Adaptability and Limitations of k-Nearest Neighbors and Its Variants in Classification and Regression

Srikanta Mehta SMEHTA44@JH.EDU
Department of Computer Science
Johns Hopkins University
Baltimore, MD 21205, USA

Abstract

This study investigates the performance differences between the k-Nearest Neighbors (KNN) algorithm and its two primary variants—Edited KNN and Condensed KNN—across a series of machine learning tasks involving both classification and regression. Utilizing a group of datasets from the UCI Machine Learning Repository, we test these algorithms to assess their performance, focusing on a comprehensive set of metrics including Mean Squared Error (MSE), Mean Absolute Error (MAE), R-squared (R2), and Pearson score for regression tasks, alongside accuracy, precision, recall, and F1 score for classification tasks. Our findings reinforce the adaptability and proficiency of KNN and its variants to handle a wide range of tasks, marked by significant accuracy improvements over null models in the majority of cases. Our findings also support our hypothesis that Edited KNN and Condensed KNN do not demonstrate significant performance improvements over the traditional KNN algorithm in certain contexts, particularly within sparser, highly-dimensional datasets or those with complex class boundaries. Finally, we noticed that all nearest-neighbor models performed significantly better in our classification tasks than in the regression tasks, suggesting limitations in their ability to model continuous outcomes effectively. We end the study with recommendations for future work in exploring the effect of alternative data processing methods, model configurations, and optimizations.

Keywords: Classification, Regression, k-Nearest Neighbors

1. Introduction

The challenge to identify the most effective machine learning algorithms for specific tasks, such as classification or regression, continues to be a major focus in Artificial Intelligence research. Understanding and optimizing model selection is a crucial key to advancing a range of ML applications, from financial forecasting to healthcare diagnostics. Among the large selection of algorithms, the k-Nearest Neighbors (KNN) algorithm stands out for its simplicity, effectiveness, and adaptability. Since KNN was first developed by Evelyn Fix and Joseph Hodges in 1951, many variants have been made to the algorithm (1). These variants, including the Edited k-Nearest Neighbors and Condensed k-Nearest Neighbors, have been designed to enhance the algorithm's efficiency and accuracy by addressing challenges such as high memory usage and sensitivity to noisy data (2)(3). These improvements have significantly broadened the applicability of KNN in handling diverse problems across different domains. This paper aims to study the differential performance of KNN and its variants—Edited KNN and Condensed KNN—across classification and regression datasets.

This study is focused on the hypothesis that: Edited k-Nearest Neighbors (Edited KNN) and Condensed k-Nearest Neighbors (Condensed KNN) will not demonstrate significant per-

formance improvements over the traditional KNN algorithm across the selected classification and regression datasets. This hypothesis is influenced by several considerations: firstly the potential limitations of Edited and Condensed KNN in small datasets where data reduction might inadvertently remove essential information; secondly the algorithms tradeoff between removing noise and preserving valuable outliers; thirdly the challenge of addressing class imbalance; and finally, the sensitivity to dataset characteristics, such as complex class boundaries and the necessity for tuning additional parameters like epsilon, alongside k parameter tuning. These considerations suggest that the performance of Edited and Condensed KNN is deeply linked to dataset specifics and may not universally surpass traditional KNN.

The remainder of this report is organized to methodically explore the primary hypothesis. Section 2, titled *Algorithms and Experimental Methods*, outlines the algorithms, experimental setup, and preprocessing of datasets from the UCI Machine Learning Repository, in preparation for evaluation. The *Results* section (Section 3) presents a detailed comparison of the algorithms' performance, using metrics such as MSE, MAE, Zero-One loss, and F1 scores, to quantify their effectiveness in classification and regression tasks. This is followed by a *Discussion* (Section 4) that analyzes the findings, discussing the implications for algorithm selection in machine learning. The report concludes (Section 5) with a summary of what was learned and suggestions for further exploration.

2. Algorithms and Experimental Methods

The purpose of this project was to assess the effectiveness of non-parametric learning methods, specifically the k-Nearest Neighbors (KNN) algorithm and its variants, including Edited and Condensed Nearest Neighbors. Our implementation of these algorithms was designed to be highly adaptable, accommodating classification and regression and various experimental configurations and adaptations depending on the dataset in question. In this section, we will provide a brief overview of the datasets we used from the UCI Machine Learning Repository, the specific details of our algorithm implementations, and the methods we utilized to test our hypothesis.

