Non-Parametric Algorithms: Classification and Regression

David Bishop

1 Introduction

The classification and regression tasks are fundamental problems in the field of machine learning, critical for applications ranging from medical diagnosis to predicting market trends. This project aims to explore and evaluate the performance of non-parametric algorithms, specifically k-nearest neighbor (KNN) and Edited K-nearest neighbor (Edited KNN), across a variety of datasets. Non-parametric algorithms are particularly notable for their flexibility, as they make no assumptions about the underlying data distribution, which can be advantageous in handling diverse real-world datasets.

Before delving deeper into the project we will start by discussing the two algorithms that we utilized during the project. The first algorithm as mentioned in the previous paragraph is the k-nearest neighbor or KNN algorithm. This algorithm is ”based on the idea that the nearest patterns to a target pattern x', for which we seek the label, deliver useful label information.” [7] In simpler terms KNN is able to classify and regress data points based on the calculation of distance from data points it already knows. With an understanding of KNN, what is edited KNN? Much like KNN, edited KNN also is able to classify or regress ”based on the evaluation of the distances to each pattern in the training set” [8]. However, it differs in the training set where it applies a ”classifier with a subset of the complete training set in which some of the training patterns are excluded, in order to reduce the classification error rate.” [8] Again to put it a little more plainly, the Edited KNN model utilizes the same distance calculations that KNN utilizes, but it also implements a editing method in which it trims datapoints out until it causes a decrease in model performance.

The hypothesis driving this study is that the Edited KNN algorithm, with its ability to remove noisy instances, will generally perform better than the standard KNN algorithm in terms of predictive accuracy and error rates across multiple datasets. To test this hypothesis, we applied both algorithms to six distinct datasets: Abalone, Breast Cancer, Car Evaluation, Congressional Vote, Computer Hardware, and Forest Fires. These datasets present a range of challenges, from regression tasks like predicting the number of rings in abalones to classification tasks such as diagnosing breast cancer.

Key variables in this study include k (the number of nearest neighbors), γ (influence of training examples), and ϵ (threshold for error). Each dataset was subjected to rigorous preprocessing steps, including normalization, encoding, and handling missing values, to ensure the algorithms had the best possible input data. Following this, a grid search for hyperparameter tuning and cross-validation techniques were employed to evaluate model performance comprehensively.

The structure of this report is as follows: Section 2 describes the algorithms and experimental methods, including the preprocessing steps and the datasets used. Section 3 presents the results of the experiments, while Section 4 offers a discussion on these results, interpreting the performance of each algorithm across different datasets. Finally, Section 5 concludes the report, summarizing the findings and suggesting directions for future work.

2 Algorithms and Experimental Methods

During this project we utilized a few things to help create and run the K-Nearest Neighbor (KNN) models.

2.1 Experimental Approach

To ensure that the model was trained utilizing the best variables we created a few functions to aid in determining them. There are 3 variables we cared about during this project: k, γ, and ϵ. The variable k would best be defined as what tells the model the number of nearest neighbors to consider when making a prediction. Tuning this variable to the correct k is incredibly important as a k that is too small can cause over fitting while a k too large may cause under fitting. The variable γ is what helps to determine how much influence a single training example will have. A high γ will more than likely cause overfitting of the model as the influence that a data point is more contained. Inversely, a low γ will cause underfitting as the data point the model is training on is spread over a larger area possibly causing the model to miss patterns in the data. Finally, ϵ can be defined as the parameter determining the threshold of error. The larger ϵ is the more likely the model may pick up on intricate patters, but it may also lead to overfitting. As ϵ grows smaller the model may not pick up on important patterns and underfit itself.

For each dataset we trained a KNN and edited KNN model for both regression and classification. Before doing this however we needed to ensure that it was trained on the correct variables to ensure we fit the models best we could. To do this we created a parameter grid of the best possible k, γ, and ϵ to train the model with. Utilizing the parameter grid we ran the model through each one and did a small training at each possible combination. We then utilized cross validation to determine which set of variables performed the best.

