Comparative Analysis of Supervised Learning Algorithms on Binary Classification Datasets

*Marcelo Delgado, †Isaac Madrigal, ‡José Antonio Mora and §Mariela Valerio
School of Computer Science
Universidad de Costa Rica, San Jose, Costa Rica
*marcelo.delgadomora@ucr.ac.cr
†isaac.madrigalsilva@ucr.ac.cr
‡jose.moramonge21@ucr.ac.cr
§ mariela.valerio@ucr.ac.cr

***Abstract*—Machine learning has become an integral part of modern Artificial Intelligence, offering solutions to complex decision-making tasks through different algorithms. This paper explores and compares the performance of four supervised learning models —Logistic Regression, Decision Trees, k-Nearest Neighbors (kNN), and Neural Networks— on two binary classification datasets one containing Red Wine Quality related data and the other containing Water Potability data. The Red Wine dataset, characterized by a balanced class distribution and stronger feature correlations, allowed the models to achieve higher performance, with kNN emerging as the top performer. In contrast, the Water Potability dataset presented additional challenges like weak feature correlations and a slight class imbalance resulting in a necessity for extensive preprocessing to better model performance. While kNN again demonstrated superior results, neural networks provided a close alternative, at the cost of higher computational requirements.**

This study highlights the importance of informed model selection and preprocessing strategies based on the dataset characteristics and necessities. The results suggest that while kNN is highly effective across diverse datasets, neural networks may be more suitable for capturing complex, nonlinear patterns in data. These insights contribute to a better understanding of model applicability and optimization for binary classification tasks.

I. Introduction

Machine learning is a critical branch of Artificial Intelligence and its implementation has been modified and refined over the years via different concepts and different models with better performance as a goal, one of the possible implementations of machine learning is called supervised learning, which involves training an algorithm to make predictions based on labeled datasets. This approach is usually applied in classification tasks where the objective is to categorize data into different classes helping in decision-making in different areas of day-to-day life where data is involved. This study aims to compare the effectiveness of different models in binary classification by comparing their results between two different datasets.

The first dataset put to test in this study is a **Red Wine Quality** dataset which includes different properties of red wine samples, the general task related to this dataset is the classification of the wine between good and bad based on these features [1]. With comparison in mind, a second dataset was selected, in this case, the data is focused on classifying **water potability** based on various indicators which will be explained in later sections of this paper [2]. With the objective of comparison in mind, for this study, four machine-learning algorithms **–Logistic Regression, Decision Trees, k-Nearest Neighbors, and Neural Networks–** are trained from training data from both datasets individually and then tested, where each model performance is assessed using different metrics and then compared.

II. Datasets

A. Red Wine Quality Dataset

1) About the Dataset: This dataset consists of 12 columns and 1,599 rows, with each row representing an individual wine entry. Of these 12 columns, 11 are independent variables that describe different wine properties, all of which are of floating-point type. The remaining column is the dependent variable, representing wine quality, and is of integer type. Additionally, this dataset contains no null values.

2) Correlation Analysis with Heatmap: To analyze the relationships between variables, a heatmap was generated to show the correlation between all variables, aiming to identify which are more or less correlated with wine quality:

The values in Figure 1 range from -1 to 1. The closer the value is to 1, the stronger the positive correlation (when one variable increases, the other also increases). For example, alcohol shows the highest positive correlation with quality, with a value of 0.48, suggesting that a higher alcohol content tends to be associated with higher wine quality. Conversely, values close to -1 indicate a negative correlation (when one variable increases, the other decreases). Finally, values near 0 suggest no linear correlation between the variables, implying they likely do not influence quality.

B. Water Quality Dataset

1) About the Dataset: The second dataset consists of 10 columns and 3,276 rows, where each row represents a distinct body of water. Of the 10 columns, 9 are independent variables that describe different properties of water, where all of them are of floating-point type. The remaining column corresponds to the dependent variable or the label, representing the potability of the body of water, and is of integer type. All variables

