Model training outcome for predicting obesity levels

Abstract

For training a model with higher accuracy in predicting obesity level of a person with given life habits, we applied different methods on the UCI obesity dataset. The results are promising. Both the Random Forest and Neural Network models scored the highest accuracy, 0.95. The other model scores are LDA:0.889, QDA:0.018, Logistic: 0.806, Decision Tree: 0.871, SVM:0.936 and K-Means Clustering: 0.142.

Introduction

Through the years, obesity has become one of the biggest and the most dangerous health problem among the society. Obesity can induce different diseases such as diabete,hypertension and various cerebro-cardiovascular diseases, which can cause serious results or even further, deaths.

To help with preventing the worst situations happening, our group decided to build a model can successfully predict the obesity level of a person based on the information of his or her eating habits and physical condition.

The dataset we are going to use to train the model is downloaded from UCI Machine Learning Repository

(https://archive.ics.uci.edu/ml/datasets/Estimation+of+obesity+levels+based+on+eating+habits+and+physical+condition+)

This dataset contains 17 variables and 2111 observations. The variable related to obesity level is NObesity and it is categorized into 7 categories: Insufficient Weight, Normal Weight, Overweight Level I, Overweight Level II, Obesity Type I, Obesity Type II and Obesity Type III. The variables related to eating habits are: The attributes related with eating habits are: Frequent consumption of high caloric food (FAVC), Frequency of consumption of vegetables (FCVC), Number of main meals (NCP), Consumption of food between meals (CAEC), Consumption of water daily (CH20), and Consumption of alcohol (CALC).

The variables related to physical conditions are: Calories consumption monitoring (SCC), Physical activity frequency (FAF), Time using technology devices (TUE), Transportation used (MTRANS).

Methods

For training the model, we decided to use the following methods: Logistic Classification, Linear Discriminant Analysis, Quadratic Discriminant Analysis, Support Vector Machine, Decision Tree, Random Forest, Neural Network and a K-Means Clustering method as a comparison to the previous supervised learning methods.

Both Logistic and SVM are designed originally for binary classification. In order to complete the job for multiclass classification, we need to use one-versus-rest method, which splits the multiclass classification process into multiple binary classification processes.

Both QDA and LDA are approximating the Bayes classifier for the boundary to separate different features. LDA provides a linear decision boundary while QDA provides a quadratic decision boundary.

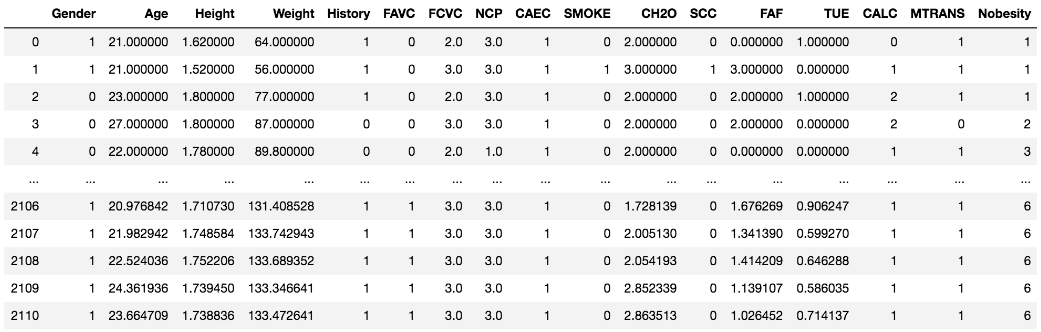
Decision Tree method uses a flowchart-like structure and makes separation divided predictor space to find out the result falling into this space with certain splitting algorithm. However, Decision Tree method is very easy to be overfitted. Therefore, Random Forest method is used. The randomness mechanisms in Random Forest method will help solve the overfitting problem.

Neural Network is a method that imitates the learning processes of human brains with hidden layers and neuron units.

K-Means Clustering is overall different from the previous method. It is an unsupervised machine learning process that separate observations into k clusters with the closest means.