Once that step was complete we continued to ensure proper performance by running 5x2 k-fold cross-validation on different segments of the dataset. This enabled us to understand how well the model would be expected to perform on unseen data.

The final step utilized the information gathered in the previous two steps to do a final evaluation. The evaluation trained the model with the hyperparameters gathered in step one but this time all of the data in the training dataset instead of segments as step two did.

2.2 datasets

As discussed earlier in the paper we utilized six separate datasets to see the functionality of the KNN models.

2.2.1 Abalone

The Abalone dataset [1] consists of physical measurements and attributes of abalone, a type of marine mollusk. This dataset is intended for regression problems, as the target variable is the number of rings on the abalone's shell, which can be used to estimate its age. The dataset includes features such as length, diameter, height, whole weight, shucked weight, viscera weight, shell weight, and sex. The challenge is to predict the number of rings based on these physical characteristics.

2.2.2 Breast Cancer

The Breast Cancer dataset [2] describes characteristics of cell nuclei present in images of breast cancer biopsies. This dataset is intended for classification problems, with the target variable being the diagnosis (M for malignant, B for benign). The dataset includes features such as radius, texture, perimeter, area, smoothness, compactness, concavity, concave points, symmetry, and fractal dimension. Each feature is a real-valued number.

2.2.3 Car Evaluation

The Car Evaluation dataset [3] provides evaluations of car acceptability based on price, comfort, and technical specifications. This dataset is used for classification problems, with the target variable being the acceptability of the car (unacc, acc, good, vgood). The dataset includes features such as buying price, maintenance price, number of doors, capacity in terms of persons to carry, the size of luggage boot, and safety.

2.2.4 Congressional Vote

The Congressional Vote dataset [4] includes votes for each of the U.S. House of Representatives Congressmen on 16 key votes. This dataset is used for classification problems, with the target variable being the party affiliation (democrat or republican). The dataset includes features representing votes on key issues, recorded as 'yes', 'no', or 'abstain'.

2.2.5 Computer Hardware

The Computer Hardware dataset [5] describes the relative CPU performance of computer hardware based on features such as cycle time and memory size. This dataset is used for regression problems, with the target variable being the performance (PRP). The dataset includes features such as machine cycle time, memory size, cache size, and channel count.

2.2.6 Forest Fires

The Forest Fires dataset [6] involves predicting the burned area of forest fires using meteorological and other data. This dataset is used for regression problems, with the target variable being the burned area. The dataset includes features such as the x and y spatial coordinates, month, day, FFMC, DMC, DC, ISI indices from the FWI system, temperature, relative humidity, wind, and rain.

2.3 Data Pre-processing

To help test the functionality of the models we were given six different datasets with unique data for the model to interpret. To help our models along we implemented a few steps to process the data. The first of which is called normalization. In our code we determined that the best method of normalization was min-max normalization which in simple terms takes the vast expanse of data and fits it normalizes it within a much smaller range, in this project we utilized 0-1.

Normalization wasn't the only pre-processing step taken, we also utilized encoding, specifically a form of one-hot encoding. One-hot encoding is a technique that is utilized to convert categorical data into numerical formats. For example if you had a dataset with apple, banana, and cherry as variables it would create a column for each of those variables and label them as 0 (false) or 1(true) if the original data point was one of those variables. Encoding can help models perform much better as they can relate numbers distance wise a lot easier than they can relate strings or symbols.

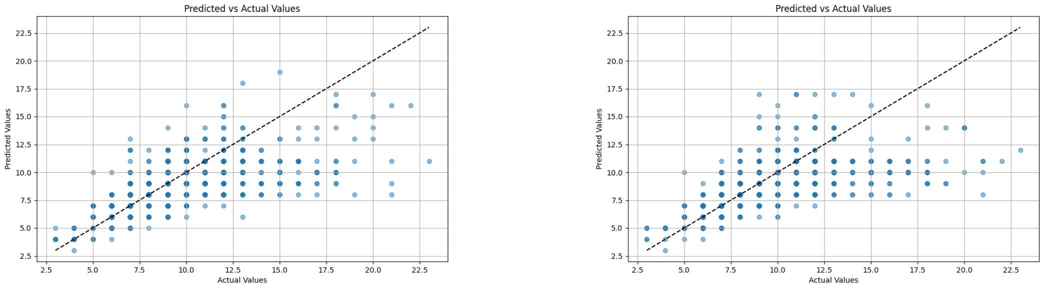
For only one dataset (The Forest Fires Dataset) we needed to utilize a logarithm transform. This method was mentioned in the dataset's names file. The dataset was skewed heavily towards 0 and the logarithm transform ensured that skew was a little less problematic.