2.1 Data Sets and Preprocessing

For this study we ran experiments on six different datasets from the UCI Machine Learning Repository, three classification datasets and three regression datasets. Since our goal was to compare the algorithms performance across these different datasets, we standardized the data preprocessing for all datasets as much as possible. This involved encoding all ordinal and nominal attributes, using z-score standardization to normalize the features, and imputing missing values using mean or mode for continuous and discrete features respectively. Some datasets required additional processing, such as the Forest Fire dataset, which had the target variable of Area heavily skewed towards zero. In this case, a log transformation was done on the target column to mitigate the skew effects during learning. Further background information on the datasets will be provided below.

2.1.1 Albalone Data Set

The Abalone dataset (7) is a regression dataset that serves to predict the age of abalone from physical measurements. Traditionally, age determination requires counting the shell rings under a microscope, a time-consuming process. This dataset, containing 4177 instances across 8 attributes, aims to simplify age prediction using more accessible measurements. All 8 attributes are continuous except for Sex (Nominal).

Data preprocessing for this dataset was standard as explained earlier. The Sex attribute was encoded using one-hot encoding and the remaining features were standardized using z-score standardization.

2.1.2 Forest Fires Data Set

The Forest fires dataset (9) is a regression dataset aimed at predicting the area of forest fires using meteorological and other relevant data. The dataset, which includes 517 instances and 13 attributes, provides a challenging regression task, particularly due to the skewness of the data towards smaller fires. Attributes such as the spatial coordinates (X, Y), time-related variables (month, day), and various weather conditions (e.g., temperature, relative humidity, wind speed, and rainfall) form the basis of the dataset.

In preparing the Forest fires dataset for analysis, standard data processing was applied. Categorical attributes (X, Y, month, and day) were treated as ordinal values and encoded into integers. All attributes were normalized using z-score standardization.

2.1.3 Computer Hardware Data Set

The hardware dataset (8) is a regression that contains 209 instances, each characterized by 10 attributes. It was created to model the relative CPU performance of computer systems, providing a benchmark for predicting the Estimated Relative Performance (ERP) from the hardware specifications. The dataset encompasses a range of attributes, including machine cycle time, minimum and maximum main memory, cache size, and the number of channels, among others. The goal attribute, PRP, is the actual relative performance, and the dataset includes ERP as an estimate from the original study. In terms of data preprocessing, Vendor Name and Model attributes were dropped as they were non-predictive. The remaining numeric features were standardized.

2.1.4 Congressional Vote Data Set

The 1984 United States Congressional Voting Records (6) is a classification dataset that includes the voting patterns of U.S. House of Representatives Congressmen during that session. The dataset includes 435 instances, each with 16 key votes as attributes. The goal is to use these attributes to predict the binary class party affiliation—Democrat or Republican—of each Congressman based on their votes. The dataset reduces the complexity of voting behaviors into a binary classification task, with 'yea' or 'nay' votes.

During the preprocessing phase, missing votes, denoted by "?", were addressed. These missing entries do not represent a lack of information but rather votes that are not straightfor-

ward 'yea' or 'nay', such as abstentions. These three voting outcomes were treated as nominal attributes and one-hot encoded during preprocessing.

2.1.5 Car Evaluation Data Set

The Car Evaluation dataset (5) is a classification dataset designed to assess the acceptability of cars based on a defined set of attributes. This dataset, consisting of 1728 instances, categorizes cars into four classes of acceptability: unacceptable, acceptable, good, and very good, based on six attributes including buying price, maintenance cost, number of doors, capacity in terms of persons to carry, the size of the luggage boot, and safety. In preparing the Car Evaluation dataset for analysis, all attributes were integer encoded and normalized.

2.1.6 Breast Cancer Data Set

The Breast Cancer dataset (4) is a classification dataset that aims to predict breast cancer malignancy through diagnostic imaging. Comprising 699 instances with 10 attributes each, this dataset records the characteristics of cell nuclei present in breast mass biopsies. Attributes include clump thickness, uniformity of cell size and shape, marginal adhesion, and others, all measured on a scale from 1 to 10. The primary goal is to classify tumor samples into the binary classes of benign or malignant.

