Lab -9
PRML
Support vector machine (SVM)

Name : Akshaykumar Kanani (B19EE008)

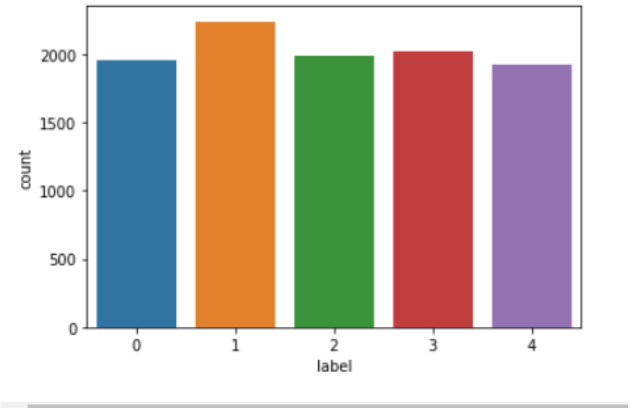
Q1:
First we did some preprocessing and got the data as:

```
mn_train_s_df.head()
```

	label	0	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	0.10	0.11	0.12	0.13	0.14	0.1
0	5	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
1	7	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
2	9	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
3	5	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
4	2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	

5 rows × 785 columns

And we remove the target 5,6,7,8,9 as we only have to work on 0,1,2,3,4 as per the instruction.
And for all remaining target we get:



And then we split the data and do model fitting and processing and got the following result after grid search:

Before normalization:
For SVM:
We got
Score = 0.961
Best grid-parameters from train data: {'C': 0.1, 'gamma': 1, 'kernel': 'linear'}

For KNeighborsClassifier
We got
score = 0.983
Best grid-parameters from train data: {'metric': 'euclidean', 'n_neighbors': 3}

For Perceptron
We got
score = 0.947
Best grid-parameters from train data: {'alpha': 0.0001, 'tol': 0.001}

After Normalization:
For SVM:
We got
score = 0.966
Best grid-parameters from train data: {'C': 0.1, 'gamma': 1, 'kernel': 'linear'}

For KNeighborsClassifier
We got
score = 0.953
Best grid-parameters from train data: {'alpha': 0.0001, 'tol': 0.001}

For Perceptron
We got
score = 0.953
Best grid-parameters from train data: {'alpha': 0.0001, 'tol': 0.001}

Q2:
First we did some preprocessing and got the data as:

	Pregnancies	Glucose	BloodPressure	SkinThickness	Insulin	BMI	DiabetesPedigreeFunction	Age	Outcome
0	6	148	72	35	0	33.6	0.627	50	1
1	1	85	66	29	0	26.6	0.351	31	0
2	8	183	64	0	0	23.3	0.672	32	1
3	1	89	66	23	94	28.1	0.167	21	0
4	0	137	40	35	168	43.1	2.288	33	1

And also from data.describe method we got:

```
<bound method NDFrame.describe of      Pregnancies  Glucose  ...  Age  Outcome
0              6      148  ...   50        1
1              1       85  ...   31        0
2              8      183  ...   32        1
3              1       89  ...   21        0
4              0      137  ...   33        1
..           ...      ...  ...  ...      ...
763           10      101  ...   63        0
764            2      122  ...   27        0
765            5      121  ...   30        0
766            1      126  ...   47        1
767            1       93  ...   23        0

[768 rows x 9 columns]>
```

And then we did data split and grid search and get the result:
For linear kernel:
score = 0.831
Best grid-parameters from train data: {'C': 1, 'kernel': 'linear'}

For rbf kernel:
score = 0.805
Best grid-parameters from train data: {'C': 10, 'kernel': 'rbf'}

And got the accuracy=0.8441558441558441

Thank you