Project 8 Report: Water Potability

Laurel Bingham | Preethi Tera

Introduction

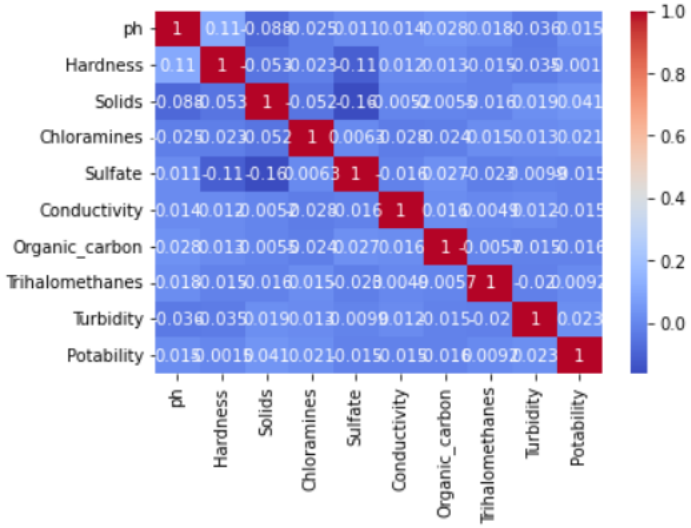
Water quality is essential for human life and environment protection. In recent years the quality of water present in rivers, lakes or any water bodies are decreasing due to pollution, industrialization etc.Our main aim of the project is to predict whether the water was safe for human consumption or not using Decision tree classifiers and Neural Networks. For this we have obtained the "water_potability" dataset obtained from Kaggle. The models' performance was assessed using various metrics, such as precision, recall, f1-score, support values, and accuracy. Additionally, we generated a decision tree to identify important features and a learning curve to evaluate model performance. Presentation | GitHub

Dataset

The dataset used in this project is "water_potability.csv" which was obtained from kaggle. It contains information on ten water quality parameters, namely pH value, hardness, total dissolved solids (TDS), chloramines, sulfate, conductivity, total organic carbon (TOC), trihalomethanes (THMs), turbidity, and potability. The potability column is the target variable, with a value of 1 indicating safe water for human consumption and 0 indicating otherwise. However, there were some missing values in the pH, sulfate, and Trihalomethanes variables, which were dropped using the dropna() function.

Analysis technique

Our goal is to predict the quality of water whether it is safe for human consumption or not by analyzing the water quality parameters. We have tried to find the correlation between variables in the data using to do feature selection.First, we divided the data into training and testing sets



and applied the Decision Tree Classifier. We attempted to determine the highest accuracy for various maximum depths and visualized the decision tree using graphviz. To evaluate the performance of our model, we created a learning curve. In the learning curve we observe how training model and validation accuracy changes as the amount of training data increases. For the Neural Networks, we

decided to test using between one and three hidden layers, with varying numbers of nodes. The smallest net tested was one with a single hidden layer of size 10, while the largest had three hidden layers, with node sizes of 100-500-100. To determine the best of the many proposed architectures, a grid search was used to find the architecture with the highest performance. From there, the learning curves of the best model, as well as a few others within the set were examined, to check for variations in accuracies among the nets, as well as to check for overfitting.

Results

Section 1:

Once we analyzed the data with the decision tree classifier, we achieved a maximum accuracy of 66.5% at the maximum depth of 5. A visual representation of our decision tree is shown in Fig. 1 .As we increased the maximum depths, the accuracy also increased, but we noticed that the model started to overfit at maximum depths of 10 and 20. Therefore, we concluded that the ideal maximum depth for our model is 5. We obtained two learning curves for our model: Fig 2 depicts the learning curve for maximum depth 5, while Fig 3 shows the learning curve for maximum depth 10.