The preprocessing of the Breast Cancer dataset involved addressing missing attribute values, denoted by "?", and ensuring that all numerical features were appropriately normalized for analysis.

2.2 Algorithms

2.2.1 K-Nearest Neighbors

Parameters: k, γ (Regression)
K-Nearest Neighbors (KNN) is a nonparametric algorithm that can generalize on unseen data through proximity-based reasoning. It makes predictions by calculating an unseen data point's proximity to each of the training instances through a distance metric. Once the distances are calculated, KNN ranks the k nearest neighbors and determines the new points class from the most common of the neighbors classes for classification or a weighted average of the neighbors classes for regression.

There are many choices for the distance metric but for this project, we used the Euclidean distance for numeric or encoded categorical values and the Value Difference metric for any non-encoded categorical values. The Euclidean distance $\mathbf{d}(\mathbf{p}, \mathbf{q})$ was calculated using the following formula:

$$d(\mathbf{p}, \mathbf{q}) = \sqrt{\sum_{i=1}^{n} (p_i - q_i)^2}$$

Where $\mathbf{p}$ and $\mathbf{q}$ represent two points in an n-dimensional Euclidean space.

The Value Difference Metric (VDM) is used to calculate distances between categorical values within the K-Nearest Neighbors (KNN). Given a dataset with n features, including

categorical ones, the VDM for two values a and b of a particular feature i, across a target class C with m classes, is defined as:

$$VDM_p(a,b) = \left(\sum_{j=1}^{m} \left| \frac{N_{a,j}}{N_a} - \frac{N_{b,j}}{N_b} \right|^p \right)^{\frac{1}{p}}$$

where:

- $N_{a,j}$ is the count of instances where feature i has value a and belongs to class j.
- N_a is the total count of instances where feature i has value a.
- $N_{b,j}$ is the count of instances where feature i has value b and belongs to class j.
- N_b is the total count of instances where feature i has value b.
- p is a parameter that sets the power of the metric, typically chosen as 2.

To compute the distance between two instances X and Y, the algorithm integrates the Euclidean distance for numerical features with the VDM for categorical features:

$$distance(X,Y) = \sqrt{\sum_{i \in numeric} (X_i - Y_i)^2 + \sum_{i \in categorical} VDM_p(X_i, Y_i)}$$

This method enables the KNN algorithm to process datasets with mixed data types. As stated earlier, for classification tasks, the predicted class of an instance is simply the most common class of the k nearest neighbors. However, for regression, we use a Gaussian Kernel to calculate a weighted average of the k nearest neighbor classes to generate a predicted class value. The weight for the i-th nearest neighbor is calculated as:

$$w_i = \exp\left(-\gamma \cdot d_i^2\right)$$

where:

- w_i is the weight of the i-th nearest neighbor.
- γ is the bandwidth parameter for the Gaussian kernel, controlling the rate of decay for the weights with respect to the distance d_i.
- d_i is the distance between the query instance and the i-th nearest neighbor.

The predicted value ($\hat{y}$) for the query instance is then obtained by calculating the weighted average of the target values (y_i) of the k nearest neighbors:

$$\hat{y} = \frac{\sum_{i=1}^{k} w_i \cdot y_i}{\sum_{i=1}^{k} w_i}$$

2.2.2 Edited K-Nearest Neighbors

Parameters: k, γ (Regression), ϵ (Regression)

The Edited K-Nearest Neighbors (Edited KNN) approach modifies the training dataset by removing instances that are incorrectly classified by their nearest neighbors, aiming to enhance the model's generalization ability (2). In our implementation, the algorithm for classification iterates through each instance in the training set, assessing its classification based on the k nearest neighbors within a temporarily reduced set (excluding the instance under consideration). Instances leading to misclassification are systematically removed, with the process iterating until no further improvements can be made, as measured by a loss metric on a separate validation data set. For regression, instances with predictions deviating beyond a specified threshold (ϵ) are considered outliers or noise and are removed from the training set. The objective is to generate a smaller, representative subset of the original training set, reducing noise and computational complexity.