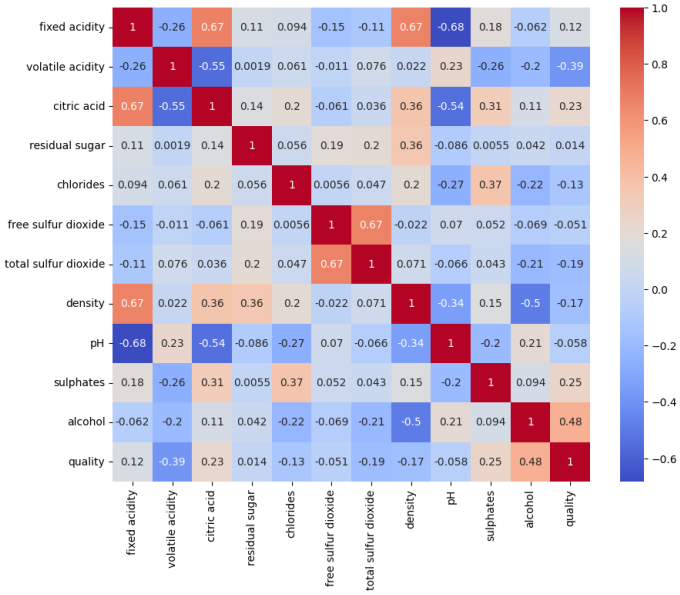

Fig. 1. Correlation Between Wine Quality Variables

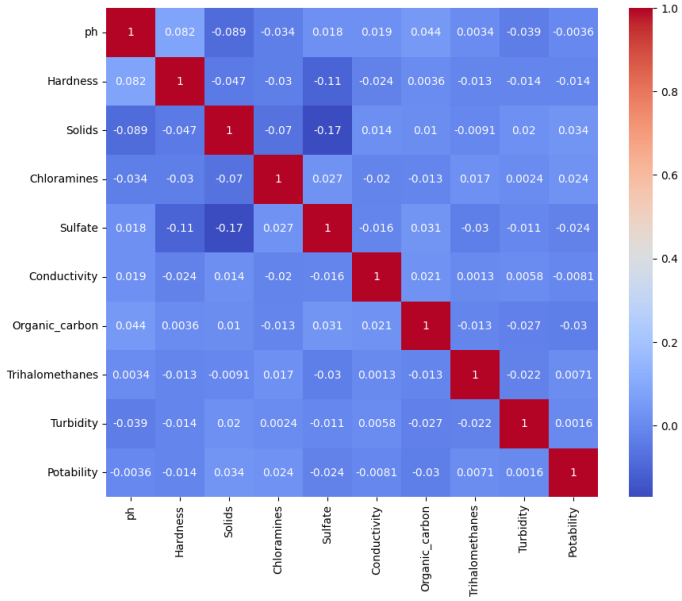

Fig. 2. Correlation Between Water Potability Variables

show a normal distribution for their values. On the other hand, the dataset contains null values, has a wide range of possible values between variables, and has a slight imbalance, so additional measures have to be taken. This particular dataset was chosen because it has a small extension compared to other datasets, and contains a manageable amount of features or variables, making it easy to compare and contrast to the wine quality dataset, without having to to extensive or overly complicated feature engineering or exploratory data analysis.

2) Correlation Analysis with Heatmap: A heatmap was generated to show the relationship and correlation between variables, to identify if there are variables with low correlation with water potability:

The values in Figure 2 follow the same principles as the heatmap from Figure 1, and it shows that there are weak correlations between the variables or there is no strong relationship between them, and none seem to directly impact water potability, which might suggest that the parameters are independent.

III. Methodology

A. Data Preprocessing

1) Red Wine Quality Dataset: Wine quality ranges from 3 to 8. Since the objective is binary classification, quality values from 3 to 5 were converted to 0 (representing ”bad wine”), and values from 6 to 8 were converted to 1 (representing ”good wine”). After this conversion, the dataset includes 855 entries labeled as ”good wine” and 744 as ”bad wine,” resulting in a relatively balanced distribution.

In Figure 1, it can be observed that the two attributes with the lowest correlation to quality are ”free sulfur dioxide” and ”residual sugar”. Consequently, these two columns were removed from the dataset before training the model, as they likely have minimal impact on wine quality. This allows the model to focus on identifying patterns in the variables that are relevant. Additionally, by reducing the number of variables, model training time should decrease due to the reduced dimensionality of the dataset.

To train the models effectively, the Stratified K-Fold technique was used. This is a cross-validation method that splits the dataset into K folds, ensuring that the class proportions remain consistent across each fold. In this case, the dataset was divided into 5 folds, each representing 20% of the data. During the training process, each fold is used once as a test set, while the other four are used as the training set. This process is repeated 5 times, ensuring that each data point is used for both training and testing. Finally, the metrics from all runs are averaged.

