Q2

Dataset description

The Dataset contains data of 3 classes of 50 instances, each, where each class refers to a type of iris plant. One class is linearly separable from the other 2; the latter are not linearly separable from each other

No of features = 4
The features in order : From leftmost to rightmost column are
1.sepal length in cm
2.sepal width in cm
3.petal length in cm
4.petal width in cm

The classes are of 3 types(column 5)
-Iris Setosa
-Iris Versicolor
-Iris Virginica

Dataset link:https://archive.ics.uci.edu/ml/datasets/Iris

There are no missing values in any row of the dataset.

Dataset preprocessing

The data was normalised using standard scalar normalisation technique.The mean and standard deviation was calculated for each column except the label column and each value of a column was subtracted from mean and divided by standard deviation of the column to convert each column to zero mean and unit variance data.This helps if the datas in column follow gaussian distribution in some order.

The label column (the class of iris plant) depicted 3 types of iris plant.It was categorically encoded using label encoding where the labels ranged from 0 to 2.
Label 0:Iris Setosa
Label 1:Iris Versicolor

Label 2:Iris Virginica

The data was randomly shuffled for splitting into train and test samples(80:20 ratio according to question)

Function definition and their usages

1.standard_scalar_normaize(df):

This function performs standard scalar normalisation on the attribute columns by computing mean and standard deviation of each column and subtracting each row of a particular column and dividing it by the standard variation of that column.

2.categorical_encoding(df):

This function performs categorical encoding of the label column (the type of iris plant) using label encoding method

0 -> Iris-Setosa
1 -> Iris-Versicolor
2 -> Iris-Virginica

3.shuffle_dataframe(df):

This column shuffles the rows of the dataframe randomly so as to ensure a proper distribution of samples in training and test data

4.train_test_split(df)->tuple:

This functions splits dataset randomly with ratio of 80:20 into train and test data and returns a tuple of train and test.

5.find_linearly_separable(df):

This function finds linearly separable classes in the data and returns it.This is done by considering each class against the other two and fitting an SVM classifier on it. The C hyperparameter was set to a high value which represents the penalty on each misclassified sample.If c is

small a precision boundary with a large margin is chosen at the expense of greater number of misclassifications.

Since C is large,it leads to overfitting and if we can overfit the data using a linear model,it means data is linearly separable.

We find using this method that class Iris-Setosa is linearly separable than class Iris-Versicolor and Iris-Virginica.

6.find_accuracy_SVM(train_x,train_y,test_x, test_y, k, basis,C=1.0,degree=3,multiclass = True):

This function finds the accuracy of SVM classifier.Based on class k it separates the data into two samples of class k as 1 and non class k as -1 if flag multiclass = True else implements binary SVM.

It then creates an SVM classifier using basis,C and degree(only for polynomial basis) as given and returns accuracy computed.

SVM classifier created using SVC function of class sklearn.svm..fit() method was used for fitting the training and test data and .predict() method was used for predicting results,given a test_set.Accuracy score was computed using accuracy_score() function of sklearn.metrics class

7.print_accuracy_SVM(df, k):

This function prints the accuracy on fitting an SVM classifier with ‘linear‘,’quadratic’ and ‘radial basis function’ kernel.Accuracy for each reported under the Results of each part section below.The k represents the class linearly separable than the other two so as to implement the binary SVM classifier.

8.find_accuracy_MLP(train,test,hidden_layers,learning_rate):

This function finds the accuracy on creating an MLP classifier given the hidden_layers and learning_rate on the train and test data.It first splits the train and test into train_x,train_y and test_x and test_y and then constructs an MLP classifier and fits onto to the training data.The accuracy of predictions on test data is returned.**Number of epochs (max_iter parameter) set to 3000 for letting the classifier converge in all cases.**

9.get_accurate_MLP(df):

This function finds the accuracy on varying the hidden layers and number of nodes in each layer as given in the question and prints accuracy of each and returns the most accurate model.

It returns 1 if 1 hidden layers with 16 nodes is more accurate and 2 if 2 hidden layers with 256 and 16 nodes is greater than or equal to accuracy than first one.