Exploratory Data Analysis

Most of the categorical variable in the dataset is recorded in words. In order to make it suitable for the machine learning process, we re-coded those into ordinal variables with numbers stand for each level. (Details will be provided in supplementary document)



	Gender	Age	Height	Weight	History	FAVC	FCVC	NCP	CAEC	SMOKE	CH2O	SCC	FAF	TUE	CALC	MTRANS	Nobesity
0	1	21.000000	1.620000	64.000000	1	0	2.0	3.0	1	0	2.000000	0	0.000000	1.000000	0	1	1
1	1	21.000000	1.520000	56.000000	1	0	3.0	3.0	1	1	3.000000	1	3.000000	0.000000	1	1	1
2	0	23.000000	1.800000	77.000000	1	0	2.0	3.0	1	0	2.000000	0	2.000000	1.000000	2	1	1
3	0	27.000000	1.800000	87.000000	0	0	3.0	3.0	1	0	2.000000	0	2.000000	0.000000	2	0	2
4	0	22.000000	1.780000	89.800000	0	0	2.0	1.0	1	0	2.000000	0	0.000000	0.000000	1	1	3
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
2106	1	20.976842	1.710730	131.408528	1	1	3.0	3.0	1	0	1.728139	0	1.676269	0.906247	1	1	6
2107	1	21.982942	1.748584	133.742943	1	1	3.0	3.0	1	0	2.005130	0	1.341390	0.599270	1	1	6
2108	1	22.524036	1.752206	133.689352	1	1	3.0	3.0	1	0	2.054193	0	1.414209	0.646288	1	1	6
2109	1	24.361936	1.739450	133.346641	1	1	3.0	3.0	1	0	2.852339	0	1.139107	0.586035	1	1	6
2110	1	23.664709	1.738836	133.472641	1	1	3.0	3.0	1	0	2.863513	0	1.026452	0.714137	1	1	6

Figure 1. Recoded Dataset

After that, we checked the frequency of independent variable, which is the Nobesity in the dataset.

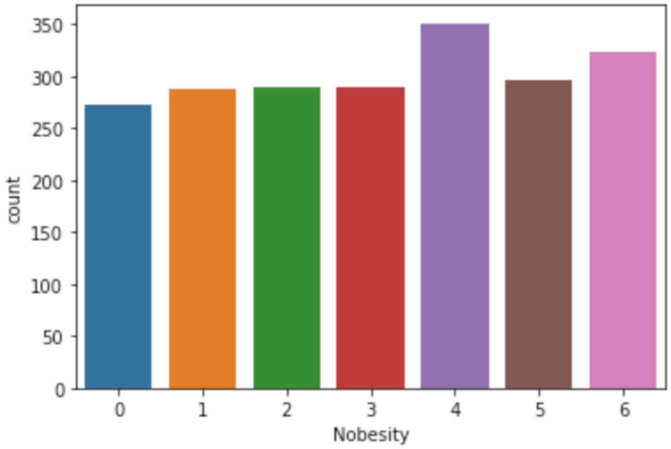
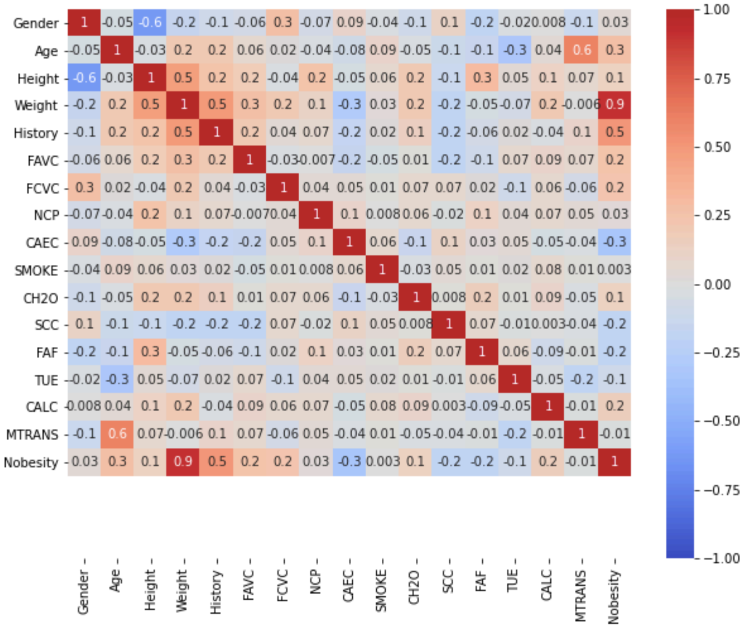


