DEPARTMENT OF PHYSICS & ASTRONOMY

PHAS0052 Individual Report Cover Sheet

Identifier: QQ

Group: 3A
Board Member: Mario Campanelli

Project Title: Assessing the grading capabilities of AI tools

Word Count: 1500

Contents

1 Introduction

From the introduction of this module, it was obvious that this module would require a focused delegation of tasks in the duration it would be carried out. The project assessed the marking capability of GPT-4 in various aspects. These included handwritten character-recognition, grading when the AI has no access to a mark-scheme, and when it is given access to a mark-scheme. The project also looked at the LLM's hint giving ability to evaluate it as a learning tool. In group 3A, I took on the role of communications officer and had the responsibility of establishing and maintaining contact with our board member. Then, I tackled a part of the data collection, and finally, I analysed the data we collected via multiple data analysis tools. The following report will provide a detailed look into my work in group 3A.

2 Communications Officer

As communications officer, my job consisted of contacting our board member (Prof. Mario Campanelli) to organise mandatory meetings where progress was discussed. These meetings were very useful for structuring our project as it made clear the expectations of the work that we were supposed to present. Updating everyone on the progress of the group was essential for avoiding high-stress situations as the completion of the project neared. To arrange these meetings, I had to talk with both my group members and the board member and find times that worked well with everyone. Getting the Risk and hazard forms signed by Prof. Campanelli was also part of my duty as communications officer. Finally, the purchase of GPT-4, the management of its account and expense were all part of my responsibility.

3 Data Collection and Analysis

With the analytical nature of project, understanding the implications of our data was essential. Our data was collected by finding three solutions for thirty questions; ten each from the Electromagnetism, Quantum Mechanics, and Classical Mechanics. This lead to a total on ninety solutions. The decision for our data to be structured this way was finalised by member 'YY' and myself. We decided that having thirty questions and ninety solutions would allow for a good number of data-points for deeper analysis.

3.1 Data Collection

The ninety solutions were marked in three ways. Once by GPT-4, where it was not provided a mark-scheme, once again by GPT-4, with a mark-scheme, and finally by two humans (with the mark-scheme).

I was one of the markers and marked the ninety solutions by referring to the mark-scheme. This was rather pain-staking, especially for questions where GPT-4 had made multiple errors. As the word limit for answers was kept controlled, GPT-4 often answered shorter questions with levels of detail they did not require. This made grading questions with multiple errors a high-focus task in order to not miss small steps the mark-scheme asked for. It was important that these human grades were collected, as it created a way for us to compare how GPT-4 would compare to the solution to marking that we use today, i.e., human-marking.

Once the GPT grades without the mark-scheme(GPT NM), the GPT grades with the mark-scheme(GPT M) and the human grades(HG) had been collected, data analysis could be performed.

3.2 Data Processing

There was much discussion within the group on how the data we would collect could be analysed. The completed literature review had suggested that the existing research had evaluated data mostly qualitatively. Thus, our efforts were focused on ways of quantitatively analysis. Regression plots were the obvious first choice, also being present in the literature. My work on these is explained in 3.2.2.

Further literature review revealed to me an idea that had not been previously explored: k-means clustering. By grouping questions that were graded similarly, I could gain insight into which questions GPT-4 graded similarly; the qualities of questions that could predict the behaviour of GPT-4.

However, before any of this analysis could be done, the data had to be prepared in a way that would make this analysis most efficient.

3.2.1 Data Preparation

We had thirty different questions and three different solutions for each question. These three solutions were found three times. The trials for each solution were averaged to give a total of ninety data points. The different trials were also used to provide a standard deviation of the marks per solution. The marks achieved and standard deviation of the mark were both normalised by dividing by the maximum mark achievable on the particular question. There were three different types of marking. I compiled the averages for these data points and standard deviations and normalised all the data so that it could then be used for analysis.

3.2.2 Regression and Results

I plotted regression plots for all permutations of our data: plotting HG against GPT M (figure 1a), GPT M against GPT NM (figure 1b), and HG against GPT NM (figure 1c). The most interesting result noticed from these plots was how the addition of the mark-scheme pushed the best-fit line towards the ideal-fit line. To quantify the relation between the variables, I found the Pearson coefficients and p-values for all the trends investigated in the plots.

