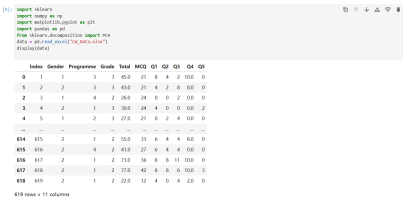
CW1 report

Enze Zhou, Student ID:2254411
TA:Zhejun.Yang

I. IMPORT

First, import all the packages that will be used for subsequent operations.
Read the file and display all the data.As shown in Figure [1].



	Index	Gender	Programme	Grade	Total	MCQ	Q1	Q2	Q3	Q4	Q5
0	1	1	3	3	45.0	21	8	4	2	10.0	0
1	2	2	3	3	43.0	21	4	2	0	8.0	0
2	3	1	4	2	26.0	24	0	0	2	0.0	0
3	4	2	1	3	30.0	24	4	0	0	0.0	2
4	5	1	2	3	27.0	21	0	2	4	0.0	0
...	...	...	...	...	...	...	...	...	...	...	...
614	615	2	1	2	55.0	33	6	4	4	8.0	0
615	616	2	4	2	41.0	27	6	4	4	0.0	0
616	617	2	1	2	73.0	36	8	8	11	10.0	0
617	618	2	1	2	77.0	42	8	8	6	10.0	3
618	619	2	1	2	22.0	12	4	0	4	2.0	0

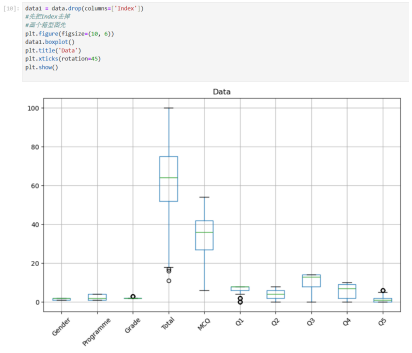
619 rows × 11 columns

[1]

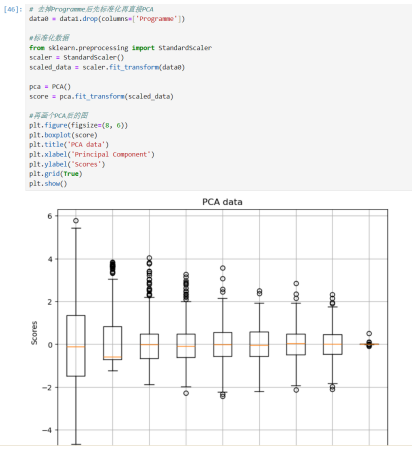
II. TASK1:BOX PLOT DRAWING AND RELATED PROCESSING

The key is to remove the index column from the beginning.Then the box plot was drawn and the distribution of the data was observed, and it was found that total and MCQ were affected more. We can use standardization to process data to reduce the impact of scale for the raw data.As shown in Figure [2],[3].

It is important to note that because of analyzing the distribution of the feature with association of the programme that a student is enrolled, we need to remove the programme to reduce the impact of overfitting.



[2]



[3]

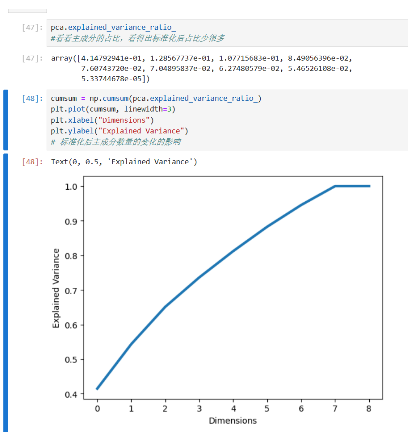
III. TASK2:PCA AND CHOOSE A SUITABLE SET OF COMPONENT

After standardization, use scikit-learn library for PCA processing and draw the PCA diagram.Since the principal component after PCA is the linear combination with the maximum variance after projection in the original data, the distribution degree of the data can be obtained from it. In addition, the principal component can be extracted by reducing and de-correlating. As shown in Figure [4].I also add pca.explained-variance-ratio to list the proportion of each component, and draw the proportion diagram of the change in the number of principal components and the diagram before and after PCA standardization, which is more convenient for me to observe the distribution of each component. It is found that the proportion of other principal components is roughly the same except the first and second principal components.

Then I drew the scatterplot of features to observe the distribution of the two features in the Programme class,As shown in Figure [5]. In addition, I made a table of principal component loading vectors and a distribution diagram of each principal component like figure [6], so as to observe the significance of contributions of each feature. I found that Q3 and Q5 contributed less, which was convenient for me to make decisions in subsequent operations.

Finally, I used different colors to represent different programmes and drew PCA scatter plots,as shown in Figure [7]. Through repeated comparisons, it was found that the second principal component and the seventh principal component could distinguish PCA results of different programees better than other combinations.Therefore,the set of components i found is the second and the seventh.