2) Water Quality Dataset: This specific dataset is already prepared for binary classification, where a water potability value of 0 represents ”not potable” and a value of 1 represents ”potable”, so there were no necessary changes or adaptations made in this part.

Figure 2 shows that variables in the dataset are likely independent, so no variables or columns were removed from the dataset, because water quality might be determined by a combination of parameters or non-linear relationships between them.

The dataset does contain null or missing values, specifically on the ph, sulfate and trihalomethanes columns, where null values represent roughly 15%, 24% and 5% of the total values in these columns respectively. To deal with this, the null values were replaced with the median value of the whole corresponding column, so the result is that there are no null values in the dataset.

The dataset includes 1998 entries labeled as ”not potable”

and 1278 as "potable", so there is a slightly imbalanced distribution in favor of the "not potable" class. Because this imbalance could affect the results of the model, the **SMOTE** library was used from **imblearn.over_sampling**. The results of converting the dataset using the **SMOTE** library are a perfectly balanced distribution, with 1998 entries labeled as "not potable" and 1998 labeled as "potable".

The same cross-validation Stratified K-Fold technique from the previous dataset was used to train the models, with the same parameters, folds, iterations, and divisions between training and testing sets.

The dataset contains a wide range between variable values, where some variables have values close to 22,000 and others have values close to 0. Because these wide ranges can significantly affect model performance and cause instability, data normalization was applied on the sets. For this, the **StandardScaler** library from **sklearn.preprocessing** was used on the training and testing sets. The result is that the data is now standardized and scaled correctly without it affecting model performance. Data normalization can also reduce model training time, improve stability and reduce complexity of the dataset, resulting in better training for the models.

B. Algorithms

First and foremost, it is important to note that **Grid Search** is used for each model. This technique explores all possible parameter combinations, selecting the combination that yields the best results as the final choice. The following parameters were used for both datasets:

1) Logistic Regression: The Logistic Regression model used was from the **LogisticRegression** library of **sklearn.linear_model**. The hyperparameters and their values tested were:

- **C:** 0.01, 0.1, 1, 10, 100
- **solver:** newton-cg, lbfgs, liblinear, saga
- **max_iter:** 200, 500, 1000

Here, **C** is the regularization parameter that controls the degree of regularization applied, helping to manage over-fitting. The **solver** parameter specifies the optimization algorithm, each suited to different types of data. Finally, **max_iter** sets the maximum number of iterations for the algorithm to converge on an optimal solution.

2) Decision Tree: The Decision Tree model used was from the library **DecisionTreeClassifier** of **sklearn.tree**. The hyperparameters and their values tested were:

- **max_depth:** None, 3, 5, 7, 10
- **min_samples_split:** 2, 5, 10, 15, 20
- **min_samples_leaf:** 1, 5, 10, 15, 20

Here, **max_depth** is the maximum depth the tree can reach, **min_samples_split** is the minimum number of samples required to split a node, and **min_samples_leaf** is the minimum number of samples required in a leaf node.

3) kNN: For kNN, the model selected was **KNieghborsClassifier** from **sklearn.neighbors**. The hyperparameters and their values values tested for the kNN model were:

- **n_neighbors:** 3, 5, 7, 9, 11
- **weights:** uniform, distance
- **metric:** euclidian, manhattan, minkowski

Here, **n_neighbors** represents the nearest number of neighbors considered to classify each new element or point. **Weights** determine how each neighbor influences the prediction for a new element or point. Finally, **metric** is the distance metric used to measure how close neighbors are to a given point.

4) Neural Network: For the neural network, the model selected was **MPLClassifier** from **sklearn.neural_network**. The hyperparameters and their values values tested for the neural network model were:

- **hidden_layer_size:** (50,), (100,), (50, 50)
- **activation:** logistic, relu
- **solver:** adam, sgd

Here, **hidden_layer_size** represents the number of neurons in each hidden layer of the network. **Activation** determines the activation function for the neurons in the hidden layers. Finally, **solver** is the algorithm used to optimize the weights in the network.