For GPT-4 without mark-scheme[GPTNM] against GPT-4 with mark-scheme[GPTM], the correlation coefficient was 0.85 and p-value was $5.2 * 10^{-26}$. For human-graded[HG] against GPT-4 with mark-scheme the correlation coefficient was 0.79 and p-value was $3.4 * 10^{-20}$. Finally, for human-graded against GPT-4 without mark-scheme the correlation coefficient was 0.72 and p-value was $8.7 * 10^{-16}$. These coefficients and p-values suggested that while adding the mark-scheme certainly helped GPT-4 align itself with expected performance, it did not stop using what it "thought" was correct to grade the questions.

3.2.3 Clustering and Results

When discussing ways to analyse the group's data, I saw that existing literature had used heat maps and dendrograms to cluster questions.[1] I was familiar with clustering methods and thought that

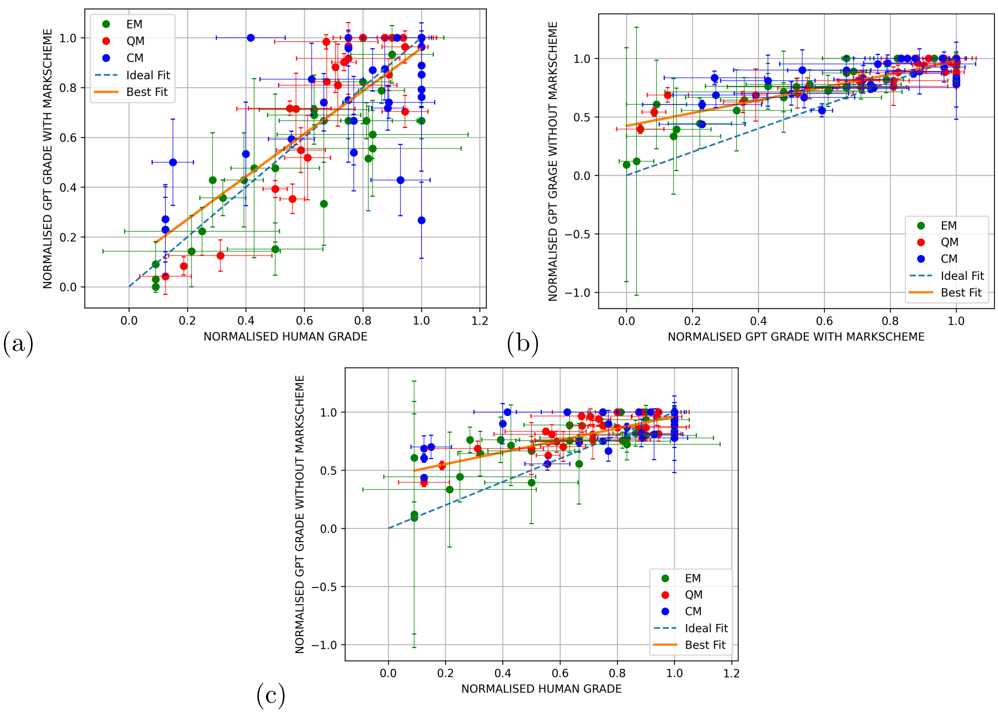


Figure 1: Regression plots for (a) GPTM vs HG (b) GPTNM vs GPTM (c) GPTNM vs HG

using this form of hierarchical clustering would not work well for our project due to the size of our data. It would also be difficult to analyse, given that the clusters would had to be chosen manually. I opted to use k-means clustering instead as it would result in clear clusters that could be analysed.

Before I started working on analysing the data, I made a presentation to explain to my group the clustering methods available to us, and my reasoning for using k-means clustering. I also explained how data would be prepared for this clustering, as well as how k-means actually worked. An idea suggested within the group was the use of dimensionality reduction to analyse our data. None of us were very familiar with this, however, I decided to take on the task as I understood the concept mathematically. Therefore, in this presentation I also included the dimensionality reduction tools we could use, finalising on t-SNE after listing the tools' advantages and disadvantages.

I gave a lot of thought to the data that should be included to the table that would be reduced using t-SNE, as I wanted to avoid redundancy. I decided that reducing the three different marking types would be the best course of action. I plotted elbow plots, and the clustered plots for the existing 2D plots as well as the t-SNE plot. Figure 2 shows the clustered plots.

Knowing what points belonged to which clusters I was able to analyse this data thoroughly.

