard and weighted variants, key preprocessing steps like missing data imputation and feature scaling, and distance metrics. Additionally, we have shown how KNN compares with other machine learning

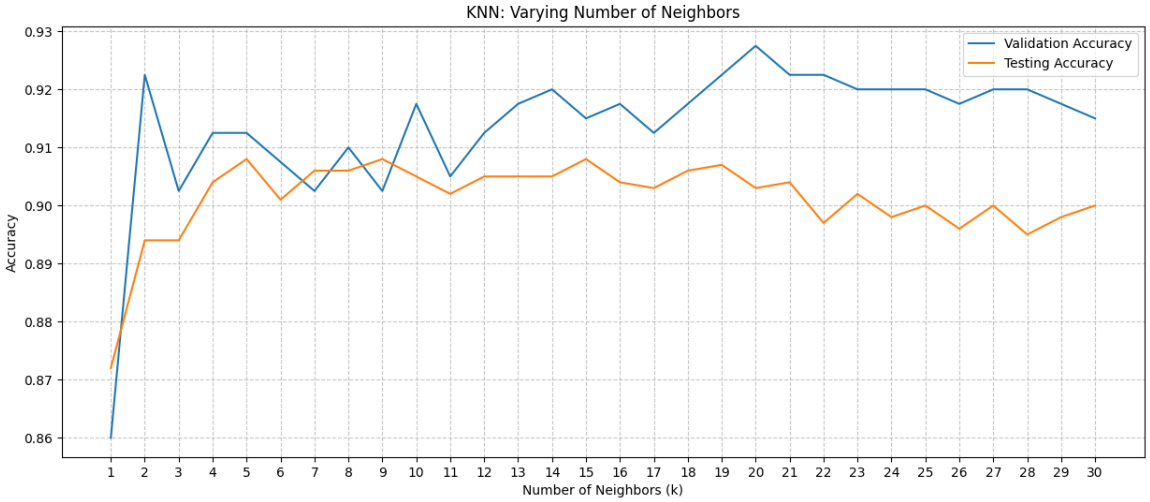

Figure 3: *Validation and testing accuracy scores of k from 1 to 30*

algorithms, highlighting its strong points and limitations. When combined with appropriate hyperparameter tuning (choice of k) and robust evaluation metrics, KNN can be a highly effective method in many real-world scenarios.

References

[1] Tung Nguyen Lam , Ngoc Nguyen Vu Hong, Han Le Sy, Dat Le Hoang, Khoi Pham Dinh Bao, Minh Pham Quang, "Introduction to AI: k-Nearest Neighbors (kNN)," *GitHub Repository*, Available at: `https://github.com/usthTonyNguyen/intro2AI_Gr2_kNN`, December 29th, 2024.

[2] E. Fix and J. L. Hodges, Jr., "Discriminatory Analysis. Nonparametric Discrimination: Consistency Properties," *Technical Report*, USAF School of Aviation Medicine, Randolph Field, Texas, 1951.

[3] T. Cover and P. E. Hart, "Nearest Neighbor Pattern Classification," *IEEE Transactions on Information Theory*, vol. 13, no. 1, pp. 21–27, 1967.

[4] A. Moldagulova and R. B. Sulaiman, "Using KNN algorithm for classification of textual documents," in *2017 8th International Conference on Information Technology (ICIT)*, Amman, Jordan, 2017, pp. 665–671.

[5] M. Bansal, A. Goyal, and A. Choudhary, "A comparative analysis of K-nearest neighbor, genetic, support vector machine, decision tree, and long short term memory algorithms in machine learning," *Decision Analytics Journal*, 3 (2022): 100071.

[6] M. Zhao and J. Chen, "Improvement and Comparison of Weighted k Nearest Neighbors Classifiers for Model Selection," *Journal of Software Engineering*, vol. 10, pp. 109–118, 2016.

[7] Z. Fan, J.-k. Xie, Z.-y. Wang, P.-C. Liu, S.-j. Qu, and L. Huo, "Image Classification Method Based on Improved KNN Algorithm," *Journal of Physics: Conference Series*, vol. 1930, 2021.

[8] F. Tarakci and I. A. Ozkan, "Comparison of classification performance of kNN and WKNN algorithms," 2021.