IV. Results and Analysis

A. Results on Red Wine Dataset

1) Logistic Regression: Training this model took **98.41 seconds** and achieved an accuracy of **0.75** on the training set. On the test set, it obtained the following metrics:

- **Accuracy:** 0.741 - This score indicates reliable but not perfect performance.
- **Precision:** 0.760 - This shows effective positive prediction reliability.
- **Recall:** 0.740 - The recall obtained demonstrates decent sensitivity.
- **AUC:** 0.816 - The model shows strong overall discriminatory power, effectively distinguishing between positive and negative classes.

Overall, the model exhibits balanced performance across accuracy, precision, and recall, with each metric around 0.74–0.76, indicating consistent but improvable results. The AUC score further reflects the model's effective classification ability.

2) Decision Tree: Training this model took **29.96 seconds** and achieved an accuracy of **0.77** on the training set. On the test set, it achieved the following metrics:

- **Accuracy:** 0.721 - This accuracy indicates a reasonable performance overall.
- **Precision:** 0.751 - This result shows a strong ability to avoid false positives.
- **Recall:** 0.720 - This demonstrates the model's good performance in identifying actual positive cases.
- **AUC:** 0.787 - The AUC obtained suggests that the model has a good capacity to differentiate between positive and negative classes.

The model achieves a balance between accuracy, precision, and recall, with an AUC score indicating strong overall discriminatory power. These metrics collectively suggest that

the model is effective in identifying true positives while maintaining a low rate of false positives. However, this model did not perform as well as the last one.

3) kNN: Training process took **12.24 seconds** for kNN and achieved an accuracy of **0.95** on the training set. On the test set, the model obtained the following metrics:

- **Accuracy:** 0.753 - This accuracy indicates a reasonable performance overall, but it could be better. It is important to check the other parameters for more information.
- **Precision:** 0.767 - This shows the model is relatively accurate when it predicts the positive class. In this case, it suggests a balanced and good handling of false positives and false negatives.
- **Recall:** 0.773 - With this parameter, it is evident that the model successfully identifies most positive cases.
- **AUC:** 0.844 - 84% shows the model is quite good at distinguishing between classes of data. This can suggest that the model is effective at ranking positive samples higher than negative ones.

In general, it can be said this model has a balanced performance, considering the precision and recall are similar. In spite of the strong AUC, there is still room to improve the accuracy of the model. Also, all the metrics for this model show an improvement to the previous two models. Furthermore, the significantly higher accuracy on the training set compared to the testing set suggests that the model may be overfitting.

4) Neural Network: Training process took **294.91 seconds** for the neural network and achieved an accuracy of **0.74** on the training set. On the test set, the model obtained the following metrics:

- **Accuracy:** 0.734 - This accuracy indicates a reasonable score overall, but it could be better. It is important to check the other parameters for more information.
- **Precision:** 0.758 - This shows the model is relatively reliable when it predicts the positive class. In this case, it suggests the model is not over-predicting the positive class and is handling false positives well.
- **Recall:** 0.739 - With this parameter, it is evident that the model identifies most positive cases successfully. However, it may still be missing some positive instances.
- **AUC:** 0.817 - 81% shows the model is good at distinguishing between classes of data. This can suggest that the model is making accurate distinctions between classes.

In general, the neural network shows an acceptable performance with room to improve. A negative aspect is this model has shown a worse performance if it is compared to the kNN model, which had better results. Furthermore, the similarity between training and testing set accuracies indicates that the model does not suffer from overfitting. Finally, it is evident that this model requires more time to train and does not deliver the best results.

It is now important to compare the algorithms and conduct a deeper analysis to determine which one performs best.

Using Figure 3, some aspects can be noticed:

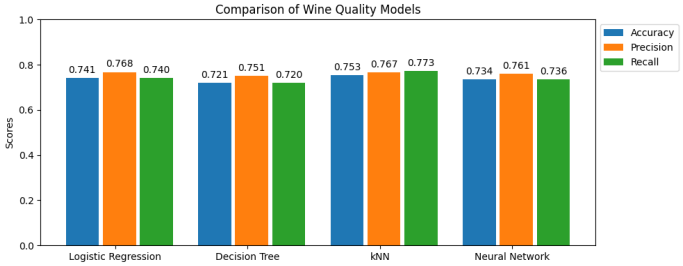

Fig. 3. Metric comparison for wine dataset

First, kNN is the model with the highest accuracy, demonstrating its overall superior performance in making predictions. Neural networks and logistic regression exhibit similar accuracies, though slightly lower. In this case, decision trees have the weakest performance, about 3% worse than the best model.