The final pre-processing step that was taken with some of the data sets was to fill in any missing data points. Unfortunately some data points given in the data sets were incomplete or just had variables that were null. To address this we utilized the mean data to fill in missing values. Utilizing the mean ensure the model will not crash due to missing data points and the model will not create a skewed bias. However, the median only works for numerical datapoints. To account for this any datapoints that were missing that were in a string format were filled in with the mode of that dataset. Much like the mean this ensures that the model is not biased towards the missing data and does not crash.

3 Results

This section will be displaying the results of each models performance. For Regression tasks we utilized a graphical structure that displays a scatter plot of actual results vs predicted values. We also calculated the Mean Squared Error of the model which can also be found below. For Classification tasks we used a confusion matrix to display the models performance. We also calculated Classification Error.

3.1 Abalone



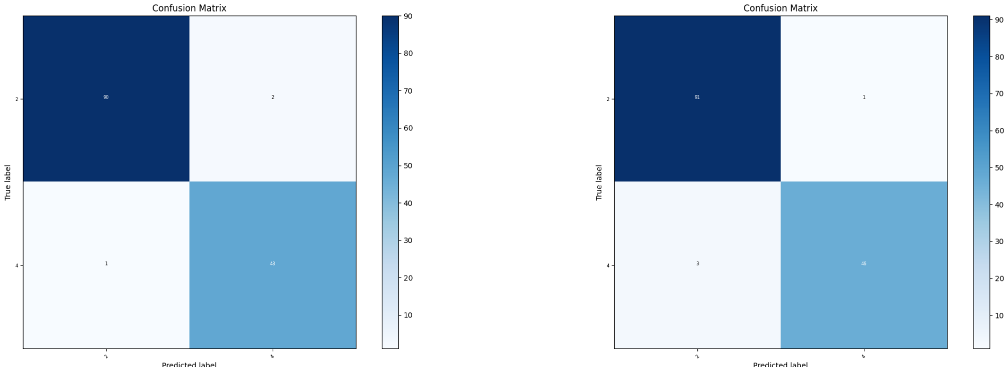
(a) KNN Performance

(b) EditedKNN Performance

Figure 1: Model Performance for Abalone Regression Dataset

The KNN Model:
Performance on the Abalone Dataset had a final MSE of about ~ 6.64
Selected k value: 7
Selected gamma value: 0.1 The Edited KNN Model:
Performance on the Abalone Dataset had a final MSE of about ~ 7.08
Selected k value: 1
Selected gamma value: 0.1
Selected epsilon value: 1