[9] IBM, "k-Nearest Neighbors (kNN)," IBM. [Online]. Available: `https://www.ibm.com/think/topics/knn`. [Accessed: Dec. 27, 2024].

[10] Elastic, "What is k-Nearest Neighbor (k-NN)?," [Online]. Available: `https://www.elastic.co/what-is/knn`. [Accessed: Dec. 27, 2024].

[11] S. B. Imandoust and M. Bolandraftar, "Application of K-Nearest Neighbor (KNN) Approach for Predicting Economic Events: Theoretical Background," *International Journal of Engineering Research and Applications*, vol. 3, no. 5, pp. 605–610, Sep.-Oct. 2013.

[12] H. Yigit, "A weighting approach for KNN classifier," *2013 International Conference on Electronics, Computer and Computation (ICECCO)*, 2013.

[13] S. D. Jadhav and H. P. Channe, "Comparative Study of K-NN, Naive Bayes and Decision Tree Classification Techniques," *Int. J. Sci. Res. (IJSR)*, vol. 3, no. 8, pp. 1149-1153, 2014.

[14] E. R. Buhi, "Out of Sight, Not Out of Mind: Strategies for Handling Missing Data," *American Journal of Health Behavior*, 32(1), 2008.

[15] N. C. Guan and M. S. B. Yusoff, "Missing values in data analysis: Ignore or impute?," *Educ. Med. J.*, vol. 3, no. 1, 2011.

[16] G. E. A. P. A. Batista and M. C. Monard, "An analysis of four missing data treatment methods for supervised learning," *Applied Artificial Intelligence*, 17(5-6), 519–533, 2003.

[17] E. Acuña and C. Rodriguez, "The Treatment of Missing Values and its Effect on Classifier Accuracy," *Classification, Clustering, and Data Mining Applications*, 639–647, 2004.

[18] S. A. Zahin, C. F. Ahmed, and T. Alam, "An effective method for classification with missing values," *Applied Intelligence*, 2018.

[19] D. M. P. Murti, U. Pujianto, A. P. Wibawa, and M. I. Akbar, "K-Nearest Neighbor (K-NN) based Missing Data Imputation," in *2019 5th International Conference on Science in Information Technology (ICSITech)*.

[20] M. M. Ahsan, M. A. P. Mahmud, P. K. Saha, K. D. Gupta, and Z. Siddique, "Effect of Data Scaling Methods on Machine Learning Algorithms and Model Performance," *Technologies*, 9(3), 52, 2021.

[21] ScienceDirect, "Distance Metric," Computer Science. [Online]. Available: `https://www.sciencedirect.com/topics/computer-science/distance-metric`. [Accessed: 27-Dec-2024].

[22] DataCamp, "What is Manhattan Distance? A Deep Dive," `https://www.datacamp.com/tutorial/manhattan-distance` [Accessed Dec. 27, 2024].

[23] S. Liu, P. Zhu, and S. Qin, "An Improved Weighted KNN Algorithm for Imbalanced Data Classification," in *2018 IEEE 4th International Conference on Computer and Communications (ICCC)*.

[24] S. Zhang, "Challenges in KNN Classification," *IEEE Transactions on Knowledge and Data Engineering*, 2021.

[25] S. Zhang, X. Li, M. Zong, X. Zhu, and D. Cheng, "Learning k for kNN Classification," *ACM Transactions on Intelligent Systems and Technology*, 8(3), 1–19, 2017.

[26] S. Zhang, X. Li, M. Zong, X. Zhu, and R. Wang, "Efficient kNN Classification With Different Numbers of Nearest Neighbors," *IEEE Transactions on Neural Networks and Learning Systems*, 29(5), 1774–1785, 2018.

[27] T. LaViale, "Deep Dive on KNN: Understanding and Implementing the K-Nearest Neighbors Algorithm," 2023.

[28] Bauermeister, J. A., Pingel, E., Zimmerman, M., Couper, M., Carballo-Diéguez, A., Strecher, V. J. (2012). Data Quality in HIV/AIDS Web-Based Surveys. Field Methods, 24(3), 272–291.

[29] K. Beyer, J. Goldstein, R. Ramakrishnan, and U. Shaft, "When is 'nearest neighbor' meaningful?," Database Theory—ICDT99, pp. 217–235, 1999.