Then, Logistic Regression and kNN have similar precisions, indicating they are the most reliable models at identifying the positive class and minimizing false positives. The other two models also perform well in terms of precision, though slightly lower.

Considering recall, kNN again outperforms the others, meaning it is the best at capturing true positive instances. The remaining three models show lower recall, with results being 3-5% worse.

Finally, kNN achieves the highest AUC, highlighting its superior ability to distinguish between positive and negative classes. Neural networks and logistic regression also have a good AUC, over 80% as well. On the other hand, decision trees present a worse AUC, below 80%.

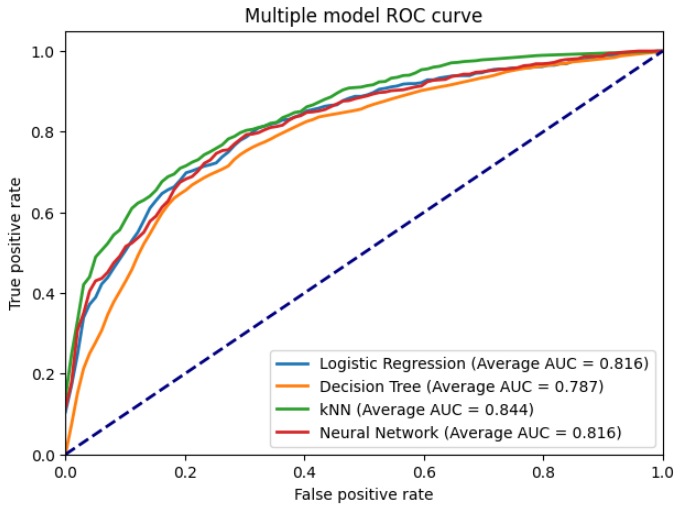

Fig. 4. ROC for wine dataset

Considering Figure 4, kNN achieves the highest performance with an average AUC of 0.844, demonstrating its superior ability to distinguish between positive and negative classes. Logistic Regression and Neural Network follow with similar AUCs of 0.816, performing well but slightly below

kNN. Decision Tree, with the lowest AUC of 0.787, is the least effective model in this evaluation.

Based on this analysis, it can be said that kNN delivers the best results. It has the highest parameters, showing it has a strong overall performance. Nevertheless, it is important to note that it may be computationally expensive for large datasets, as it stores all training instances. Therefore, logistic regression may be a good alternative, as it is the second-best model in terms of parameters and offers lower computational cost.

About the other models, Neural Networks also has a good performance in general but is lower than the previous two. But, it is important to mention that these kinds of models are highly effective to capture complex patterns. Lastly, decision trees have the lowest performance among the models analyzed. These can be useful when interpretability and understanding of results are important.

Another aspect to consider is overfitting; only kNN exhibits signs of it. Despite this, kNN remains the model with the best performance, suggesting its strong suitability for this dataset. This may also imply a moderate fit of the other three models to this specific dataset.

Finally, training time is another aspect to consider. Decision trees and kNN have the shortest training times, with decision trees producing suboptimal metrics and kNN delivering the best performance. Logistic regression also demonstrates a reasonable training time, though slightly longer than the first two. Neural networks, on the other hand, require the longest training time and do not achieve the best results, which is a disadvantage in this context.

In conclusion, it can be said that all models offer acceptable performance, with kNN being the best and decision trees the least effective. Nonetheless, there is still room for improvement in all models. Although kNN shows signs of overfitting, it still yields the best results, which may suggest a strong compatibility with this dataset. Considering training times, kNN stands out as the strongest since takes the least time to train, while decision trees and logistic regression also offer reasonable training times. In contrast, neural networks require significantly more time to train, making them less favorable in this aspect.

B. Results on Water Quality Dataset

1) Logistic Regression: Training this model took **21.49 seconds** and achieved an accuracy of **0.52** on the training set. On the test set, it obtained the following metrics:

- **Accuracy:** 0.508 - This score indicates mild and mediocre performance.
- **Precision:** 0.508 - This shows mild positive prediction reliability.
- **Recall:** 0.506 - The recall obtained demonstrates mild sensitivity.
- **AUC:** 0.519 - The model shows mediocre overall discriminatory power when distinguishing between positive and negative classes.