First, I calculated the subtopic count within each cluster as subtopics were a major distinction in the data. I found that quantum mechanics and classical mechanics were graded similarly, while GPT-4 struggled a lot more with electromagnetic theory questions. Next, I calculated what solu-

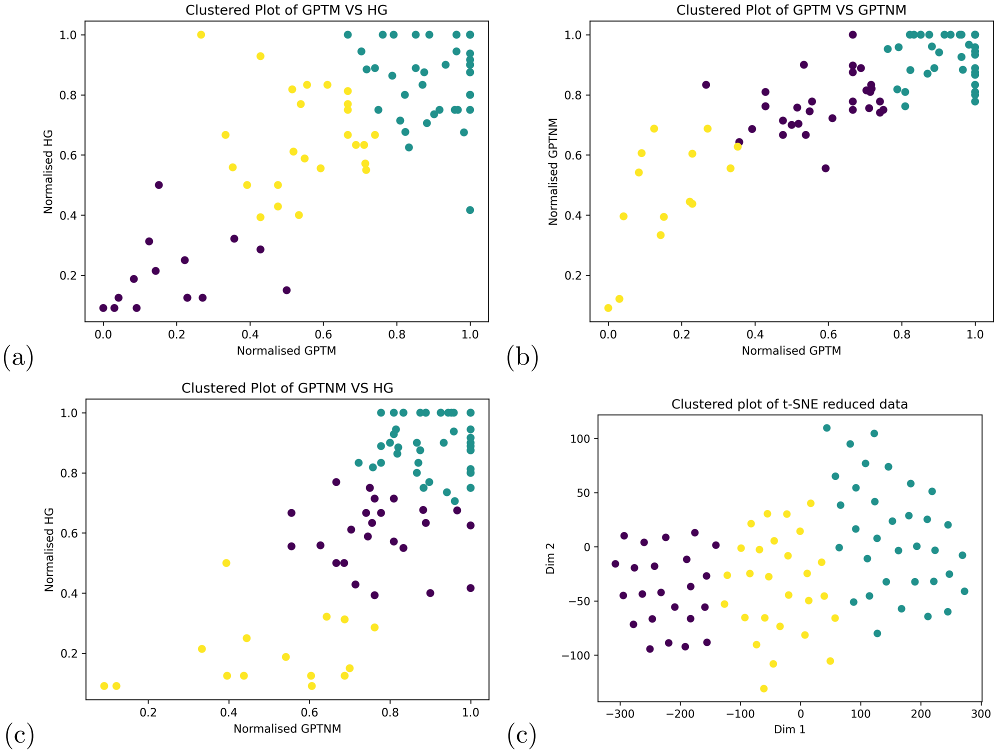


Figure 2: Cluster plots for (a) GPTM vs HG (b) GPTNM vs GPTM (c) GPTNM vs HG (d) t-SNE Reduced Data

tions had been grouped together, and found that it was most common for all solutions to be in the same clusters. More interestingly, the first and third solution were noticed to be in one cluster more frequently the first and second or second and third solutions. This suggested cyclicity in GPT-4 responses. Finally, I calculated what solutions were not clustered with the rest of the solutions for the same questions, and whether these solutions received a higher or lower grade. I found that every time solution 3 was clustered separately, it scored lower, suggesting a worsening performance with iteration.

4 Conclusions and Reflections

To conclude, the results found by the data-analysis I conducted not only increased the rigour of the project, but also brought about results not explored in existing literature. Furthermore, the analysis provided results that could be applied to understand how GPT-4 reacts to prompts generally, which can be used for many extensions to this and other problems related to LLMs. Having never practically applied k-means clustering or t-SNE before, having the opportunity to apply it to a real life case was very useful for professional development.