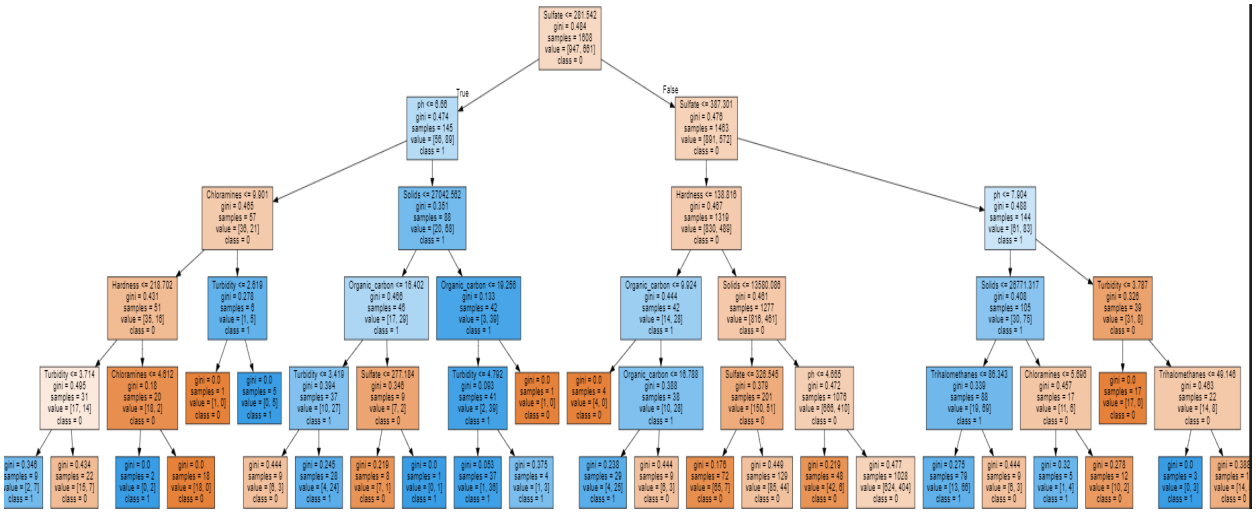


Fig. 1

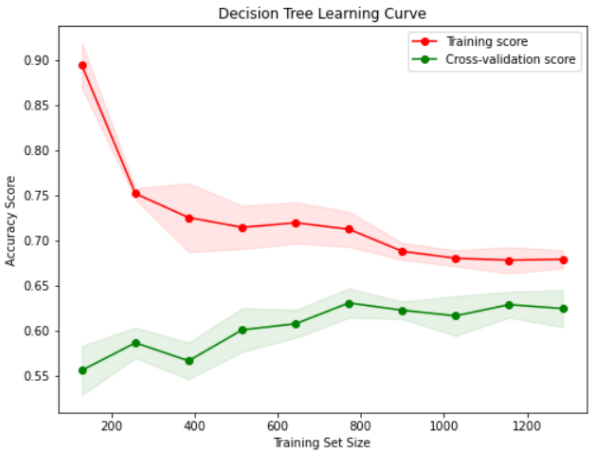


Fig. 2

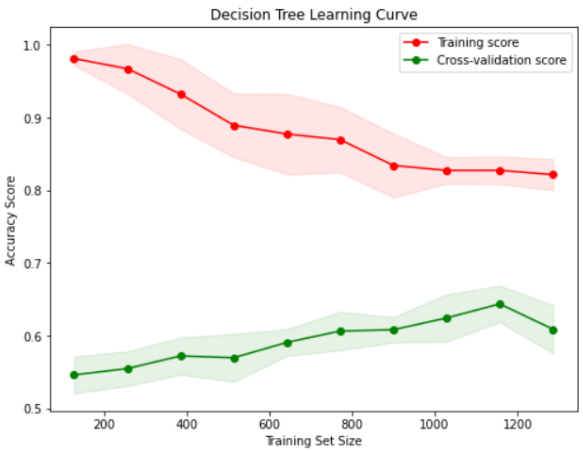


Fig. 3

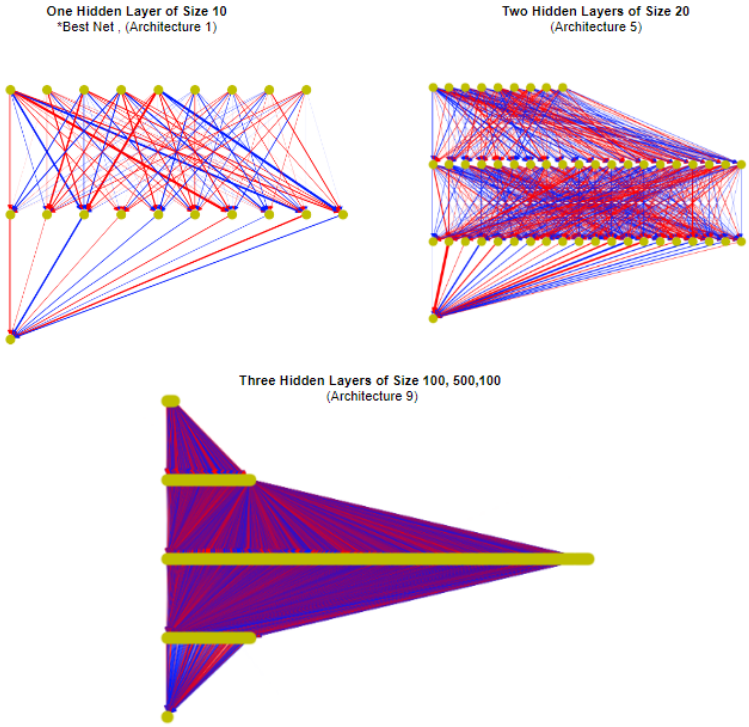
Section 2:

After attempting a variety of decision trees, we turned our attention to building a Neural Network to solve the problem of water potability. The goal, as in the previous section, was to attain the highest possible accuracy without overfitting the data. A grid search was performed across 9 different architectures. Each architecture had a varied number of hidden layers and number of nodes. Naturally, each net had an input layer of size nine, for the nine attributes analyzed, and an output layer of size one, which predicts if the sample of water is potable or not. Shown below are the architectures tested in this project.