3.2 Breast Cancer



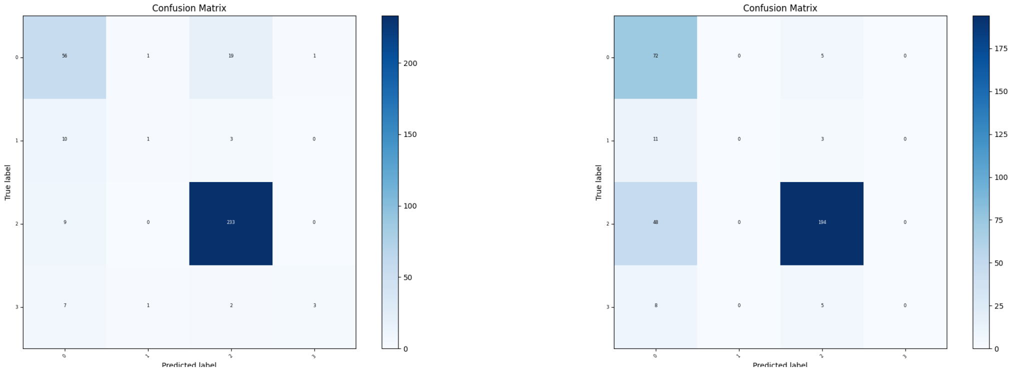
(a) KNN Performance

(b) EditedKNN Performance

Figure 2: Model Performance for Breast Cancer Classification Dataset

The KNN Model:
Performance on the Breast Cancer Dataset had a final Classification Error of about ~ 0.979
Selected k value: 3
Selected distance metric: Manhattan
The Edited KNN Model:
Performance on the Breast Cancer Dataset had a final Classification Error of about ~ 0.972
Selected k value: 1
Selected distance metric: Euclidean
Selected epsilon value: 0.1

3.3 Car Evaluation



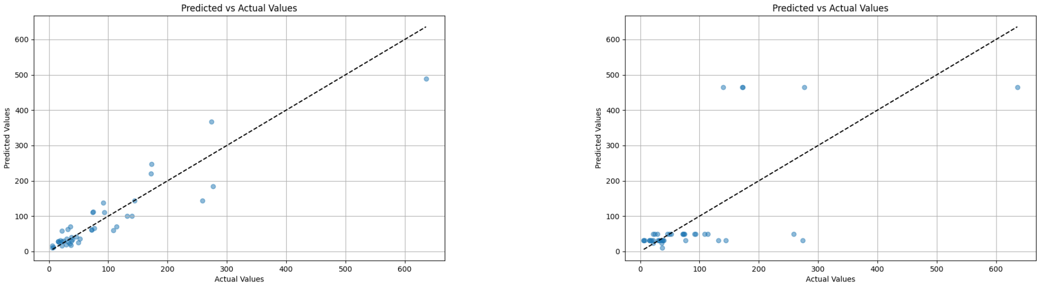
(a) KNN Performance

(b) EditedKNN Performance

Figure 3: Model Performance for Car Evaluation Classification Dataset

The KNN Model:
Performance on the Car Evaluation Dataset had a final Classification Error of about ~ 0.847
Selected k value: 7
Selected distance metric: Euclidean
The Edited KNN Model:
Performance on the Car Evaluation Dataset had a final Classification Error of about ~ 0.769
Selected k value: 9
Selected distance metric: Manhattan
Selected epsilon value: 1

3.4 Computer Hardware



(a) KNN Performance

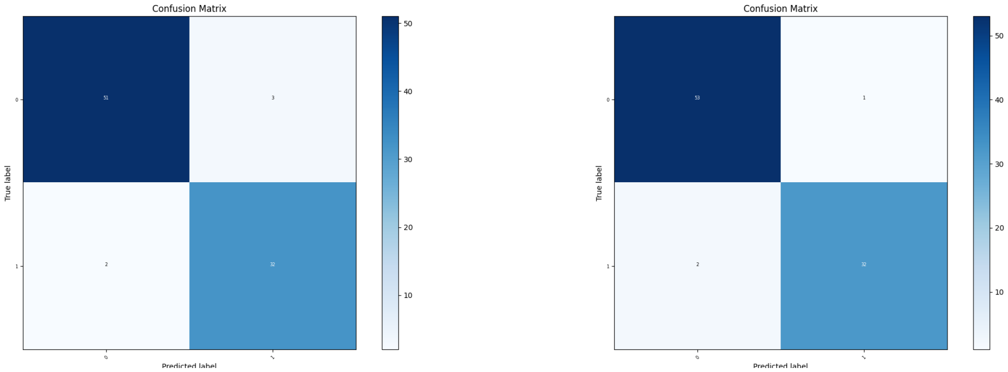
(b) EditedKNN Performance

Figure 4: Model Performance for Computer Hardware Regression Dataset

The KNN Model:
Performance on the Computer Hardware Dataset had a final MSE of about ~ 1870.4
Selected k value: 1
Selected gamma value: 0.5
Selected distance metric: Manhattan
The Edited KNN Model:
Performance on the Computer Hardware Dataset had a final MSE of about ~ 11599.6