Figure 2. Frequency of Independent Variable

From the histogram we can tell that the data are overall balanced and none of them have extremely high or low frequency.



	Gender	Age	Height	Weight	History	FAVC	FCVC	NCP	CAEC	SMOKE	CH2O	SCC	FAF	TUE	CALC	MTRANS	Nobesity
Gender	1	-0.05	-0.6	-0.2	-0.1	-0.06	0.3	-0.07	0.09	-0.04	-0.1	0.1	-0.2	-0.02	0.008	-0.1	0.03
Age	-0.05	1	-0.03	0.2	0.2	0.06	0.02	-0.04	-0.08	0.09	-0.05	-0.1	-0.1	-0.3	0.04	0.6	0.3
Height	-0.6	-0.03	1	0.5	0.2	0.2	-0.04	0.2	-0.05	0.06	0.2	-0.1	0.3	0.05	0.1	0.07	0.1
Weight	-0.2	0.2	0.5	1	0.5	0.3	0.2	0.1	-0.3	0.03	0.2	-0.2	-0.05	-0.07	0.2	-0.006	0.9
History	-0.1	0.2	0.2	0.5	1	0.2	0.04	0.07	-0.2	0.02	0.1	-0.2	-0.06	0.02	-0.04	0.1	0.5
FAVC	-0.06	0.06	0.2	0.3	0.2	1	-0.03	-0.007	-0.2	-0.05	0.01	-0.2	-0.1	0.07	0.09	0.07	0.2
FCVC	0.3	0.02	-0.04	0.2	0.04	-0.03	1	0.04	0.05	0.01	0.07	0.07	0.02	-0.1	0.06	-0.06	0.2
NCP	-0.07	-0.04	0.2	0.1	0.07	-0.007	0.04	1	0.1	0.008	0.06	-0.02	0.1	0.04	0.07	0.05	0.03
CAEC	0.09	-0.08	-0.05	-0.3	-0.2	-0.2	0.05	0.1	1	0.06	-0.1	0.1	0.03	0.05	-0.05	-0.04	-0.3
SMOKE	-0.04	0.09	0.06	0.03	0.02	-0.05	0.01	0.008	0.06	1	-0.03	0.05	0.01	0.02	0.08	0.01	0.003
CH2O	-0.1	-0.05	0.2	0.2	0.1	0.01	0.07	0.06	-0.1	-0.03	1	0.008	0.2	0.01	0.09	-0.05	0.1
SCC	0.1	-0.1	-0.1	-0.2	-0.2	-0.2	0.07	-0.02	0.1	0.05	0.008	1	0.07	-0.01	0.003	-0.04	-0.2
FAF	-0.2	-0.1	0.3	-0.05	-0.06	-0.1	0.02	0.1	0.03	0.01	0.2	0.07	1	0.06	-0.09	-0.01	-0.2
TUE	-0.02	-0.3	0.05	-0.07	0.02	0.07	-0.1	0.04	0.05	0.02	0.01	-0.01	0.06	1	-0.05	-0.2	-0.1
CALC	-0.008	0.04	0.1	0.2	-0.04	0.09	0.06	0.07	-0.05	0.08	0.09	0.003	-0.09	-0.05	1	-0.01	0.2
MTRANS	-0.1	0.6	0.07	-0.006	0.1	0.07	-0.06	0.05	-0.04	0.01	-0.05	-0.04	-0.01	-0.2	-0.01	1	-0.01
Nobesity	0.03	0.3	0.1	0.9	0.5	0.2	0.2	0.03	-0.3	0.003	0.1	-0.2	-0.2	-0.1	0.2	-0.01	1