10.perform_backward_elimination(df,hidden_layers,learning_rate):

This function performs backward elimination on the feature set in the following manner:
It uses the best accurate model found in part 3.

```
# performs backward elimination of features and prints the best set of features
# find the feature whose removal causes least error and remove if error less than original error
# repeat until no feature can be removed,i.e, original error is less than error after removing a feature
# error = number of samples*(1 - accuracy)
# since number of samples is constant (80:20 split),error is proportional to (1-accuracy)
# therefore we can compare accuracies instead of errors
```

The same comments is provided in the code as well.This function also prints the accuracy on removing each feature and also the feature name if it is removed from the dataset.

Finally prints the reduced set of features.

11.ensemble_max_voting_technique(df,hidden_layers):

This function performs max voting technique(ensemble learning) using the following models:

1.SVM with quadratic kernel
2.SVM with radial basis function kernel
3.most accurate model found in part 3

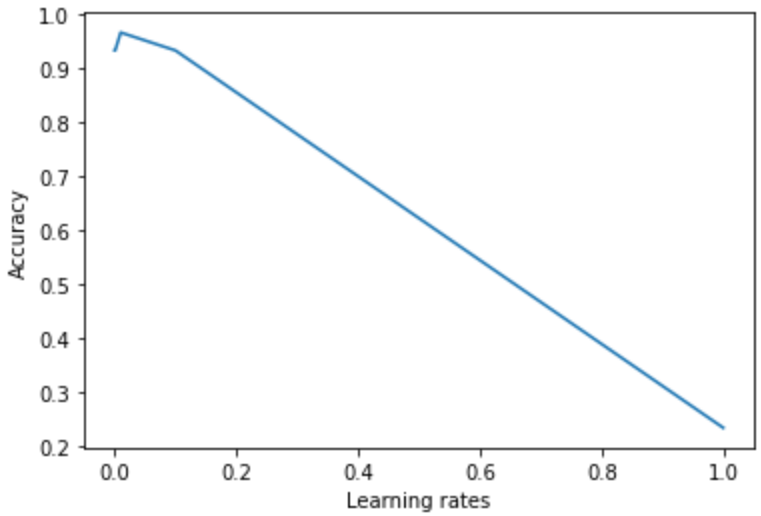
It predicts each data points,using the class prediction receiving maximum votes.In case of ties,it breaks ties in favour of the most accurate model and prints the accuracy after computing all predictions.

Results of each part

2.accuracy of SVM (kernel = linear) =0.9333333333333333
accuracy of SVM (kernel = quadratic) =0.8333333333333334
accuracy of SVM (kernel = rbf) = 0.9333333333333333

3.(a) accuracy = 0.9333333333333333
(b) accuracy = 0.9333333333333333
chosen model second one with hidden layers 2 and 256 and 16 nodes.

4. **Accuracy vs Learning rate plot**



5. features after backward elimination = sepal length, sepal width and petal length

accuracy = 0.9666666666666667

6. accuracy (ensemble model) = 0.9333333333333333

Output.txt

FINDING LINEARLY SEPARABLE DATA
Class Iris-setosa is linearly separable than other two classes, Iris-versicolor and Iris-virginica
Accuracy of SVM using linear kernel is 0.9333333333333333
Accuracy of SVM using quadratic kernel is 0.8333333333333334
Accuracy of SVM using radial basis function kernel is 0.9333333333333333
The accuracy of MLP classifier with one hidden layer of 16 nodes is 0.9333333333333333
The accuracy of MLP classifier with two hidden layers of 256 and 16 nodes is 0.9333333333333333
Second option given in question gives more accuracy,hidden layers = (256,16)
Learning rates are [1e-05, 0.0001, 0.001, 0.01, 0.1, 1]
Accuracies are [0.9333333333333333, 0.9333333333333333, 0.9333333333333333, 0.9666666666666667, 0.9333333333333333, 0.23333333333333334]

```
Maximum accuracy learning rate is 0.01
BACKWARD ELIMINATION STARTED
Current accuracy is  0.9333333333333333
Accuracy after removing feature 0 is 0.9333333333333333
Accuracy after removing feature 1 is 0.9333333333333333
Accuracy after removing feature 2 is 0.9
Accuracy after removing feature 3 is 0.9666666666666667
Removing feature petal width
Current accuracy is  0.9666666666666667
Accuracy after removing feature 0 is 0.9666666666666667
Accuracy after removing feature 1 is 0.9666666666666667
Accuracy after removing feature 2 is 0.8666666666666667
Stopped removing features
The best set of features is ['sepal length', 'sepal width',
'petal length']
BACKWARD ELIMINATION ENDED
Accuracy of ensemble model is 0.9333333333333333
```