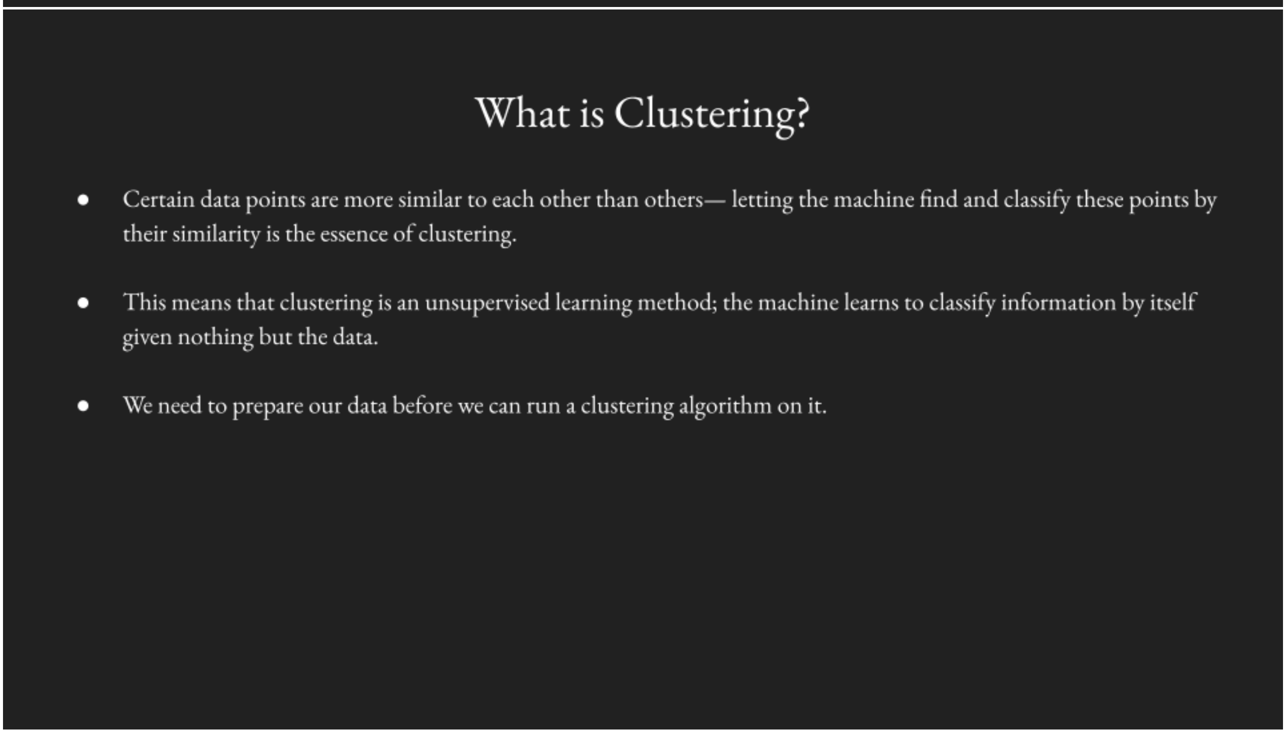
As the communications officer, I was constantly in contact with the entire team. This recognised position helped me collaborate well with my group. I am not often placed in high-communicating roles, making the experience of a role where I had to talk to my group members, my board member, as well as the communication officer of 3B, very valuable.

References

[1] Kortemeyer Gerd. Can an AI-tool grade assignments in an introductory physics course?. Phys. Rev. Phys. Educ. Res. [online]. 2023;19. [13/03/2024]. Available from:https://arxiv.org/abs/2304.11221

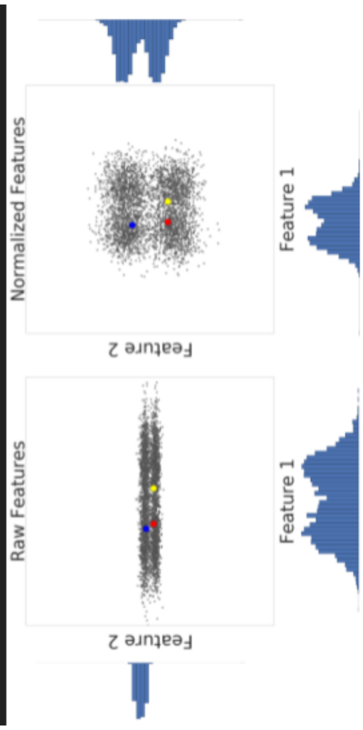
5 Appendix

A Clustering Presentation



Preparing Data

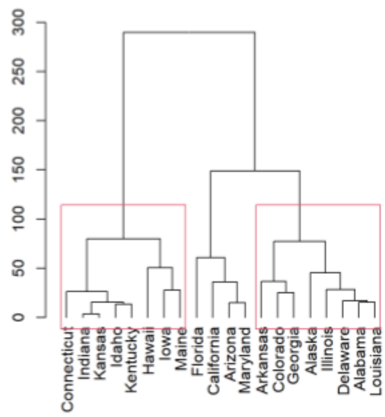
- Before implementing any algorithms, we need to decide what sort of data we will be collecting:
  - What data will be numerical?
  - Will we have any categorical data? [This would add dimensions as categorical data will require one-hot encoding]
  - What exactly are we looking for when clustering our data? This will help decide what algorithm we use as well as the amount of clusters we should be searching for.
  - Once data has been collected it will need to be normalised. The image shows intuitively how clusters are more easily recognisable in normalised data.