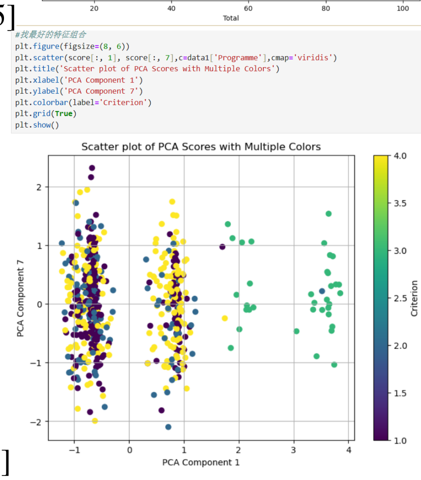
Moreover, by analyzing the variance ratio and observing the scatter plot, I found that grade and gender had the greatest effect on classification.In summary, the features I found were grade and gender.



[4]

[5]

[6]



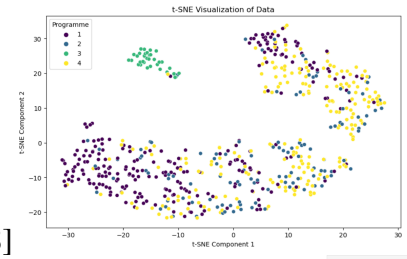
[7]

IV. Task3:t-SNE and Manual dimension reduction

I first carried out tsne dimensionality reduction, in which I used a more complex process, I chose to re-carry out PCA and standardization, and then carried out t-SNE algorithm to reduce dimensionality, and drew the scatter plot,like figure [8], and found that only Programme3 was perfectly separated.

Not satisfied with this effect, I conducted manual dimension reduction again, removed Q1,Q3 and Q5, and conducted t-SNE algorithm dimension reduction again. The final result still only separated Programme3, and the effect was only a little better than before. The possible reason was that the feature combination selected by me in PCA was not correct. Or there are still problems in the order and analysis of my tsne,as shown in Figure [9],[10].

```
# 将标准化后的数据转换为 DataFrame
data_scaled_df = pd.DataFrame(data_scaled)

# 应用 t-SNE 算法进行降维
tsne = TSNE()
tsne_data = tsne.fit_transform(data_scaled_df)

# 将 t-SNE 降维后的数据点绘制成散点图，根据 Programme 类别进行着色
plt.figure(figsize=(10, 6))
sns.scatterplot(x=tsne_data[:, 0], y=tsne_data[:, 1], hue=data['Programme'], palette='viridis')
plt.title('t-SNE Visualization of Data')
plt.xlabel('t-SNE Component 1')
plt.ylabel('t-SNE Component 2')
plt.legend(title='Programme')
plt.show()
```

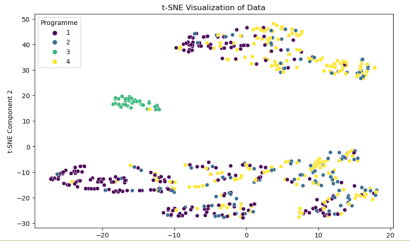


[8]

```
#手动降维
data2 = data0.drop(columns=['Q1'])
data3 = data2.drop(columns=['Q3'])
data4 = data3.drop(columns=['Q5'])

#标准化数据
from sklearn.preprocessing import StandardScaler
scaler = StandardScaler()
scaled_data = scaler.fit_transform(data4)
```

[9]

```
#tsne
tsne = TSNE(n_components = 2)
tsne_data = tsne.fit_transform(scaled_data)

# 将 t-SNE 降维后的数据点绘制成散点图，根据 Programme 类别进行着色
plt.figure(figsize=(10, 6))
sns.scatterplot(x=tsne_data[:, 0], y=tsne_data[:, 1], hue=data['Programme'], palette='viridis')
plt.title('t-SNE Visualization of Data')
plt.xlabel('t-SNE Component 1')
plt.ylabel('t-SNE Component 2')
plt.legend(title='Programme')
plt.show()

#为什么效果不一样？？？？
```



[10]

V. Task4:Visualise and compare raw features, scaled features, PCA features

Finally, I visualized all the original features, normalized features and PCA features for comparison and observation.

```
# 可视化原始特征
plt.figure(figsize=(12, 8))
for i, feature in enumerate(data0.columns):
    plt.subplot(2, 5, i+1)
    plt.hist(data0[feature], bins=20)
    plt.title('Raw Feature: {feature}')
    plt.xlabel('Value')
    plt.ylabel('Frequency')
plt.tight_layout()
plt.show()

# 可视化标准化特征
plt.figure(figsize=(12, 8))
scaled_data_df = pd.DataFrame(scaled_data)
for i, feature in enumerate(scaled_data_df.columns):
    plt.subplot(2, 5, i+1)
    plt.hist(scaled_data[:, i], bins=20)
    plt.title('Scaled Feature: {feature}')
    plt.xlabel('Value')
    plt.ylabel('Frequency')
plt.tight_layout()
plt.show()

# 可视化PCA特征
plt.figure(figsize=(8, 6))
plt.scatter(scores[:, 0], scores[:, 1], c=data1['Programme'], alpha=0.5)
plt.xlabel('Principal Component 1')
plt.ylabel('Principal Component 2')
plt.title('PCA Features Scatter Plot')
plt.grid(True)
plt.show()
```

[11]