Project 2

Machine Learning: Mine vs Rock Report

Raakesh Sureshkumar (1218477559)

1. Accuracy Inference:

From part(3) in the deliverables as mentioned in the problem statement, we get the number of components that give the maximum test accuracy. In our case, it produces an output of 90.48%. This indicates that there is 90.48% chances of predicting if an object is a mine or a rock. So, we have a chance of 90.48% to survive a mine.

2. Confusion matrix:

The confusion matrix has four components. The (0,0) element [26 in this case] and (1,1) element [29 in this case] contain the number of samples that predicted the same output as the expected output given in the given dataset target. (0,0) represents the classification for '1' in this case which correspond to rocks and (1,1) correspond to the number of samples with '2' which correspond to mines. The other elements (0,1) [5 in this case] and (1,0) [3 in this case] represent the number of samples that were predicted false by the model. In this case (0,1) represents the wrong prediction of a rock as a mine and vice versa for (1,0). The total number of components in the confusion matrix is the total number of components used for the prediction. In our case, the total number of samples is '208'. The data has been split as 70% train data and 30% test data. So, the test sample space is '63'. So, the sum of elements in the confusion matrix must be 63.

3. Explanation for maximum number of components:

In this case, the maximum number of components for max accuracy is 53. The components that produce the maximum variance in the data determine the difference between a mine and a rock more clearly than the ones with the smaller variance. PCA performs this analysis and provides the components providing the most variance. So, these components are subsequently taken into consideration to produce the maximum accuracy.

4. Shape of the plot

The accuracy increases at the rate of 3x until there are top 10 PCA components. From this point, there is abundant variation in the accuracies in the range of 80% to 90%. However, the maximum accuracy is achieved when the top 34 components of the PCA are for the MLPClassifier. The maximum accuracy is 90.48% that remains stable from 34 through 36 components.

5. Choosing the parameters for the MLPClassifier

MLP (Multilayer Perceptron) classifier is a supervised learning algorithm that uses a neural network to assign the class of a set of samples with pre-defined targets. The weights are assigned dynamically veried for every iteration based upon the error in the classification.

a. **hidden_layer_size:**

The higher the number of hidden layers and the number of nodes in each hidden layer, the higher the accuracy. But, it also depends on the number of features that can be extracted from the sample space. The accuracy may reach a saturation level for increase in the number and the size of the layer. So, in our case, with multiple layers tested, we have fixated to 1 hidden layer with a size of 100

b. **max_iter, n_iter_no_change, tol:**

These 2 are stabilization parameters. The maximum itertions denotes the max number of iterations the program must execute before, it exits to get the maximum final accuracy. This might be disadvantageous if set a high value as the convergence would have happended sooner than the maximum iterations set. So, the time taken increases which is unnecessary. Setting a smaller value may also, have a profound effect on the results by giving a sub-optimal accuracy.

The n_iter_no_change parameter is set to exit the minimization once the accuracy is maintained within the specified tolerance for more than n_iter_no_change number of iterations.

The tolerance specifies the range within which the final accuracy can produce an error from its previous iteration.

max_iter was verified by setting a range: (200, 7000, 100) in the grid search. max_iter: 5000

n_iter_no_change was verified by setting a range: (10, 100, 10) in the grid search. n_iter_no_change: 20

tol was verified by setting a range: (0.0001, 0.01, 0.0001) in the grid search

tol: 0.0001

c. **activation, solver:**

'Logistic' functions are the most widely used non-linear activation functions for a classification learning model with less number of hidden layers. As these perform efficiently for a small hidden layered model, we can use logistic function which is sigmoid function. "relu" (Rectified Linear Unit) activation comes in good comparison with the "logistic" functions and suppresses the effect of vanishing gradient incase of increased number of hidden layers. Also, it is more efficient in training and is widely in much non-trivial classification solutions.

So, we choose 'relu' - Rectified Linear Unit

Solvers are the optimizers that perform the minimization of the errors between the calculated classification prediction and the actual target. These are based upon the activation function and assign the weights for the next itration based upon the error in the previous iteration. Adam is the most widely used with an affinity to produce better results for huge databases.

So, we choose 'adam'

d. **alpha, learning_rate, learning_rate_init:**

alpha is the learning rate or the gradient descent step which limits the gradient descent of the optimizer. The lower the value, the slower the optimizer reaches the required global minimum. This highly depends on the maximum iterations as the epochs have high chances of exceeding the maximum iteration without converging.

So. alpha is taken to be optimal of 0.0001

learning_rate_init: Specifies the learning with which the optimizing must start.

So, using grid search range of (0,0.02, 0.0001), the answer was 0.0012

learning_rate: This defines whether the learning rate needs to be a constant or adapt based on the trend in the convergence. If the gradient descent is higher, the learning rate is reduced to achieve a slower descent to get to the most accurate convergence.

So, learning_rate is taken to be 'constant'

Random state has been maintained the same as the random seed that has been used for the train_test_split and the PCA. Any constant non-zero value, set to the maximum accuracy to be the same though variations are observed

Significance of PCA in this project:

The PCA (Principal Component Analysis) concept analyses the dataset and provides the impact of each of the components in the dataset on the total varaiation in the data. So, by setting the number of PCA components to 1, we get a single component that has the maximum impact on the variance of the dataset and when number of components is 2, we get the first 2 components producing the highest percentage of variance from within the entire dataset. This involves a linear fit algorithm that maximizes the squared distances from a line fit in the sample space.

The random seed in this part has an effect on the accuracies for every execution of the program or a fit to the model. So, maintain consistency in the data output, the random seed has been set to a constant

2. From part(2), we get the accuracy of the MLPClassifier. The table denotes the number of components output from the PCA and these components are used to perform MLPClassifier on the train set. The test set is later used to get the predict accuracies using the previously fit MLPClassifier model.

The random seed in splitting the data has a noted effect on the accuracyas the dataset changes every time a model is fit or accuracies are predicted.