Machine Learning (CS60050)
Assignment 3 - ANN &SVM

Submitted by: Group No: 083
Ravi Pratap Singh (20CS60R60)
Sahil Jain (20CS60R64)

Task 1:

- Converted ThoracicSurgery.arff to csv file
- Dropped the unnecessary columns
- Encoded the categorical values
- Split into 80-20 sets

	DGN	PRE4	PRE5	PRE6	PRE7	PRE8	PRE9	PRE10	PRE11	PRE14	PRE17	PRE19	PRE25	PRE30	PRE32	AGE	Risk1Yr
0	2	2.88	2.16	1	0	0	0	1	1	4	0	0	0	1	0	60	0
1	3	3.40	1.88	0	0	0	0	0	0	2	0	0	0	1	0	51	0
2	3	2.76	2.08	1	0	0	0	1	0	1	0	0	0	1	0	59	0
3	3	3.68	3.04	0	0	0	0	0	0	1	0	0	0	0	0	54	0
4	3	2.44	0.96	2	0	1	0	1	1	1	0	0	0	1	0	73	1

Part a :

Implemented SVM classifier using
-Linear kenal
- Polynomial kernel with degree 2 (gamma= 1)
-Radial Basis Function kernel with (gamma =0.0001)

Part b:

Used GridsearchCV from scikit learn package with a number of jobs as 4 , verbose as 10 and 5 fold cross validation for every kernel .

(i) Training and Test set accuracies for the Linear kernel for C varying in [0.01,0.1, 1, 10, 100, 1000,10000] . Below 0.01 for every value of C the accuracy was almost the same.

C	0.01	0.1	1	10	100	1000	10000
Train Accuracy	0.8537193	0.8537193	0.8537193	0.8537193	0.8537193	0.8377193	0.8217193
Test Accuracy	0.840425531	0.840425531	0.840425531	0.840425531	0.840425531	0.8276193	0.8116193

Best Test accuracy : 0.840425531

(ii) Training and Test set accuracies for the Polynomial(degree 2) kernel for C varying in [0.01,0.1, 1, 10, 100, 1000,10000] . Below 100 for every value of C the accuracy was almost the same.

C	0.01	0.1	1	10	100	1000	10000
Train Accuracy	0.8537193	0.8537193	0.8537193	0.8537193	0.8537193	0.8377193	0.8217193
Test Accuracy	0.8191489	0.8191489	0.8191489	0.8191489	0.8191489	0.8033319	0.7977193

Best Test accuracy : 0.8191489

(iii) Training and Test set accuracies for the RBF kernel for C varying in [0.01,0.1, 1, 10, 100, 1000,10000] and gamma in [1, 0.1, 0.01, 0.001, 0.0001]. Below 1 for every value of C the accuracy was almost the same and we got best gamma value as 1 .

C	0.01	0.1	1	10	100	1000	10000
Train Accuracy	0.8537193	0.8537193	0.8537193	0.8537193	0.8537193	0.8377193	0.8217193
Test Accuracy	0.8404255 31	0.8404255 31	0.8404255 31	0.8377894 7	0.8351228 1	0.8351228 1	0.8351228 1

Best Test accuracy : 0.840425531

Task 2:

Builded a MLP classifier for the given dataset & Use stochastic gradient descent optimiser for the models .

- Building a MLP classifier using Sklearn
- Defined function to measure accuracy

Part a: (i) 0 hidden layer : Accuracy :84.04255319148936

(ii)1 hidden layer with 2 nodes : 77.6595744680851

(iii)1 hidden layer with 6 nodes : 84.04255319148936

(iv)2 hidden layers with 2 and 3 nodes : 84.04255319148936

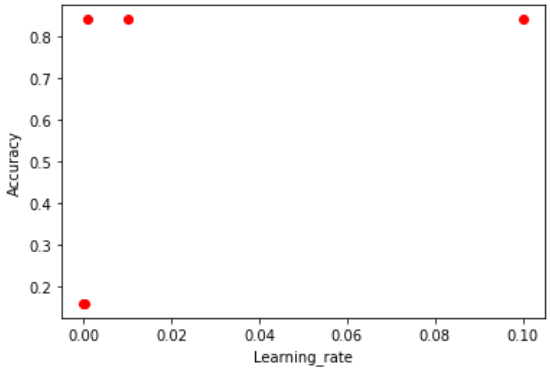
(v) 2 hidden layers with 3 and 2 nodes : 84.04255319148936

Part b:

Learning rate vs accuracy ; Varying the learning rates as 0.1, 0.01, 0.001, 0.0001, 0.00001

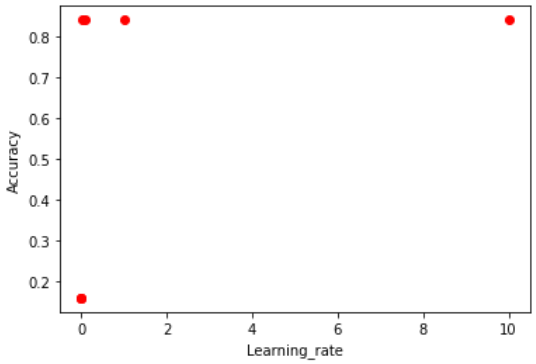
(i) 0 hidden layers :

```
Test set score: 0.840426
Test set score: 0.840426
Test set score: 0.840426
Test set score: 0.159574
Test set score: 0.159574
```