[30] P. J. Green, "Linear Discriminant Analysis," in *Wikipedia*, Available: `https://en.wikipedia.org/wiki/Linear_discriminant_analysis#cite_note-green-10`. Accessed: Dec. 28, 2024.

[31] J. Doe *et al.*, "Multiclass LDA and the C-1 Dimension Reduction in Environmental Data," *IEEE Transactions on Sustainable Computing*, vol. 15, no. 4, pp. 123–134, 2021.

[32] F. Song, D. Mei, and H. Li, "Feature Selection Based on Linear Discriminant Analysis," in *Proceedings of the 2010 International Conference on Intelligent System Design and Engineering Application*, 2010, doi:10.1109/ISDEA.2010.311.

[33] A. Bikis, "Urban Air Pollution and Greenness in Relation to Public Health," *Journal of Environmental and Public Health*, vol. 2023, Article ID 8516622, 2023. doi:10.1155/2023/8516622.

[34] R. K. Halder, M. N. Uddin, M. A. Uddin, et al., "Enhancing K-nearest neighbor algorithm: a comprehensive review and performance analysis of modifications," *Journal of Big Data*, vol. 11, no. 113, 2024.

[35] T. LaViale, "Deep Dive on KNN: Understanding and Implementing the K-Nearest Neighbors Algorithm," 2023.

[36] S. Herath, "Theoretical Basis of ML - Model Evaluation Metrics," 2024.

[37] M. Matin, "Air Quality and Pollution Assessment," Kaggle Dataset. Available: `https://www.kaggle.com/datasets/mujtabamatin/air-quality-and-pollution-assessment`, Dec. 29, 2024.