The model shows mediocre performance across accuracy, precision, and recall, with each metric around 0.51, which reflects balanced results but with a great room for improvement. The AUC score further confirms the model's mild classification capabilities. The similar results on the training set suggest that the model may be underfitting. It is clear that this model requires more time to train and does not deliver the best results.

2) Decision Tree: Training this model took **84.38 seconds** and achieved an accuracy of **0.94** on the training set. On the test set, it achieved the following metrics:

- **Accuracy:** 0.613 - This accuracy indicates a moderate performance overall.
- **Precision:** 0.612 - This result shows a moderate ability to avoid false positives.
- **Recall:** 0.618 - This demonstrates the model's moderate performance in identifying actual positive cases.
- **AUC:** 0.623 - The AUC obtained suggests that the model has a moderate capacity to differentiate between positive and negative classes.

The model achieves a balance between accuracy, precision, and recall, with an AUC score indicating moderate overall discriminatory power. The metrics suggest that the model is moderately effective at identifying true positives, but also maintains a moderate rate of false positives. The model performs better than the last one, but the high metric results on the training data and the moderate results on the testing data suggest the model may be overfitting.

3) kNN: Training process took **47.52 seconds** for kNN and achieved an accuracy of **1.00** on the training set. On the test set, the model obtained the following metrics:

- **Accuracy:** 0.666 - This accuracy indicates a moderate performance overall.
- **Precision:** 0.648 - This shows the model is moderately accurate when it predicts the positive class.
- **Recall:** 0.727 - This shows that the model decently identifies positive cases.
- **AUC:** 0.731 - This indicates the model is decently good at distinguishing between classes of data. This can suggest that the model is effective at ranking positive samples higher than negative ones.

This model has a slightly unbalanced performance between metrics, although this imbalance is relatively small, mostly favoring recall in return for precision. It contains a strong AUC score, although there is still room for improvement. This model shows the best performance compared to the previous models. On the other hand, the high accuracy on the training set compared to the testing set suggests model overfitting once again.

4) Neural Network: Training process took **523.34 seconds** for the neural network and achieved an accuracy of **0.86** on the training set. On the test set, the model obtained the following metrics:

- **Accuracy:** 0.660 - This accuracy indicates a fair score overall.

- **Precision:** 0.661 - This shows that the model is fairly reliable when it predicts the positive class.
- **Recall:** 0.658 - This shows that the model identifies positive cases modestly, because it may still be missing positive instances.
- **AUC:** 0.725 - This shows the model is fairly good at distinguishing between classes of data. This can suggest that the model is making moderately accurate distinctions between classes.

The model shows a mild performance with some room to improve. This model shows a similar performance to the kNN model, even though kNN had slightly better results. Once again, the higher training set accuracy compared to the testing set accuracy suggests that the model is overfitting slightly, although less than previous models.

Now, a comparison and deep analysis of the algorithms will be conducted to determine which performs best. Based on Figure 5, the following aspects can be discussed:

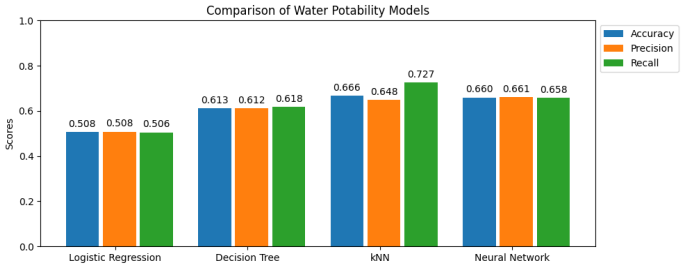

Fig. 5. Metric comparison for water dataset

In terms of accuracy, kNN is the model with the highest metric, meaning its has the best performance for making predictions. Neural Networks come in a close second, with an almost negligible difference in performance. Decision Trees come up next, with a slight drop in accuracy, though still present, and Logistic Regression comes in last with a significant drop in accuracy, making it the model with the weakest performance.

When referring to precision, Neural Networks come in first, making it the most reliable model at identifying the positive class and minimizing false positives, with kNN coming in slightly behind. Decision Trees see another small drop in performance, and Logistic Regression seeing the biggest drop again regarding this metric.

Regarding recall, kNN does marginally better than the other models, showing it identifies true positive instances better than the other models. Next, Neural Networks come in, with a meaningful drop in performance in comparison to the previous model. Then, Decision Trees are next with a slight drop in score for the corresponding metric compared to Neural Networks, and finally Logistic Regression comes in with a significant drop in performance compared to the other models once again.

