Machine Learning: Mine Versus Rock Report

Carson Wagner

Confusion Matrix:

The maximum accuracy achieved was 0.952 for 15 components. This means there is a 95.2% chance of surviving in the minefield. The Confusion Matrix shown in **Table 1** shows that the total of 63 samples in the data set. For Rock there was a total of 28 samples and there were 35 samples for Mines. Based on the Confusion Matrix, there was **1 false negative** where it was predicted to be a Rock but was actually a Mine. Meanwhile there were **2 false positives** as there were 2 samples that were predicted to be mine when it was actually a Rock. Having **false positives** is okay as it is a mistake that can be afforded. However, a **false negative** is not okay as this mistake can be deadly as the Mine was not detected correctly. Due to this issue, the machine learning model should be trained for minimizing the number of **false negatives** to reduce the risk of death. The model created failed to correctly predict **1 Mine** which means this model can be improved further.

	Rock (Predicted)	Mine (Predicted)
Rock (Actual)	26	2
Mine (Actual)	1	34

Table 1

Conclusion:

The maximum accuracy of this model was 0.952 with 10 components which means the components have high correlation to the classes and are important in improving the machine learning model. As a result of 10 out of 60 components being used, the other 50 dimensions were discarded, allowing for more efficiency. This enabled the model to become faster and less prone to fitting noise.

Parameter	**Value**
hidden_layer_sizes	(120, 110)
activation	'relu'
max_iter	2000
alpha	0.0001
solver	‘adam’
tol	0.0001
learning_rate	‘constant’
random_state	1

Table 2

The parameters chosen for the MLPClassifier can be seen in **Table 2**. I took the original parameters given and edited the values to find the best accuracy and for n-components around 15 components. I changed the value of **hidden_layer_sizes** to be **(120, 110)** to make the depth 120 and the width 110 of the neural network. This allowed for the model to be deep and wide to give high capacity to learn the relationships in the sonar data given. The **activation** parameter was set to **‘relu’** for the Rectified Linear Unit as it is the standard choice for deep networks because it computes simple and helps prevent vanishing gradient problems, which allows the network to train faster and more efficiently. **Solver** was kept at **'adam'** as it is highly efficient and automatically adapts the learning rate for each weight individually. **Learning_rate** was set to **'constant'** to set the learning rate to be the same throughout the entire process. **Max_iter** was set to **2000** to tell the model that the maximum number of passes through the algorithm would be 2000 times. **Tol** was set to **0.0001** to stop the training process if the score doesn't improve by at least this value over two consecutive iterations. **Alpha** was set to **0.0001** to penalize the model's loss function proportional to the square of the weights. This prevents the weights from becoming too large, this is the default value. **Random_state** was set to **1** to ensure the network's initial random weights are the same every time the code is ran. This is essential for reproducibility so if the code is ran multiple times, the results will be the same each time. Originally the **hidden_layer_sizes** was set **(200, 150)**, but this used 33 components, which was too many, so **(120, 110)** was used instead as it was around the 15-component mark. These parameters enabled the PCA Model to achieve an accuracy of **95.2% with 10 components. Figure 1** shows the **Accuracy vs Number of Components.**

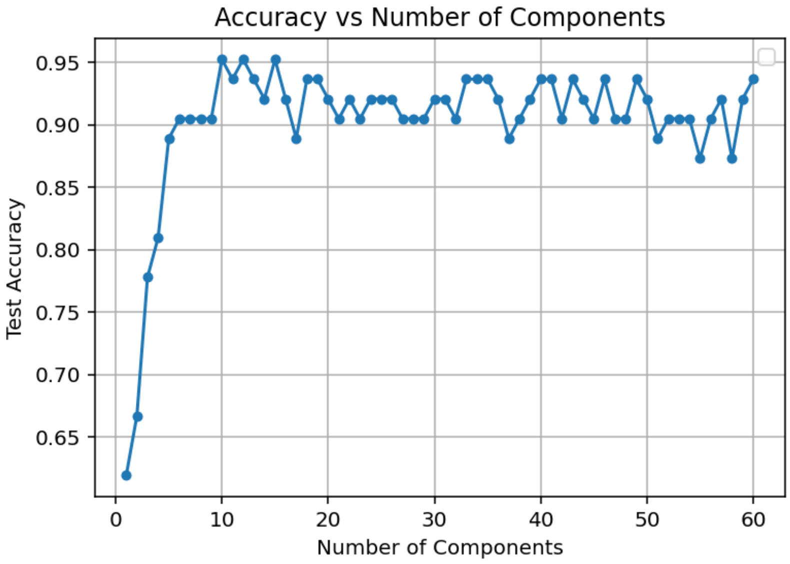

Figure 1