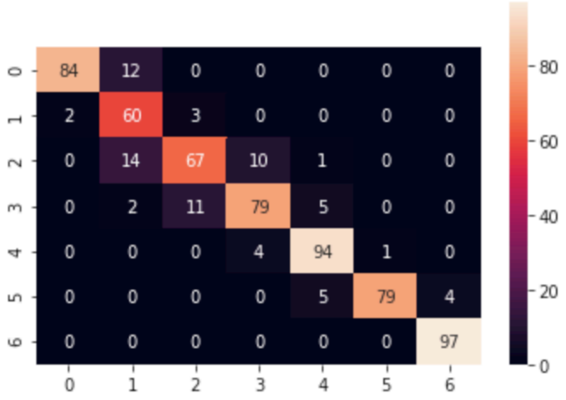
Figure 3. Correlation Check

We can find out that variable weight and Age are most correlated to the level of the obesity from the correlation heat map.

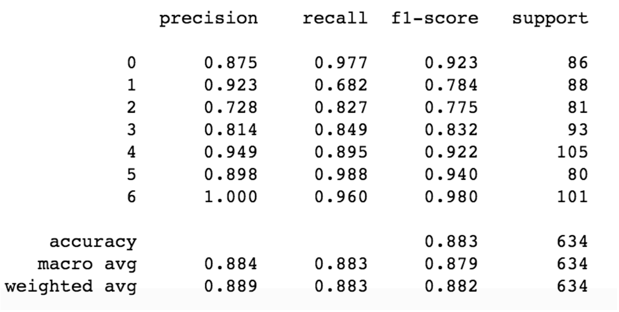
Modeling

We start the modeling process with splitting the dataset into training and testing sets with a ratio of 7:3. Then we used several different methods with training set and test the model with testing set to find out a model with the highest accuracy.

LDA



	0	1	2	3	4	5	6
0	84	12	0	0	0	0	0
1	2	60	3	0	0	0	0
2	0	14	67	10	1	0	0
3	0	2	11	79	5	0	0
4	0	0	0	4	94	1	0
5	0	0	0	0	5	79	4
6	0	0	0	0	0	0	97



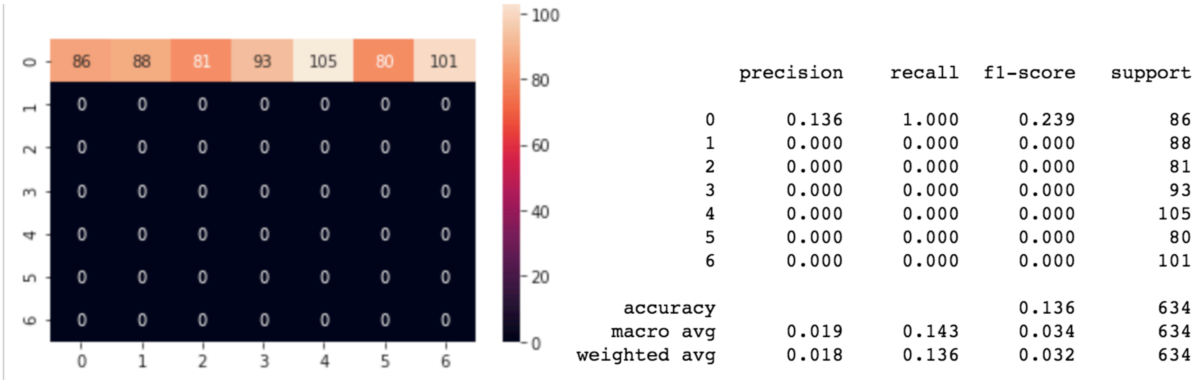
```
              precision    recall  f1-score   support

           0      0.875     0.977     0.923        86
           1      0.923     0.682     0.784        88
           2      0.728     0.827     0.775        81
           3      0.814     0.849     0.832        93
           4      0.949     0.895     0.922       105
           5      0.898     0.988     0.940        80
           6      1.000     0.960     0.980       101

    accuracy                          0.883       634
   macro avg      0.884     0.883     0.879       634
weighted avg      0.889     0.883     0.882       634
```