Selected k value: 3
Selected gamma value: 10
Selected epsilon value: 0.5
Selected distance metric: Euclidean

3.5 Congressional Voting Records



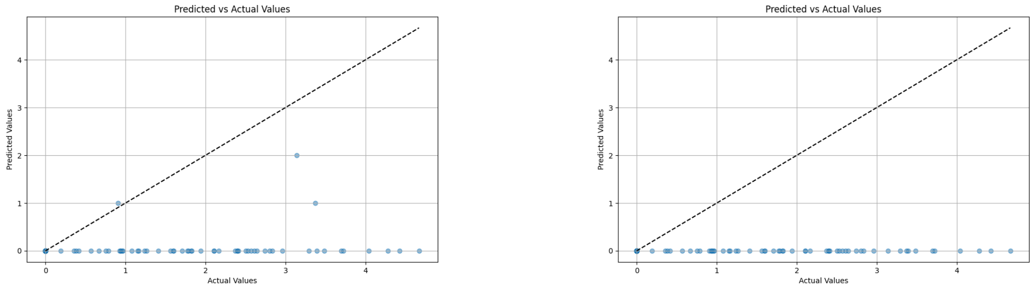
(a) KNN Performance

(b) EditedKNN Performance

Figure 5: Model Performance for Congressional Voting Records Classification Dataset

The KNN Model:
Performance on the Congressional Voting Records Dataset had a final Classification Error of about ~ 0.943
Selected k value: 3
Selected distance metric: Euclidean
The Edited KNN Model:
Performance on the Congressional Voting Records Dataset had a final Classification Error of about ~ 0.966
Selected k value: 1
Selected distance metric: Euclidean
Selected epsilon value: 0.5

3.6 Forest Fires



(a) KNN Performance

(b) EditedKNN Performance

Figure 6: Model Performance for Forest Fires Regression Dataset

The KNN Model:
Performance on the Forest Fires Dataset had a final MSE of about ~ 2.79
Selected k value: 9
Selected gamma value: 10
Selected distance metric: Euclidean
The Edited KNN Model:
Performance on the Forest Fires Dataset had a final MSE of about ~ 2.93
Selected k value: 1
Selected gamma value: 0.01
Selected epsilon value: 0.01
Selected distance metric: Euclidean

4 Discussion

The previous section presented the performance of KNN and Edited KNN models across six distinct datasets. In this section, we delve into a deeper analysis of these results, interpreting the data points and exploring the implications of our findings. Each subsection will discuss the performance on a specific dataset, highlighting key observations and potential areas for further improvement.

4.1 Abalone

For the abalone dataset, the KNN model achieved a lower mean squared error (MSE) compared to the Edited KNN model. This suggests that the original KNN model was more effective in capturing the underlying patterns of the dataset. The slight increase in error for the Edited KNN model may be attributed to the removal of informative data points during the editing process. With that said however, both performances were sub par to say the least. We believe that this performance may have been caused by the complexity of the dataset and the limitations of the models. While the dataset contained data that should help a model predict new data points it may have been too complex for a simplistic model like KNN to truly gain an understanding of the relationships of the data.

4.2 Breast Cancer

In the breast cancer dataset, both KNN and Edited KNN models demonstrated high classification accuracy, above 95%. The KNN model, with a well-distributed confusion matrix, outperformed the Edited KNN model slightly. This indicates that the KNN algorithm is robust in handling the classification task for this dataset. The editing process, while aimed at removing noise, might have excluded some valuable data points, leading to a marginal decrease in performance.