2.2.3 Condensed K-Nearest Neighbors

Parameters: k, γ (Regression), ϵ (Regression)

The Condensed K-Nearest Neighbors (Condensed KNN) algorithm aims to reduce the training dataset by only retaining instances that are critical for the model's decision-making process (3). For classification, our implementation starts with a randomly chosen instance from the training set to form the initial condensed set. It then iterates over the remainder of the training set, incorporating any instance that is misclassified by the current condensed set. This process continues until no further additions are made, resulting in a compact subset of the original training data.

In regression, the algorithm follows a similar process, utilizing a threshold, ϵ, to determine significant deviations between the predicted and actual values. Instances that lead to a prediction error greater than ϵ are added to the condensed set.

2.3 Experiment Design

The experiments were designed to compare the performance of K-Nearest Neighbors (KNN) and it's variants against a null model, hypothesizing that Edited and Condensed KNN would not offer significant improvements in performance. The experimental setup was as follows:

1. Data was split into 80% for training and 20% for validation.
2. Hyperparameter tuning was conducted through 5x2 cross-validation, involving:
    (a) Splitting the training data into two equal parts, using stratification in classification to maintain class balance.
    (b) Modifying the training set in Edited and Condensed KNN each iteration.
    (c) Evaluating two models per hyperparameter set against the 20% validation set.

3. Averaging results over ten runs to select optimal hyperparameters.

4. Sequentially tuning k for classification and k, γ, and ϵ for regression.

5. Evaluating average performance with the optimal parameters through another round of 5x2 cross-validation.

6. The final performance metrics include MSE, MAE, R2, and Pearson for regression, and 0-1 Loss, Precision, Recall, and F1 for classification.

For all KNN variations and the null models, we calculate Mean Squared Error (MSE), Mean Absolute Error (MAE), R-squared (R2), and Pearson's correlation for regression, and 0-1 Loss, Precision, Recall, and F1 score for classification.

3. Results

This section presents the experimental results obtained from evaluating the K-Nearest Neighbors (KNN) algorithm and its variants (Edited KNN and Condensed KNN) across different datasets for both regression and classification tasks. The performance of these models is compared against null models to provide a baseline for evaluation.

Table 1 displays the aggregated performance metrics for regression tasks. For these experiments, the KNN models were evaluated with hyperparameters optimized through a validation process, with k set dynamically based on the dataset characteristics. Table 2 outlines the performance of the KNN models in classification tasks. Similar to the regression tasks, the models' hyperparameters were fine-tuned based on preliminary validation experiments.

The figure 1 displays the percent improvement in performance from the null model to the KNN, Edited KNN, and Condensed KNN models across both regression and classification datasets. MAE and Zero-One Loss were used for regression and classification performance respectively The horizontal lines are drawn to show the average percent improvement for regression and classification datasets separately.

Table 1: Regression Task Performance Comparison

Dataset	**Algorithm**	**MSE**	**MAE**	**R2 Score**	**Pearson Score**
Machine Data	Null Model	19073.87	86.17	N/A	N/A
	KNN	7512.19	34.98	0.6524	0.8250
	Edited KNN	7120.23	33.98	0.6787	0.8429
	Condensed KNN	7984.20	39.76	0.5983	0.7956
Abalone Data	Null Model	10.281	2.358	N/A	N/A
	KNN	5.109	1.570	0.5032	0.7136
	Edited KNN	5.283	1.576	0.4863	0.7070
	Condensed KNN	5.187	1.626	0.4953	0.7051
Forest Fire Data	Null Model	2161.23	11.25	N/A	N/A
	KNN	2158.38	11.38	-0.0462	0.0170
	Edited KNN	2171.83	11.23	-0.0587	-0.0208
	Condensed KNN	2149.80	11.74	-0.0404	0.0150