Figure 4. Confusion Matrix and Accuracy Score for LDA model

QDA



	0	1	2	3	4	5	6
0	86	88	81	93	105	80	101
1	0	0	0	0	0	0	0
2	0	0	0	0	0	0	0
3	0	0	0	0	0	0	0
4	0	0	0	0	0	0	0
5	0	0	0	0	0	0	0
6	0	0	0	0	0	0	0

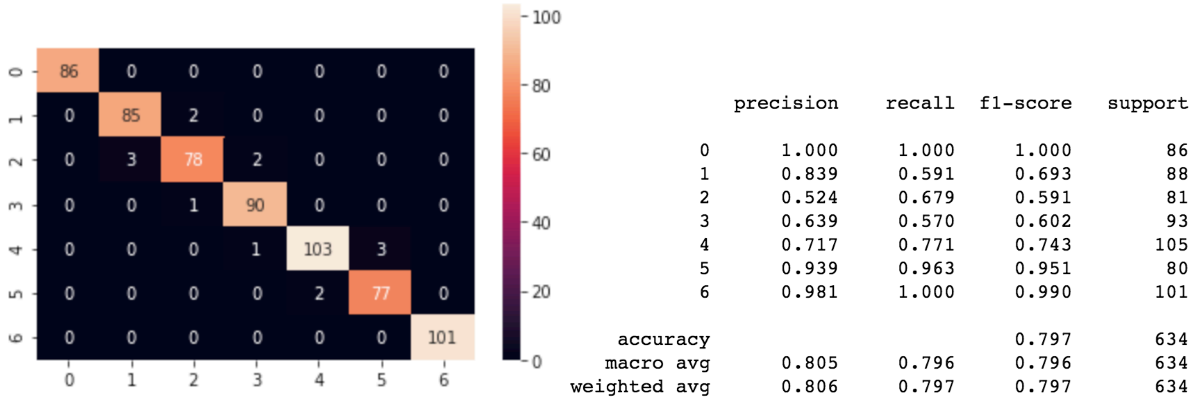
```
              precision    recall  f1-score   support

           0      0.136     1.000     0.239        86
           1      0.000     0.000     0.000        88
           2      0.000     0.000     0.000        81
           3      0.000     0.000     0.000        93
           4      0.000     0.000     0.000       105
           5      0.000     0.000     0.000        80
           6      0.000     0.000     0.000       101

    accuracy                          0.136       634
   macro avg      0.019     0.143     0.034       634
weighted avg      0.018     0.136     0.032       634
```

Figure 5. Confusion Matrix and Accuracy Score for QDA model

Logistic



	0	1	2	3	4	5	6
0	86	0	0	0	0	0	0
1	0	85	2	0	0	0	0
2	0	3	78	2	0	0	0
3	0	0	1	90	0	0	0
4	0	0	0	1	103	3	0
5	0	0	0	0	2	77	0
6	0	0	0	0	0	0	101

