Machine Learning and Neural Computing

Data Classification Coursework

20070280

Zohaib Rehman

Experimental Report

Task 1

1a) Before loading the dataset into the notebook, Pandas, NumPy, MatplotLib.pyplot and Sk-learn were imported into the file as pd, np, plt and sk respectively.

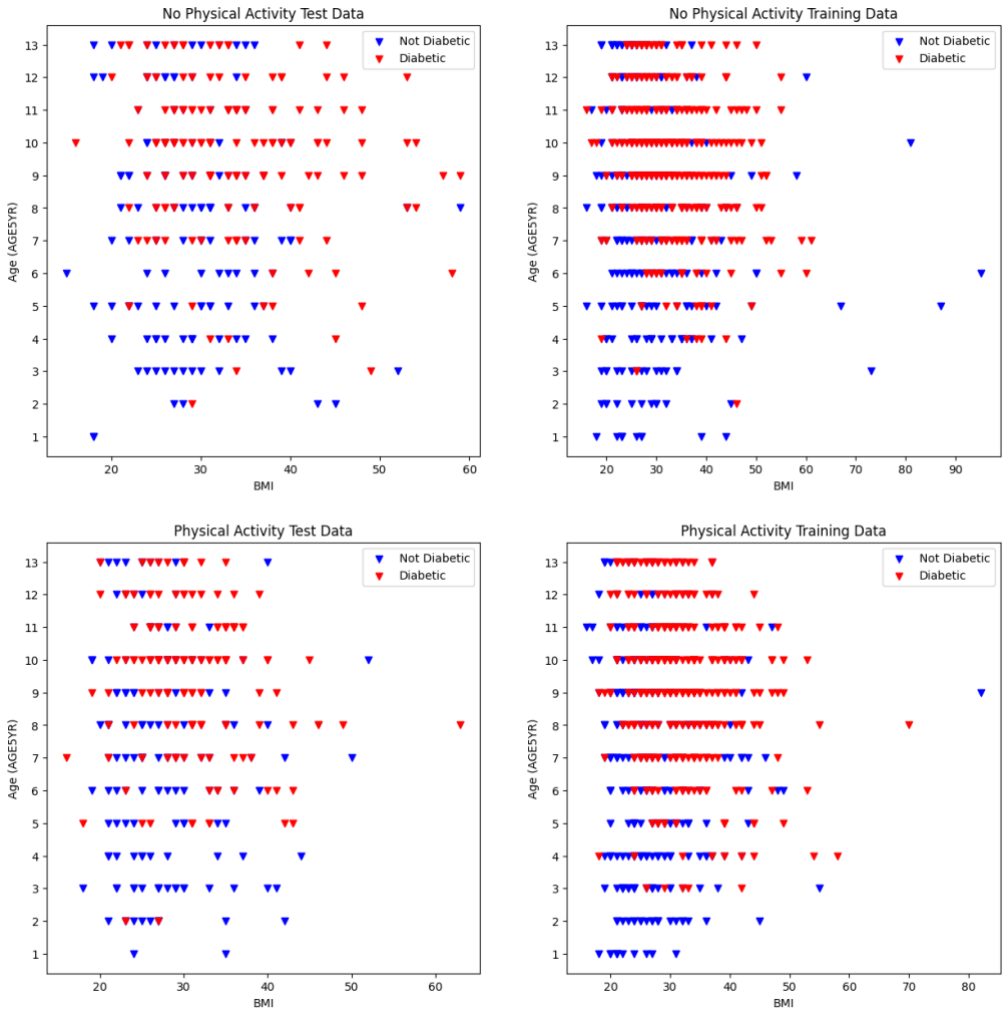
The function **pd.read_csv("file_name.csv")** was used to load and read the CSV files and they were stored as follows:

- NoActivity_training = pd.read_csv('diabetes_NoActivity_training.csv')
- NoActivity_test = pd.read_csv('diabetes_NoActivity_test.csv',)
- Activity_training = pd.read_csv('diabetes_PhysActivity_training.csv')
- Activity_testing = pd.read_csv('diabetes_PhysActivity_test.csv')

To split the features and class labels into separate variables, 'iloc' (Pandas) was used to split each of the 4 data frames into 8. The training and test sets and their corresponding labels were separated by using this.

iloc[:,0] was used for the labels as the first column of the data frames was the class label: Diabetes_binary and we needed all the rows from the first column as the function is iloc[x,y] where (x) are the rows and (y) are the columns. [:,0] means that we needed all the rows from the first column.