Table 2: Classification Task Performance Comparison

Dataset	Algorithm	0-1 Loss	Precision	Recall	F1 Score
Car Data	Null Model	0.2945	0.4977	0.7055	0.5837
	KNN	0.0815	0.9187	0.9185	0.9150
	Edited KNN	0.1327	0.8595	0.8673	0.8559
	Condensed KNN	0.0823	0.9211	0.9177	0.9167
House Votes Data	Null Model	0.3937	0.3676	0.6063	0.4577
	KNN	0.0649	0.9367	0.9351	0.9353
	Edited KNN	0.0753	0.9319	0.9247	0.9254
	Condensed KNN	0.0695	0.9319	0.9305	0.9306
Breast Cancer Data	Null Model	0.3506	0.4217	0.6494	0.5113
	KNN	0.0358	0.9645	0.9642	0.9641
	Edited KNN	0.0408	0.9594	0.9592	0.9591
	Condensed KNN	0.0390	0.9615	0.9610	0.9609

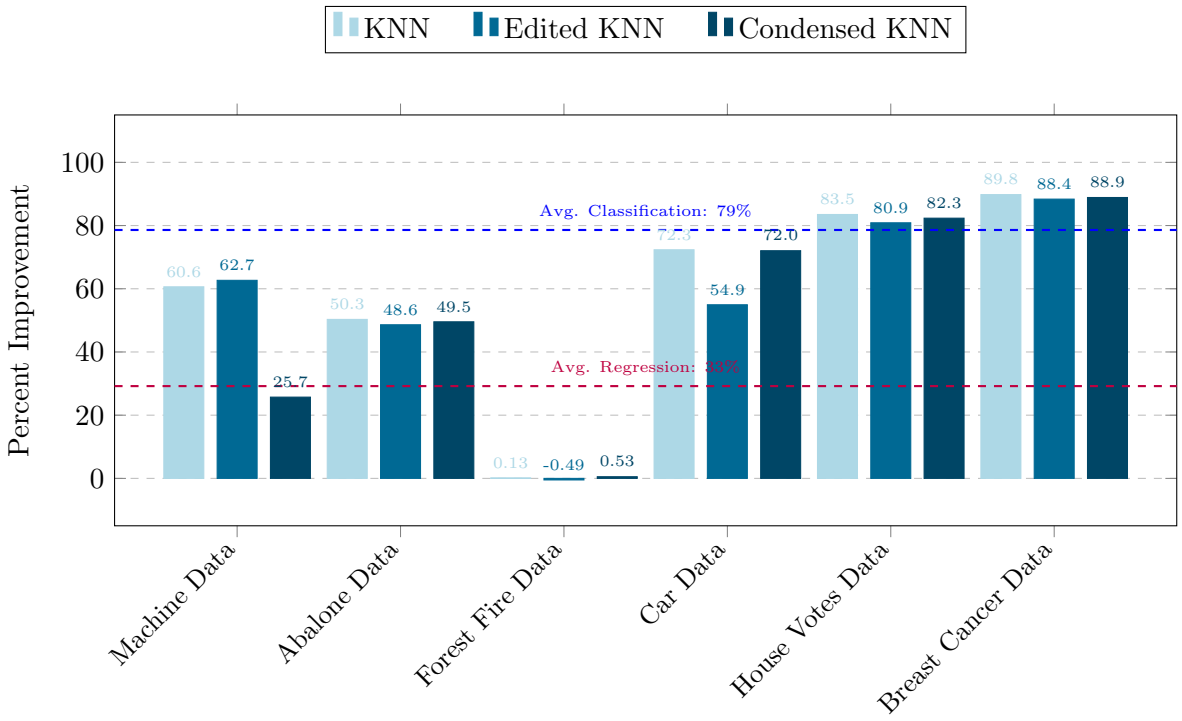

Figure 1: Percent improvement from the null model to KNN, Edited KNN, and Condensed KNN across classification and regression datasets.

4. Discussion

In the results presented above, the performance of k-Nearest Neighbors (KNN), Edited KNN, and Condensed KNN, show interesting patterns across both classification and regression tasks. For one, our algorithms demonstrated substantial ability in classification tasks, significantly outperforming null models with an average reduction of zero-one loss by 79%. This shows

the adaptability of KNN-based methods to classification problems, where their use of local similarity provides a strong mechanism for predicting class.