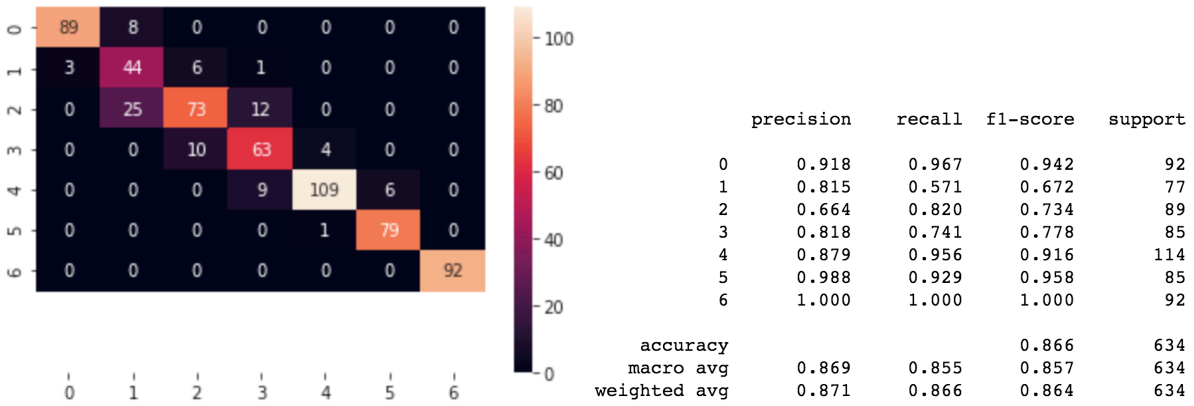
```
              precision    recall  f1-score   support

           0      1.000     1.000     1.000        86
           1      0.839     0.591     0.693        88
           2      0.524     0.679     0.591        81
           3      0.639     0.570     0.602        93
           4      0.717     0.771     0.743       105
           5      0.939     0.963     0.951        80
           6      0.981     1.000     0.990       101

    accuracy                          0.797       634
   macro avg      0.805     0.796     0.796       634
weighted avg      0.806     0.797     0.797       634
```

Figure 6. Confusion Matrix and Accuracy Score for Logistic model

Decision Tree



	0	1	2	3	4	5	6
0	89	8	0	0	0	0	0
1	3	44	6	1	0	0	0
2	0	25	73	12	0	0	0
3	0	0	10	63	4	0	0
4	0	0	0	9	109	6	0
5	0	0	0	0	1	79	0
6	0	0	0	0	0	0	92

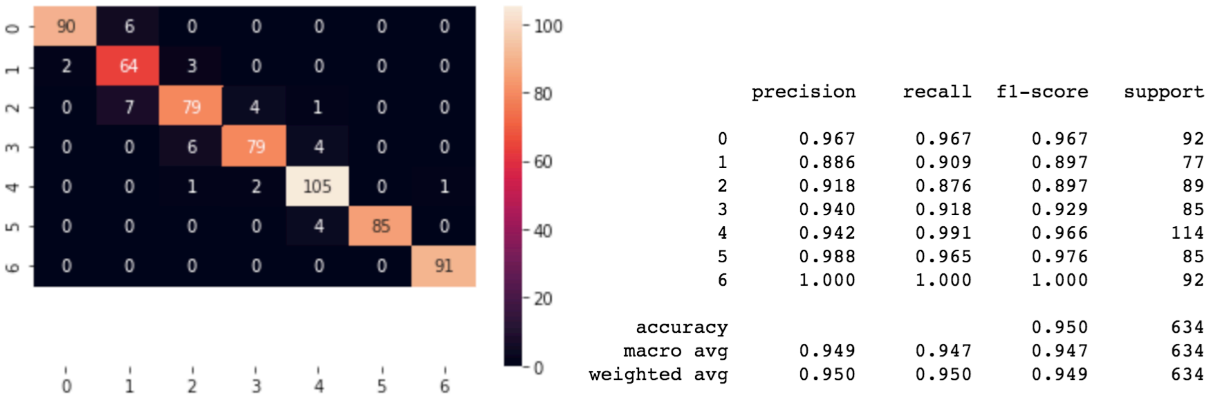
```
              precision    recall  f1-score   support

           0      0.918     0.967     0.942        92
           1      0.815     0.571     0.672        77
           2      0.664     0.820     0.734        89
           3      0.818     0.741     0.778        85
           4      0.879     0.956     0.916       114
           5      0.988     0.929     0.958        85
           6      1.000     1.000     1.000        92

    accuracy                          0.866       634
   macro avg      0.869     0.855     0.857       634
weighted avg      0.871     0.866     0.864       634
```