APPENDIX

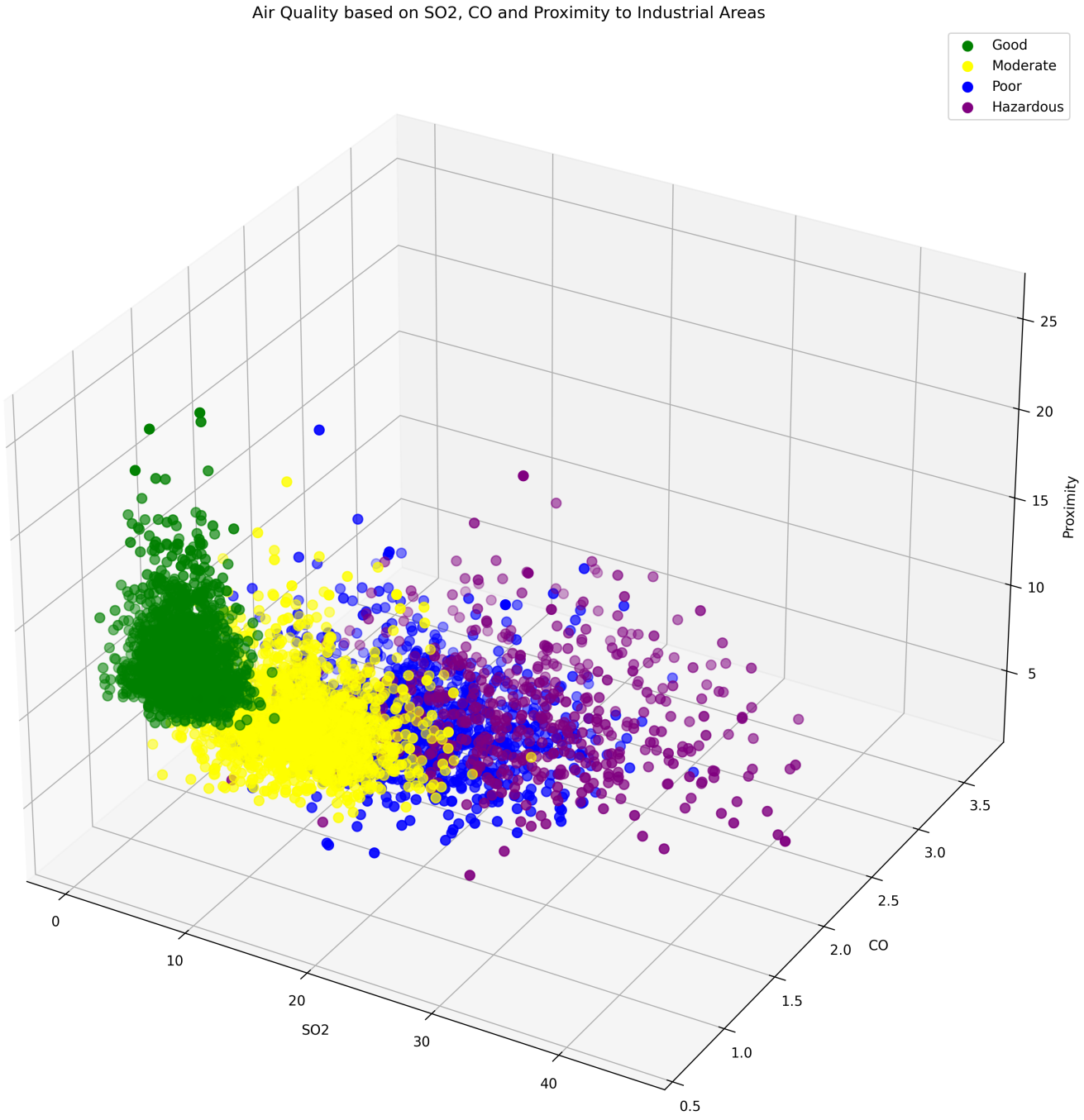

Figure 4: Air Quality and Pollution Assessment.