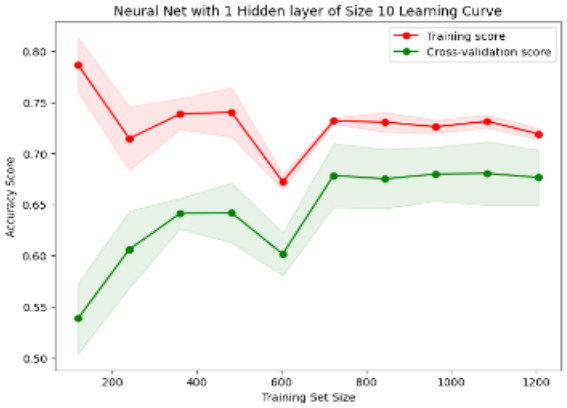
	Hidden Layer 1	**Hidden Layer 2**	**Hidden Layer 3**
Architecture 1	10	-	-
Architecture 2	10	10	-
Architecture 3	10	20	10
Architecture 4	20	-	-
Architecture 5	20	20	-
Architecture 6	20	50	20
Architecture 7	100	-	-
Architecture 8	100	100	-
Architecture 9	100	500	100

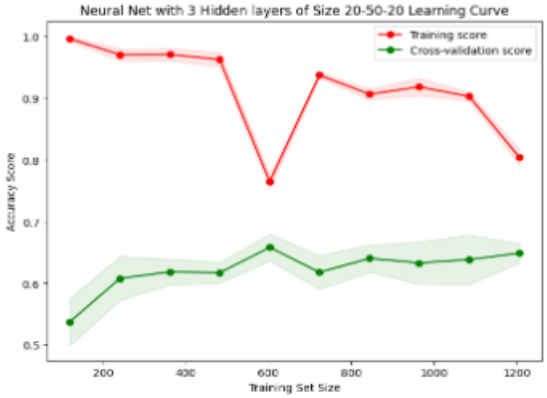
These architectures test a wide range of node sizes, as well as three different amounts of hidden layers. The learning rate was left as the sk learn default. As were any other parameters of this

net. The only variable attribute was the size and structure of the hidden layers. Below, three different net architectures are shown.



Even before observing the results, it is evident by the diagrams that Architecture 9 is absurdly sized for this problem. Below are the learning curves for these same architectures.





Quite clearly, both Architectures 5 and 9 are overfit, with accuracies close to 100 percent on the training data, but much lower accuracy (Though still close to the accuracy best performing model) on the 5 fold cross validated tests. Meanwhile, the simplest architecture, Architecture 1, has found the most generalized, and highest accuracy solution of the models. This came initially as a surprise to the project. Many times, deep learning, with millions of parameter nets are touted as the best, most state of the art models. However, upon reflection, this problem is not so high dimensional that many layers of deep nodes are required to solve the problem. Going deeper does nothing more than overfit the model on each individual data point. Compared to the decision trees, the best performing neural net only just outperformed the best decision tree. Because the best neural net had a simple architecture, the run time of building a tree and training the neural net were relatively similar. However, the grid search to find the most optimal net had a much larger run time than the decision tree construction.

Technical

Data Preparation - On the whole, this dataset was relatively clean. The two major cleaning steps taken were first to remove all rows with a Nan value in them. Even though only two columns had Nan values in them, because all columns appeared to have relatively similar importance, it

was decided that losing a few data points was preferable to losing two features of the data. Next, when the data was prepared for training on the neural networks, all training values were standardized.

Analysis - For this project, both a multilayer perceptron neural net and a decision tree were used to attempt to classify the data. A decision tree and a neural net were both good fits for this dataset because the decision boundary between the two classes, safe and unsafe drinking water, was clearly non-linear. There were not apparent correlations between any of the dataset features. Neural nets with between one and three hidden layers were tested. Each layer had between 10 and 500 nodes. In addition to graphing the accuracy and learning curves of the best model, the middle sized model with three hidden layers of size 20-50-20 was graphed, along with the largest net, which also had three hidden layers, of size 100-500-100. It should be noted that occasionally, the 20 or 100 node single layer neural net architecture performed best. However, the 10 node single hidden layer net was the most consistent performer. In all cases, a single layer proved to be superior to many hidden layers. Both architectures ultimately performed well, achieving well over chance accuracy on a dataset that had no humanly visible patterns to it. While the Neural net ultimately performed slightly better on the decision making problem, both showed promise.

Analysis process - For the most part, the analysis was simple for this assignment. For the decision tree, after minimal dataset cleaning we iterated through different tree depth sizes, first checking for accuracy, then cross validating and checking the models for over fitting. From there, it was fairly simple to identify which tree depth was the most accurate and robust model. A similar process was undertaken for the neural nets. After standardization, a grid search with 10 fold cross validation was used to determine the best model. Afterwards, the best network, as well as the mid-sized and largest network, were examined. An image of their architecture was constructed, and their learning curves were plotted with a second, 5 fold cross validation test.