Figure 7. Confusion Matrix and Accuracy Score for Decision Tree

Random Forest



	0	1	2	3	4	5	6
0	90	6	0	0	0	0	0
1	2	64	3	0	0	0	0
2	0	7	79	4	1	0	0
3	0	0	6	79	4	0	0
4	0	0	1	2	105	0	1
5	0	0	0	0	4	85	0
6	0	0	0	0	0	0	91

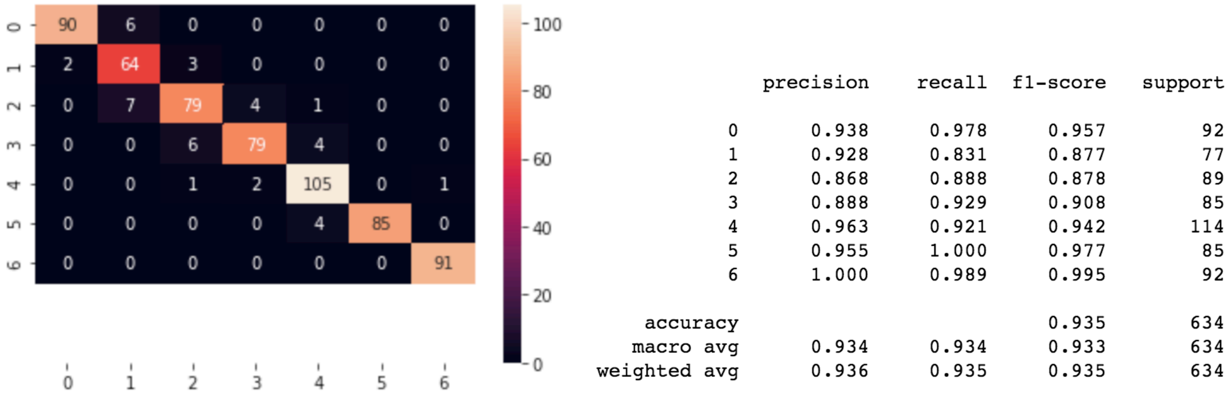
```
              precision    recall  f1-score   support

           0      0.967     0.967     0.967        92
           1      0.886     0.909     0.897        77
           2      0.918     0.876     0.897        89
           3      0.940     0.918     0.929        85
           4      0.942     0.991     0.966       114
           5      0.988     0.965     0.976        85
           6      1.000     1.000     1.000        92

    accuracy                          0.950       634
   macro avg      0.949     0.947     0.947       634
weighted avg      0.950     0.950     0.949       634
```

Figure 7. Confusion Matrix and Accuracy Score for Random Forest Model

Support Vector Machine



	0	1	2	3	4	5	6
0	90	6	0	0	0	0	0
1	2	64	3	0	0	0	0
2	0	7	79	4	1	0	0
3	0	0	6	79	4	0	0
4	0	0	1	2	105	0	1
5	0	0	0	0	4	85	0
6	0	0	0	0	0	0	91

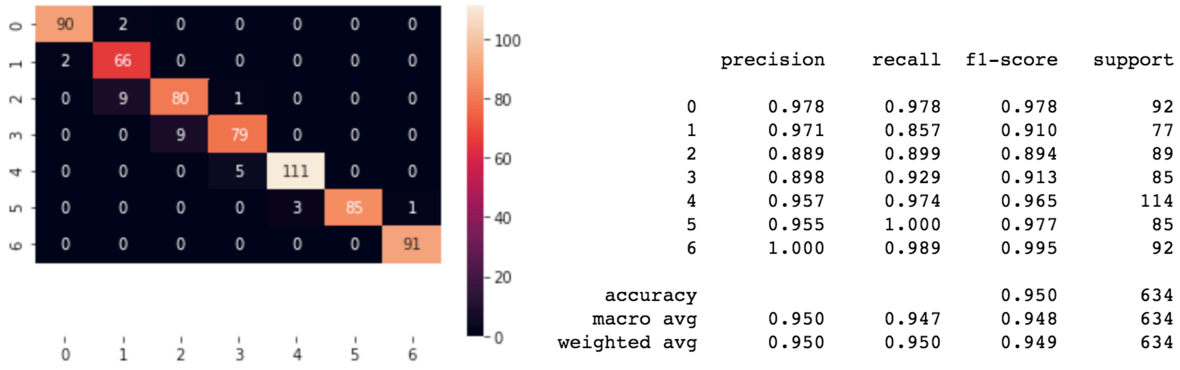
```
              precision    recall  f1-score   support

           0      0.938     0.978     0.957        92
           1      0.928     0.831     0.877        77
           2      0.868     0.888     0.878        89
           3      0.888     0.929     0.908        85
           4      0.963     0.921     0.942       114
           5      0.955     1.000     0.977        85
           6      1.000     0.989     0.995        92

    accuracy                          0.935       634
   macro avg      0.934     0.934     0.933       634
weighted avg      0.936     0.935     0.935       634
```