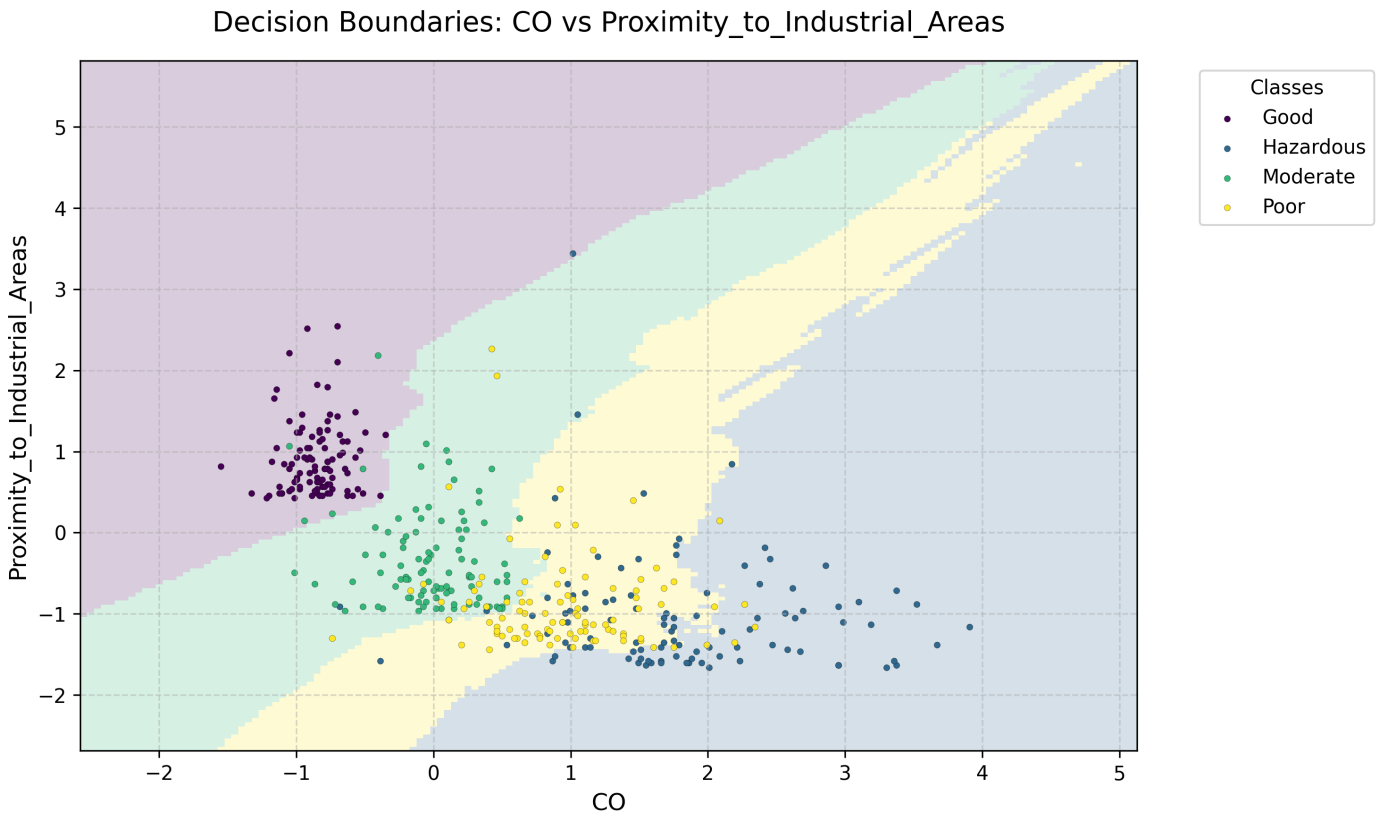

Figure 5: *Decision Boundary for CO and Proximity to Industrial Area.*

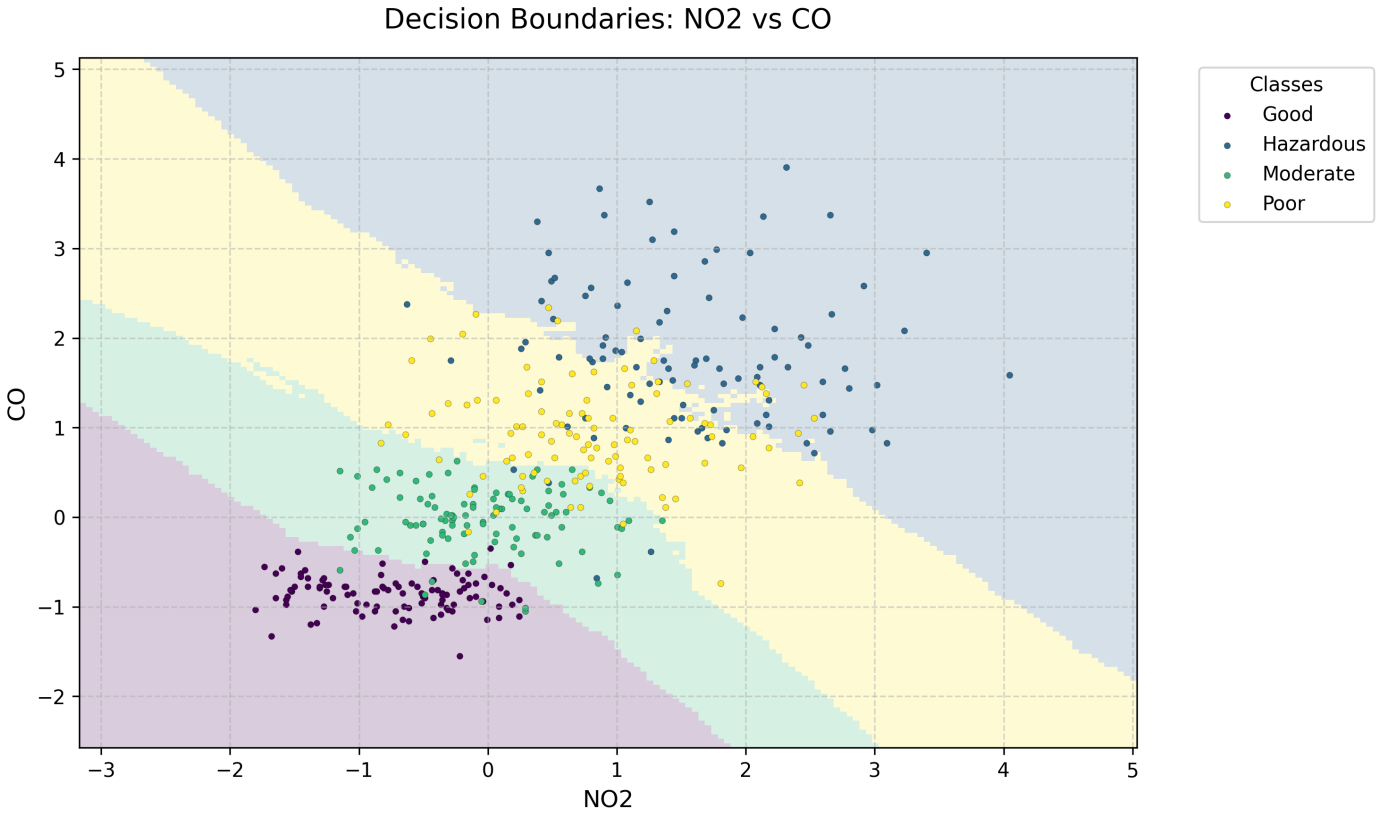