1b) Scatter-plot and findings



The figure above shows 4 plots for each of the datasets where the BMI is plotted on the x-axis against Age (AGE5YR) on the y-axis. The data points that are Diabetic or Pre-Diabetic are red whereas Not Diabetic are blue. The 'Test' scatter plots for both Physical Activity and No Physical Activity have fewer points as they are much smaller (300 vs. 700 entries). Comparing the two plots for "No Physical Activity Training Data" and "Physical Activity Training Data" there is no strong correlation between BMI and diagnoses for Diabetes but, Age does seem to have an impact with increased risk of diabetes. No Physical Exercise may or may not put an individual at an increased risk of diabetes but it is difficult to reach that conclusion, based on these graphs.

1c) The two training sets and two test sets are normalised separately using sk.StandardScaler( ) and the fit_transform(x) functions. This is done by first initializing a scaler object with scaler = sk.StandardScaler( ).

Next, the fit_transform( ) function is called on the scaler object with the frame of features as the input. The output is then stored in a separate variable for each of the normalised datasets.

Below are the mean and standard deviation values of the first feature (BMI) from the normalised test sets:

- Mean of No Physical Activity BMI-Test = 1.2434497875801754e-16
- Std. of No Physical Activity BMI-Test = 0.9999999999999998
- Mean of Physical Activity BMI-Test = -2.042810365310288e-16
- Std. of Physical Activity BMI-Test = 1.0

1d)

No Physical Activity Training Data

Principle Component	PC1	PC2	PC3	PC4	PC5	PC6	PC7
%age Variance Captured	31.26%	17.24%	15.17%	13.72%	8.88%	8.31%	5.41%

Physical Activity Training Data

Principle Component	PC1	PC2	PC3	PC4	PC5	PC6	PC7

%age Variance Captured	30.45%	17.53%	14.49%	13.85%	9.3%	8.27%	6.11%

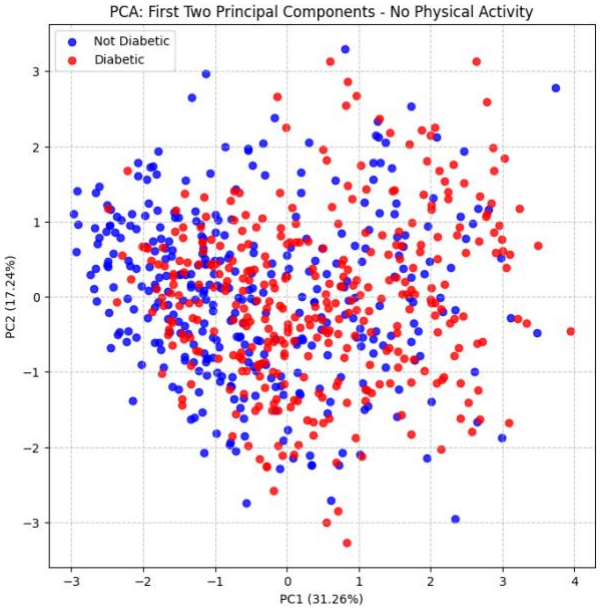


Figure 1

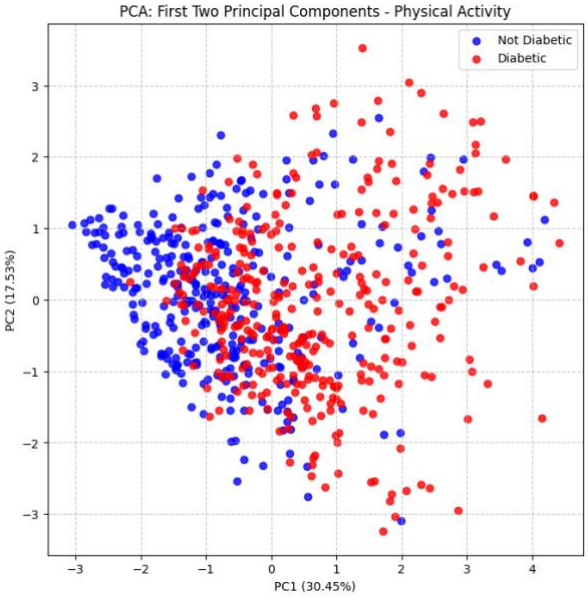


Figure 2

Figure 1 and Figure 2 capture 48.50% and 47.98% of the variance in the No Physical Activity and Physical Activity dataset, respectively. It can be gathered that even though PC1 is the most impactful principal component in both analyses, other principal components such as PC3 and PC4 might need to be considered to get a better understanding of the overall data.

It can also be seen that there is more of an emerging trend (clustering of 'not diabetic' data points) in Figure 2 and it could suggest that PC1 has a higher impact on the patient not being diagnosed as diabetic or pre-diabetic versus PC2. However, it can also be assumed based on the clustering for 'diabetic' data points that PC2 has a higher impact on the patient being diagnosed as 'diabetic' than PC1 does.