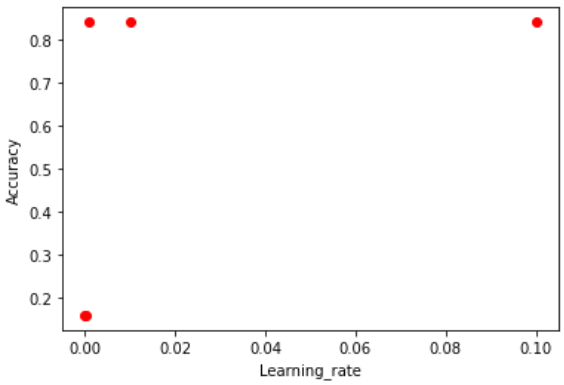
(ii)1 hidden layer with 2 nodes :

```
Training set score: 0.853723
Training set score: 0.853723
Training set score: 0.146277
Training set score: 0.146277
Training set score: 0.146277
```



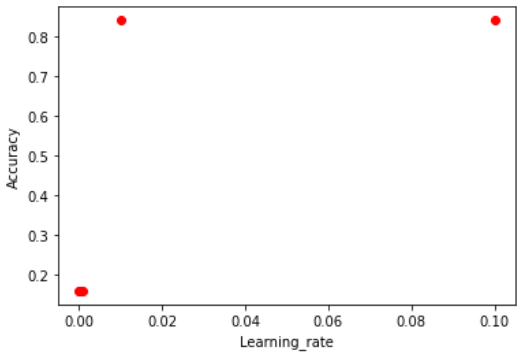
(iii)1 hidden layer with 6 nodes :

```
Test set score: 0.840426
Test set score: 0.840426
Test set score: 0.840426
Test set score: 0.159574
Test set score: 0.159574
```



(iv)2 hidden layers with 2 and 3 nodes :

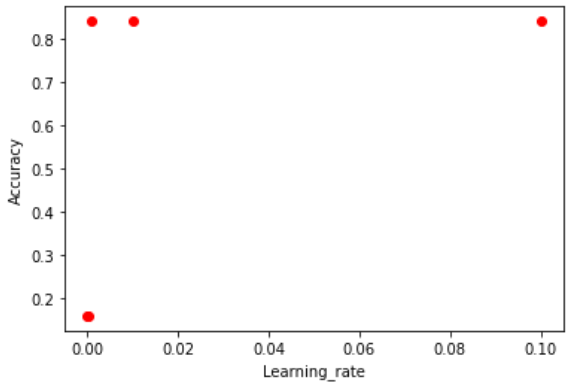
```
Test set score: 0.840426
Test set score: 0.840426
Test set score: 0.159574
Test set score: 0.159574
```



Test set score: 0.159574

(v) 2 hidden layers with 3 and 2 nodes :

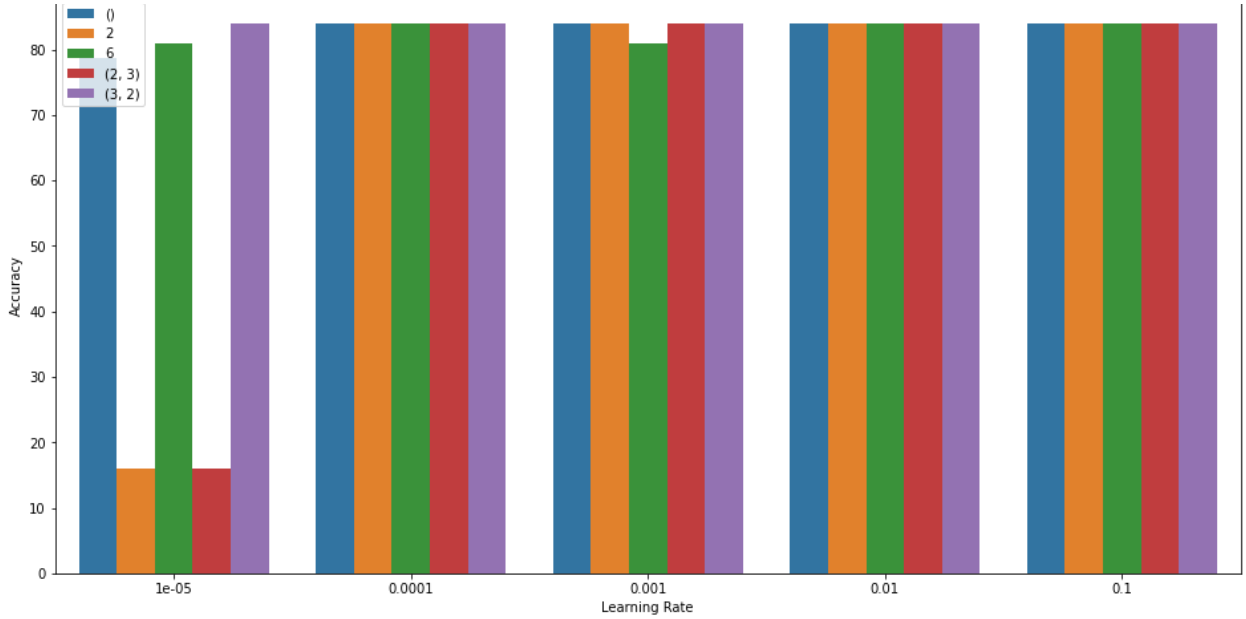
```
Test set score: 0.840426
Test set score: 0.840426
Test set score: 0.840426
Test set score: 0.159574
Test set score: 0.159574
```