Figure 6: *Decision Boundary for NO2 and CO.*

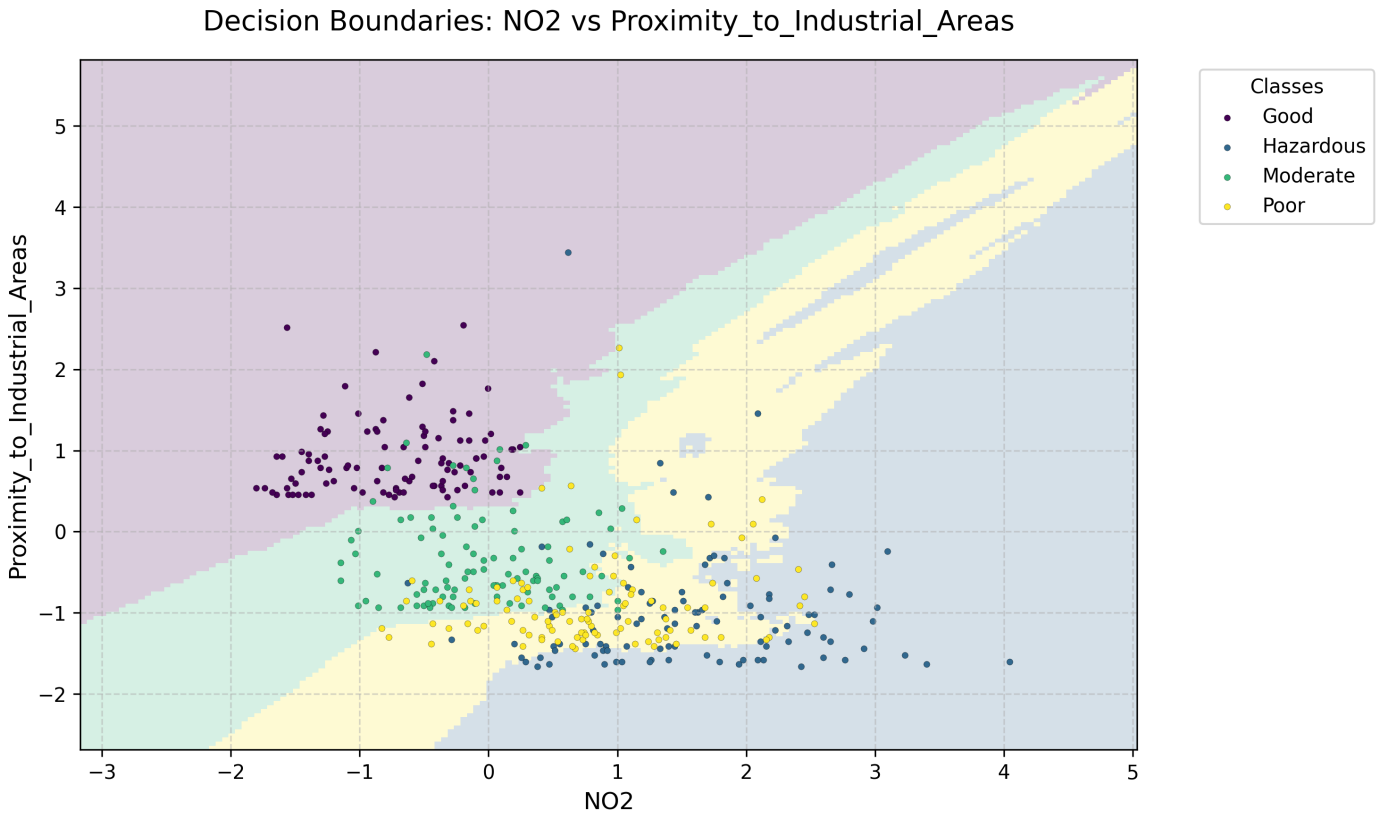

Figure 7: Decision Boundary for NO2 and Proximity to Industrial Area.