Task 2

2a) A 75:25 ratio was used to split the training, and validation sets and both the training and validation set were normalised using the StandardScalar( ) function and the fit_transform function. This was done by initialising a scaler object and using scaler.fit_transform( ).

Feature	**Train Mean**	**Train Std**	**Val Mean**	**Val Std**
BMI	31.617143	8.818575	31.057143	8.529674
GenH1th	3.304762	8.818575	7.274286	11.551691
MentH1th	5.533333	9.516625	7.274286	11.551691

PhysH1th	10.321905	12.388479	10.491429	12.459068
Age	8.668571	2.789132	8.777143	2.961737
Education	4.533333	1.079534	4.628571	1.090337
Income	4.889524	2.188447	5.000000	2.253988

Generally, the mean and standard derivation for the features in the training and validation sets are very similar if not identical. GenH1th and MentH1th have some noticeable variation, however.

2b,c) Using accuracy_score and classification_report from sklearn.metrics, the 3 different SVCs with the varied parameters were evaluated.

Model No.	[C , γ]	Validation Accuracy	Classification Precision: 'Diabetic'	Classification Precision: 'Not Diabetic'
Model 1	[1 , 1]	64.0%	58%	71%
Model 2	[5 , 0.5]	62.3%	57%	68%
Model 3	[0.5 , 0.05]	74.3%	72%	76%

Since Model 3 with [C, γ] = [0.5, 0.05] yielded the most accurate SVC out of the tested parameter values, these values have been selected for the final model.

2d) Model results:

[C , γ]	Validation Accuracy	Classification Precision: 'Diabetic'	Classification Precision: 'Not Diabetic'
[0.5 , 0.05]	69.3%	66%	73%

Confusion Matrix:

	Predicted: No Diabetes	Predicted: Diabetes
Actual: No Diabetes	95 (TN)	57 (FP)
Actual: Diabetes	35 (FN)	113 (TP)

Conclusions:

The model has moderate accuracy (indicated by higher values along the diagonal) when it comes to classifying the data, but it does have **more** trouble correctly predicting whether a patient is **diabetic** compared to predicting whether they are **not diabetic** as can be seen from the classification precision percentage. The higher value for false-positives could indicate that the model tends to over-predict diabetes. In comparison, values for false-negatives are lower and suggest that the model has a comparatively lower likelihood of wrongly predicting not diabetic when the patient is diabetic.

Task 3

3a) A 75:25 ratio was used to split the training, and validation sets and both the training and validation set were normalised using the StandardScalar( ) function and the fit_transform function. This was done by initialising a scaler object and using scaler.fit_transform( ).

Feature	Train Mean	Train Std	Val Mean	Val Std
BMI	29.900952	7.250540	29.325714	7.902580
GenH1th	2.880000	1.145386	2.942857	1.097093
MentH1th	4.649524	8.645423	5.422857	9.034642
PhysH1th	6.994286	10.601651	6.760000	9.359340
Age	8.005714	3.038237	8.097143	2.933505
Education	4.860952	1.049218	4.897143	0.983042
Income	5.447619	2.258913	5.308571	2.185818

Generally, the values for mean and standard deviation don't vary between the training and validation sets. There is a notable difference in the mean and standard deviation for GenH1th between the two sets, but it seems like an isolated occurrence.

3b,c) Using accuracy_score and classification_report from sklearn.metrics, the 3 different SVCs with the varied parameters were evaluated.

Model No.	[C , γ]	Validation Accuracy	Classification Precision: 'Diabetic'	Classification Precision: 'Not Diabetic'
Model 1	[1 , 1]	73.1%	74%	72%
Model 2	[5 , 0.5]	72.0%	75%	68%
Model 3	[0.5 , 0.05]	72.0%	74%	70%

Since Model 1 with [C, γ] = [1 , 1] yielded the most accurate SVC out of the tested parameter values and had a good balance of precision between the 'diabetic' and 'not diabetic' classification, these values have been selected for the final model.

3d) Model results:

[C , γ]	Validation Accuracy	Classification Precision: 'Diabetic'	Classification Precision: 'Not Diabetic'
[1 , 1]	69.0%	65%	75%

Confusion Matrix:

	Predicted: No Diabetes	Predicted: Diabetes
Actual: No Diabetes	88 (TN)	64 (FP)
Actual: Diabetes	29 (FN)	119 (TP)

Conclusions:

The model has moderate accuracy (indicated by higher values along the diagonal and the percentage for validation accuracy: 69.0%) when it comes to classifying the data. Based on the higher values for false-positives, it can be concluded that the model has a harder time predicting whether a patient is diabetic compared to not diabetic. It tends to over-predict diabetes, similar to the model for the No Physical Activity dataset. The precision for predicting 'not diabetic' versus 'diabetic' is higher (75% vs. 65%) and support this conclusion.