Model vs accuracy for each learning rate ;

Values for plot :

```
for hidden layer size  () with learning rate  0.1 accuracy is :  84.04255319148936
for hidden layer size  2 with learning rate  0.1 accuracy is :  84.04255319148936
for hidden layer size  6 with learning rate  0.1 accuracy is :  84.04255319148936
for hidden layer size  (2, 3) with learning rate  0.1 accuracy is :  84.04255319148936
for hidden layer size  (3, 2) with learning rate  0.1 accuracy is :  84.04255319148936
for hidden layer size  () with learning rate  0.01 accuracy is :  84.04255319148936
for hidden layer size  2 with learning rate  0.01 accuracy is :  84.04255319148936
for hidden layer size  6 with learning rate  0.01 accuracy is :  84.04255319148936
for hidden layer size  (2, 3) with learning rate  0.01 accuracy is :  84.04255319148936
for hidden layer size  (3, 2) with learning rate  0.01 accuracy is :  84.04255319148936
for hidden layer size  () with learning rate  0.001 accuracy is :  84.04255319148936
for hidden layer size  2 with learning rate  0.001 accuracy is :  84.04255319148936
for hidden layer size  6 with learning rate  0.001 accuracy is :  80.85106382978722
for hidden layer size  (2, 3) with learning rate  0.001 accuracy is :  84.04255319148936
for hidden layer size  (3, 2) with learning rate  0.001 accuracy is :  84.04255319148936
for hidden layer size  () with learning rate  0.0001 accuracy is :  84.04255319148936
for hidden layer size  2 with learning rate  0.0001 accuracy is :  84.04255319148936
for hidden layer size  6 with learning rate  0.0001 accuracy is :  84.04255319148936
for hidden layer size  (2, 3) with learning rate  0.0001 accuracy is :  84.04255319148936
for hidden layer size  (3, 2) with learning rate  0.0001 accuracy is :  84.04255319148936
for hidden layer size  () with learning rate  1e-05 accuracy is :  78.72340425531915
for hidden layer size  2 with learning rate  1e-05 accuracy is :  15.957446808510639
for hidden layer size  6 with learning rate  1e-05 accuracy is :  80.85106382978722
for hidden layer size  (2, 3) with learning rate  1e-05 accuracy is :  15.957446808510639
for hidden layer size  (3, 2) with learning rate  1e-05 accuracy is :  84.04255319148936
best architecture:  () with best learning rate:  0.1 the best acuracy is :  84.04255319148936
```



Part c:

Best Architecture : 1 hidden layer with 2 nodes
Optimizer : stochastic gradient-based optimizer
Learning rate: 0.1
Accuracy : 0.853723

Usually if we have a lot of training examples, we can use multiple hidden units, but sometimes just 2 hidden units works best with little data.Here only 1 Hidden layer worked to give best accuracy .

The learning rate hyperparameter controls the rate or speed at which the model learns. Specifically, it controls the amount of apportioned error that the weights of the model are updated with each time they are updated, such as at the end of each batch of training examples.
Given a perfectly configured learning rate, the model will learn to best approximate the function given available resource

During training, the backpropagation of error estimates the amount of error for which the weights of a node in the network are responsible. Instead of updating the weight with the full amount, it is scaled by the learning rate.
Since we cannot analytically calculate the optimal learning rate for a given model on a given dataset. Instead, a good (or good enough) learning rate must be discovered via trial and error. For our dataset we got the best learning rate as 0.1 by trying in [0.1, 0.01, 0.001, 0.0001, 0.00001]. (the trial and error statistics for every learning rate and hidden layers are provided above)

Task 3:

Comparison between the two classifier :

Classifier		Accuracy	Parameters
SVM	Linear	0.8404255319148937	C=0.01
	Polynomial (degree=2)	0.8191489361702128	C= 0.01
	Radial Basis Function	0.8404255319148937	C=0.01 Gamma =1
MLP	0 hidden layer	0.840426	-
	1 hidden layer with 2 nodes	*0.853723*	-
	1 hidden layer with 6 nodes	0.840426	-
	2 hidden layers with 2 and 3 nodes	0.840426	-
	2 hidden layers with 3 and 2 nodes	0.840426	-