However, the regression task results are more inconsistent. Despite improvements over null models, the performance metrics—Mean Squared Error (MSE), Mean Absolute Error (MAE), R-squared (R2), and Pearson score—suggest that KNN and its variants face challenges in accurately predicting continuous classes. This is particularly evident in the Forest Fire dataset, where the KNN models barely preform better than null model, or in the case of Edited KNN, perform worse as shown in Figure 1. This outcome likely stems from the sparse distribution of data points and the high-dimensional feature spaces.

Model performance on these datasets was closely related to the class complexity. The KNN models, unsurprisingly, performed best on the Breast Cancer dataset which has a binary class space of Malignant or Benign. Average performance was then followed by the House Votes dataset which had three class options, followed by the Car Dataset with four. Once these classes change from discrete to continuous the effects of complexity and sparsity of data get even more pronounced as shown by the performance in regression tasks.

The Edited and Condensed KNN variants did not perform better than traditional KNN in most cases. Their performance was almost identical to that of traditional KNN which can be seen as a positive given their improvements in space complexity. The performance results on the Machine dataset are particularly interesting as this was the sparsest dataset with only 209 instances. The Edited KNN model's implementation of a stop condition based on performance degradation allowed it to reduce the dataset enough to remove outliers while still improving performance. However, the Condensed KNN algorithm, which only uses the minimal dataset required to maintain the decision boundaries appears to have reduced the dataset too far, eliminating critical data to performance.

These observations support our hypothesis and highlight the importance of dataset characteristics in algorithm performance. Edited KNN and Condensed KNN variants both show potential in specific contexts. However, their effectiveness is closely tied to the nature of the dataset, with certain characteristics such as size, dimensionality, and class complexity playing important roles.

5. Conclusion

This study was able to demonstrate the effectiveness of KNN models in generalizing on a wide variety of datasets containing a complex collection of numerical and categorical data. With a fairly standard data processing pipeline and model configuration using Euclidean distance calculations, we were able to achieve 95%+ accuracy in some of the classification tasks. It also demonstrated the strengths of Edited and Condensed KNN variations that achieve similar performance to KNN with reduced memory requirements.

Given more time it would be interesting to continue to explore the modularity of KNN algorithms to try and achieve better performance, especially on regression tasks that suffer from data sparsity. Specifically, experimenting with other distance metrics like Manhattan, Minkowski, and the many others could provide further performance gains. Other future work could also include experiments with different kernel functions as well as alternate hyperparameter tuning techniques.

References

[1] Evelyn Fix and J. L. Hodges, *Discriminatory Analysis. Nonparametric Discrimination: Consistency Properties*, International Statistical Review / Revue Internationale de Statistique, vol. 57, no. 3, 1989, pp. 238–247, JSTOR, https://doi.org/10.2307/1403797. Accessed 19 Feb. 2024.

[2] Dennis L. Wilson, "Asymptotic Properties of Nearest Neighbor Rules Using Edited Data," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. SMC-2, no. 3, pp. 408–421, 1972.

[3] P. Hart, "The condensed nearest neighbor rule (Corresp.)," *IEEE Transactions on Information Theory*, vol. 14, no. 3, pp. 515–516, 1968.

[4] University of Wisconsin, 1993.
Breast Cancer Wisconsin (Original).
Available: https://archive.ics.uci.edu/ml/datasets/Breast+Cancer+Wisconsin+%28Original%29.

[5] Jozef Stefan Institute, Yugoslavia (Slovenia), 1988.
Car Evaluation.
Available: https://archive.ics.uci.edu/ml/datasets/Car+Evaluation.

[6] University of California, Irvine, 1987.
Congressional Voting Records.
Available: https://archive.ics.uci.edu/ml/datasets/Congressional+Voting+Records.
Notes: "?" does not indicate a missing attribute value but means "abstain".

[7] Marine Research Laboratories, Tasmania, 1995.
Abalone.
Available: https://archive.ics.uci.edu/ml/datasets/Abalone.

[8] Tel Aviv University, Israel, 1987.
Computer Hardware.
Available: https://archive.ics.uci.edu/ml/datasets/Computer+Hardware.

[9] University of Minho, Portugal, 2007.
Forest Fires.
Available: https://archive.ics.uci.edu/ml/datasets/Forest+Fires.