Task 4

4a)i) NoPhysActivity model + PhysActivity test set

The PhysActivity group testing data is normalised using the StandardScaler( ) from the sklearn library. It is normalised by using the mean and standard deviation from the PhysActivity Testing data.

ii) The model trained for the NoPhysicalActivity group in 2b was used with the parameters [C, γ] = [0.5, 0.05] as it was the most accurate out of the three models tested for that data with a validation accuracy of 69.3% after being fitted to the larger NoPhysActivity training dataset.

iii)

[C , γ]	Validation Accuracy	Classification Precision: 'Diabetic'	Classification Precision: 'Not Diabetic'
[0.5 , 0.05]	74.0%	71%	78%

Confusion Matrix:

	Predicted: No Diabetes	Predicted: Diabetes
Actual: No Diabetes	104 (TN)	48 (FP)
Actual: Diabetes	30 (FN)	118 (TP)

Based on the validation accuracy of the model, it can be gathered that it can correctly predict whether a patient is diabetic or not, around 74% of the time which is moderately accurate.

The classification precision results reveal that the model is better at accurately predicting a negative diabetes diagnosis than a positive one. The TN and TP values in the confusion matrix reveal that the model accurately identified 104 'not diabetic' and 118 'diabetic' cases respectively. The FN and FP values in the confusion matrix reveal that the model incorrectly predicted 30 'diabetic' and 48 'not diabetic' cases. Based on both the higher number of false-positives in the confusion matrix and lower value of classification precision for 'diabetic', it can be concluded that the model tends to over-predict 'diabetic' cases and because of it, has a higher number of false-positives. The model overall, is better at predicting that a patient is 'not diabetic' than it is at predicting that they are diabetic.

4b)i) PhysActivity model + NoPhysActivity test set

The NoPhysActivity group testing data is normalised using the StandardScaler( ) from the sklearn library. It is normalised by using the mean and standard deviation from the NoPhysActivity Testing data.

ii) The model trained for the PhysicalActivity group in 3b was used with the parameters [C, γ] = [1 , 1] as it was the most accurate out of the three models tested for that data with a validation accuracy of 69.0% after being fitted to the larger PhysActivity training dataset.

iii)

[C , γ]	Validation Accuracy	Classification Precision: 'Diabetic'	Classification Precision: 'Not Diabetic'
[1 , 1]	65.7%	63%	70%

Confusion Matrix:

	Predicted: No Diabetes	Predicted: Diabetes
Actual: No Diabetes	85 (TN)	67 (FP)
Actual: Diabetes	36 (FN)	112 (TP)

Based on the validation accuracy of the model, it can be gathered that it can correctly predict whether a patient is diabetic or not, around 66% of the time which is mildly accurate.

The classification precision results reveal that the model is better at accurately predicting a negative diabetes diagnosis than a positive one. The TN and TP values in the confusion matrix reveal that the model accurately identified 85 not diabetic and 112 diabetic cases respectively. The FN and FP values in the confusion matrix reveal that the model incorrectly predicted 36 diabetic and 67 not diabetic cases. Based on both the higher number of false-positives in the confusion matrix and lower value of classification precision for 'diabetic', it can be concluded that the matrix tends to over-predict 'diabetic' cases and because of it, has a higher number of false-positives. The model overall, is better at predicting that a patient is 'not diabetic' than it is at predicting that they are diabetic.

4d) Comparing the results of the two models, the model trained on the NoPhysActivity training dataset ends up performing better than the PhysActivity model when it comes to evaluating it on the other's test dataset. It has a higher validation accuracy (74.0% vs. 65.7%) and has proportionally lower values for false-positives and false-negatives than its counterpart.

The NoPhysActivity model's better performance can be attributed to the parameters set when initialising the SVC. Since it has lower values for C (0.5 vs. 1) and γ (0.05 vs. 1), the model fares better in a generalised scenario and suggests that the PhysActivity model was overfitted to the training dataset. This made it degrade in performance as soon as it introduced to values that it was previously unfamiliar with. Higher values for regularisation and kernel can more aggressively fit the model to the data and make it more sensitive to changes in the data points which is desirable in some cases but causes poor performance against generalised scenarios.

It should also be noted that the PCA for both the NoPhysActivity and PhysActivity training sets reveal some information about the patterns in the data. When plotting PC1 against PC2 for the Physical Activity training dataset, we can see a trend emerging. In comparison, there is no distinguishable trend that emerges in the for the graph of PC1 against PC2 for the No Physical Activity training dataset. This suggests that in the long run, a more generalised model will yield better results.