Figure 7. Confusion Matrix and Accuracy Score for SVM

Neural Network



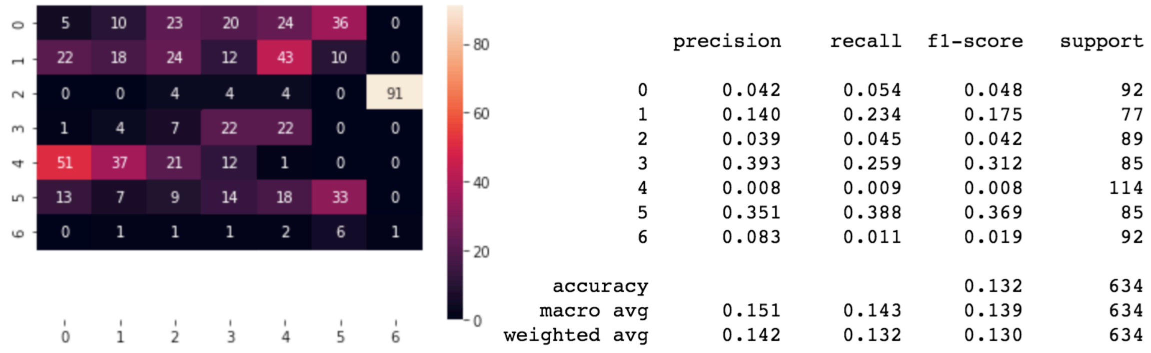
```
              precision    recall  f1-score   support

           0      0.978     0.978     0.978        92
           1      0.971     0.857     0.910        77
           2      0.889     0.899     0.894        89
           3      0.898     0.929     0.913        85
           4      0.957     0.974     0.965       114
           5      0.955     1.000     0.977        85
           6      1.000     0.989     0.995        92

    accuracy                          0.950       634
   macro avg      0.950     0.947     0.948       634
weighted avg      0.950     0.950     0.949       634
```

Figure 7. Confusion Matrix and Accuracy Score for Neural Network Model

Clustering (K-Means)



```
              precision    recall  f1-score   support

           0      0.042     0.054     0.048        92
           1      0.140     0.234     0.175        77
           2      0.039     0.045     0.042        89
           3      0.393     0.259     0.312        85
           4      0.008     0.009     0.008       114
           5      0.351     0.388     0.369        85
           6      0.083     0.011     0.019        92

    accuracy                          0.132       634
   macro avg      0.151     0.143     0.139       634
weighted avg      0.142     0.132     0.130       634
```

Figure 8. Confusion Matrix and Accuracy Score for K-Means Clustering

Discussion

Among all the methods, it is not surprised to see that K-Means Clustering is not performing well since clustering algorithms are not designed for doing classification. Besides K-Means, QDA has also not performed ideally in classifying this dataset. One possible guess is that the projection boundary is not even close to the shape of quadratic.

For the SVM, the linear kernel performs much better than radial kernel, which also partially verified the hypothesis above about the boundary problem.

For Neural Network, 5 layers with 4 neuron units performs the best. However, the current knowledge of mine could not give a proper explanation about the reason for choosing this combination.