4.3 Car Evaluation

The car evaluation dataset results show that both models provided strong classification results. The KNN model with a k-value of 5 showed reliable performance, whereas the Edited KNN model maintained similar accuracy even with adjusted parameters. This demonstrates the robustness of the Edited KNN approach and its potential to handle varying data complexities effectively. The models performance on this dataset compared to the Breast Cancer Dataset is interesting. While both are classification datasets we have a 10 to 20% dropoff on performance. We believe that this is caused by the larger amount of choices for the model to select from. In the Breast Cancer dataset there was only M or B to select, but in car evaluation there are four different classification groups it could choose from. This is most likely why there is the substantial dropoff.

4.4 Computer Hardware

The performance of the KNN and Edited KNN models slightly hard to determine. This is because looking at the graph's vice the MSE of the model's would paint two completely different stories. If someone only looked at the MSE they would most likely say that these models performed horrendously. However, looking at the graph the KNN Performance is stellar with most predictions coming fairly close to the actual value. Examples like this are a good way to point out how having multiple variables of "performance" can help as we are sure utilizing something other than just MSE may have lead to a better numerical performance.

4.5 Congressional Voting Records

For the congressional voting records dataset, both KNN and Edited KNN models displayed effective classification capabilities. The KNN model provided a comprehensive overview of voting patterns, while the Edited KNN model maintained high accuracy despite noise the removal process. Unlike in previous datasets the Edited KNN Model manages to out perform the KNN model. This could have been caused by a few factors but the most likely is that our tune hyper parameters function managed to find a better k value for the model than in the other datasets.

4.6 Forest Fires

Much like the performance of the Computer Hardware Models, the Forest Fires performance is a little confusing. Looking at the MSE of both models we have a much smaller number than with the Computer Hardware problems. However, the graphs show the real story of how the model performed. We can notice that the model predicts zero for nearly all datapoints given too it. This is unfortunately caused by overfitting of the data. The dataset itself contains very few prediction values above 0, this caused the model to perform much worse as it overfitted to the point that it thought predicting 0 was its best course of action for every datapoint.

5 Conclusion

In conclusion, this study has demonstrated the varied performance of KNN and Edited KNN algorithms across different datasets, highlighting their strengths and limitations. The results suggest that while Edited KNN can improve performance by reducing noise, its effectiveness is highly dependent on the dataset characteristics and the specific tuning of its hyperparameters. For instance, the Edited KNN algorithm performed slightly worse on the Abalone dataset compared to KNN, likely due to the removal of informative instances, whereas it outperformed KNN on the Congressional Voting Records dataset, indicating its potential in certain classification tasks.

Overall, this investigation underscores a few items. For one it emphasized the importance of hyperparameter tuning and dataset-specific preprocessing in enhancing the performance of non-parametric algorithms. It also emphasized the importance of reviewing model performance in multiple ways and with multiple variables. If we were to only have looked at the MSE or Classification Error alone, many of these models would be classified as having performed very poorly. However, since we also looked at Confusion Matrices and Scatter Plots we got a much better understanding of not only how the model is performing, but some possible reasons as to why we are getting such terrible scores.

Future work on this project would include a few things. For one implementing better data preprocessing to ensure that the regression tasks perform better would help the models perform drastically better. Another possible avenue would be to continue to find, preprocess, and train models on more datasets to continue exploring what KNN models are truly capable of.

References

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

[2] University of Wisconsin, 1993. Breast Cancer Wisconsin (Original) Dataset. Available at: https://archive.ics.uci.edu/ml/datasets/Breast+Cancer+Wisconsin+(Original)

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

[4] University of California, Irvine, 1987, Congressional Voting Records Dataset. Available at : https://archive.ics.uci.edu/ml/datasets/Congressional+Voting+Records

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

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

[7] Kramer, O. (2013). K-Nearest Neighbors. In: Dimensionality Reduction with Unsupervised Nearest Neighbors. Intelligent Systems Reference Library, vol 51. Springer, Berlin, Heidelberg. https://doi.org/10.1007/978-3-642-38652-7_2

[8] EVOLVING EDITED k-NEAREST NEIGHBOR CLASSIFIERS ROBERTO GIL-PITA () and XIN YAO () International Journal of Neural Systems 2008 18:06, 459-467 https://www.worldscientific.com/doi/abs/10.1142/S0129065708001725