As for AUC, its results further confirm the trend that has been followed by previous metrics. kNN achieves the highest score, demonstrating its dominance for distinguishing between positive and negative classes. Neural Networks come in a close second, with a negligible difference, then Decision Trees see a significant drop in performance, and Logistic Regression sees the biggest drop, making it the worst performing model.

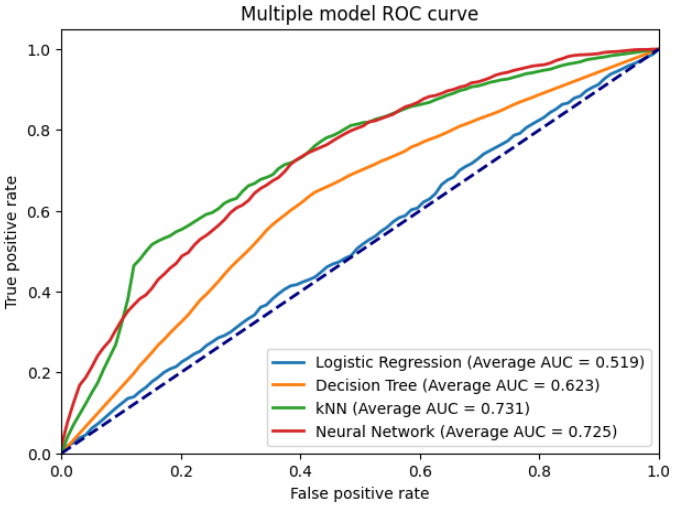

Fig. 6. ROC for water dataset

Regarding Figure 6, kNN performs the best with an average AUC of 0.731, followed closely by Neural Networks, with an AUC of 0.725. Decision Trees come next with slightly worse results, with an average AUC of 0.623 and Logistic Regression performs the worst out of all models with a low average AUC score of 0.519

Based on the previous analysis, it can be abstracted that kNN is the best performing model for this scenario, based on that it produces the highest scores in several important metrics of comparison and has an overall strong performance across parameters. Neural Networks offers a good alternative, as it comes in closely in performance to kNN across metrics, with a negligible difference in some instances. These models are the most complex and computationally expensive, but they perform the best, suggesting that for this particular dataset, more complex and bigger architectures are needed to achieve decent performance, because it might contain complex and difficult patterns and associations, that even with the models chosen still produce moderate results.

About the other models, Decision Trees have a drop in performance, but still produce a relatively decent result, and Logistic Regression sees the lowest performance across all metrics from all models, with a significant drop in performance.

Regarding overfitting and how it might affect the results of the models, kNN and Decision Trees show prominent signs of having it, and despite of this they still produce decent results, with kNN producing the best results. Neural Networks also has signs of slight overfitting, and its the second best option. This suggests that even though overfitting is present, the complexity of the models fits good with the dataset, and is able to produce moderately decent results with the data given, which might

contain complex relations and associations, outweighing the effects of overfitting. It also implies that special techniques and more complex data preparation can produce better results using the same models. It is important noting that overfitting is a common issue in machine learning models, and sometimes it can only be reduced, but not eliminated.

Concerning underfitting and how it influences the results of the models affected, only Logistic Regression shows signs of having it, and also its the model which produces the worst results in comparison to the other models. This might suggest several things, like the model did not have enough training time, as it was the fastest training model of them all by quite a margin, or that the model is too simple or not fit for such a complex and non-linear dataset, as suggested by the heatmap of Figure 2, the model does not appear to have strong linear correlations, which would cause a linear model to produce worse results.

Finally, training time is the last aspect to be considered when comparing these models. Logistic Regression has the shortest training time, but also produces the worst results, showing its not a favorable trade-off. kNN is next in training times, and it delivers the best performance. Next, Decision Trees have a tolerable training time, slightly longer than the previous model, while also producing moderately decent results. Finally, Neural Networks, although they produce the second best results, have the longest training times by quite a big margin, marking a big disadvantage for the model. This denotes that training time does not correlate directly to better or worse results, and it depends on the model's architecture and characteristics.

In conclusion, it can be abstracted that kNN is the best model from the ones chosen for the given dataset, given that it produces the best results with a low training time, with Neural Networks coming in a close second, although this model has the disadvantage of having the highest training times. These models are the most complex and are useful for finding difficult or non-linear relationships and associations, explaining why they produce the best results on the chosen dataset. Decision Trees have a slightly worse training times and results compared to kNN, but still is a viable option with similar results. All these models show signs of overfitting in various degrees, but still produce moderately decent results, suggesting they fit well to the dataset, although there is still room for improvement, that can be achieved by utilizing different or more complex techniques and procedures. The only model that shows signs of underfitting is Linear Regression, which has the lowest training times and the worst results, implying that the model does not work well with the dataset and does not have enough training time. Training times do not seem to correlate directly with the model's results, but they work as an additional metric for model comparison.

C. Cross-Dataset Analysis

When comparing the performance of the machine learning models across the two datasets, different patterns highlight the advantages and difficulties of each one.

In the case of the Red Wine Quality dataset, kNN stood out as the best-performing model with the highest score across the metrics taken for the different models. This demonstrates a balanced dataset regarding class distribution and well-defined correlations between the features and the target variables since KNN is best suited to predict on this type of dataset given the unbiased neighborhood sampling having a balanced dataset provides.

In contrast, the Water Quality dataset became a greater challenge due to its imbalance regarding classes and weak feature connections. KNN also seemed to perform best but its performance was significantly lower than in the Red Wine dataset. The difference in performance suggests that even though KNN is capable of adapting to complex data, as we can see in the case of the Red Wine quality dataset, it performs badly when a combination of features is needed for prediction or in general when nonlinear relationships are present, for example, the dependencies on specific ranges of pH in relation to sulfate levels in the decision on water potability. Neural networks, which are designed to work with these complex interactions, performed closely to KNN but required a significantly longer training time suggesting low efficiency on datasets of this scale.

Decision trees showed a consistent lower performance across both datasets indicating it may not be the most suitable model for the characteristics of the dataset. While they demonstrated average accuracy and AUC scores, a notable difference can be seen in the comparison of the datasets concerning this model since in this case there is overfitting present in the Water Quality dataset which might correlate to the complexity of this dataset in contrast to the Red Wine quality one.

Logistic Regression on the other hand, demonstrated two different performances in the datasets, in the case of the Red Wine Quality it ranked second in performance while on the Water Quality one it was the worst. This could be due to the Red Wine dataset being more linear in its relationships between features, making it easier for the logistic regression model to make assumptions which becomes harder in more complex nonlinear patterns like those found on the Water Quality dataset.

Finally, an important observation is the difference in data preprocessing techniques for the two sets. The normalization and SMOTE balancing applied on the Water Quality dataset were heavily required to achieve a performance close to that of the Red Wine quality set, this made the two preprocessing sections of the study very different in complexity giving the Red Wine dataset an advantage in that area.

V. Conclusion

This comparative study shows the importance of accurate and meditated model selection based on the dataset properties. KNN showed its adaptability to varying levels of complexity and linearity as it worked well on both datasets. However, its tendency to overfitting, which appeared in both cases, suggests that some hyperparameter tuning and normalization techniques could enhance its performance even more.

Neural networks, while performing well, their resource-heavy nature may not be the most practical choice for small datasets such as the ones used in this study. Decision trees on the other hand, seem limited on scenarios where insufficient complexity is present. Lastly logistic regression, as the simplest of the models, proves insufficient for datasets requiring more intricate decisions and relationships.

In terms of the datasets themselves, the Red Wine dataset proved to be more complete, showing balance and more direct linear connections between the features allowing for minimum preprocessing needed for an efficient training of the models. On the other hand, the Water Quality dataset showed a need for a more intricate preprocessing and hyperparameter tuning to even come close to achieving similar results to the Red Wine dataset.

Future investigations could explore hybrid approaches on datasets like this that show smaller sizes and complex relations. Additionally, more intricate data preprocessing and feature engineering might benefit the performance gaps shown on the Water Quality dataset and similar unbalanced sets. In summary, while no model is completely optimal in any case, kNN and Neural Networks show high potential on datasets presenting complex nonlinear patterns.

REFERENCES

[1] "Red Wine quality," Kaggle, Nov. 27, 2017. https://www.kaggle.com/datasets/uciml/red-wine-quality-cortez-et-al-2009

[2] "Water quality," Kaggle, Apr. 25, 2021. https://www.kaggle.com/datasets/adityakadiwal/water-potability