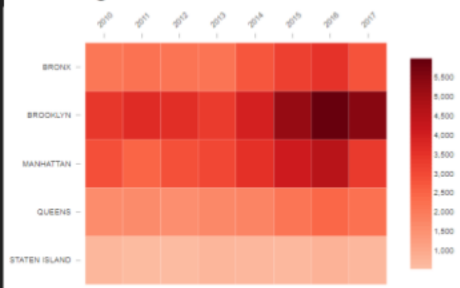
Hierarchical Clustering

- Hierarchical Clustering compares every data point to other data points and sorts itself compared to what points are most similar to each other. Through this it clusters similar points and then can be represented in many ways.
- Possible visualisation methods through this method are heat maps and dendrograms. [Personally, I find these clunky and hard to read—we may not figure out any real clusters through this. I have put some pictures for reference anyway.]
- This would use all the data that is collected through our questions.

Dendrogram



Heat map



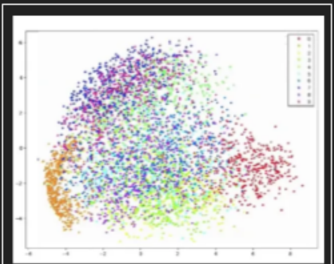
Dimensionality Reduction

- Our data will almost certainly be multi-dimensional (more than 3 dimensions). Therefore, we need a way to reduce these dimensions. This will not only help in the clustering of our data through other methods, but also in the data visualisation that can be then be included in our final poster.
- Two viable ways we can explore reducing data dimensionality:
  - PCA [Principal Component Analysis]
  - t-SNE [t-distributed Stochastic Neighbor Embedding]

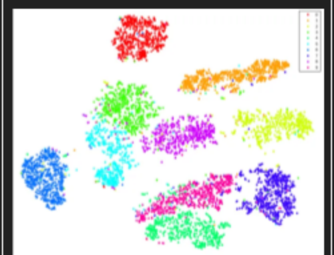
PCA and t-SNE

- Principal Component Analysis [PCA]
  - PCA creates components that account for a decreasing amount of variance in the data.
  - We would either visualise in two or three dimensions—therefore using two or three PCA components. If there is use for a scree plot (showing how much variation is accounted for by each component), it can be another visual added to the poster.
- t-distributed Stochastic Neighbor Embedding [t-SNE]
  - t-SNE's main focus is conserving the clustering of points as it maps the points to a lower dimension. It does this via the use of t-distribution to measure similarity.
  - It is objectively the better dimensionality reduction method between PCA and t-SNE but may be more difficult to implement. While libraries to perform it exist, it contains many parts that we may want to experiment with for the results we want.

MNIST



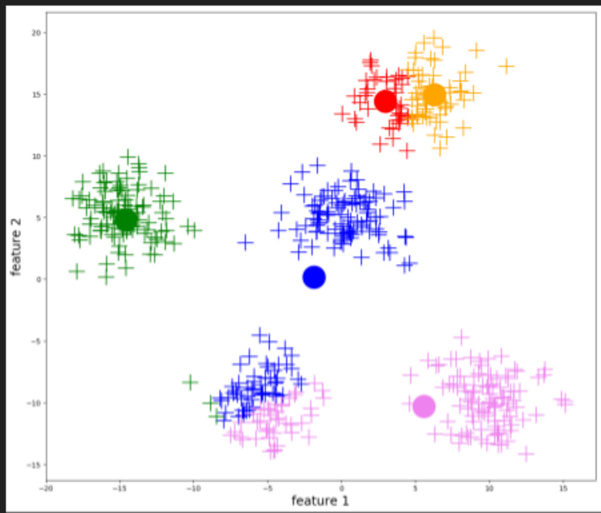
PCA



t-SNE

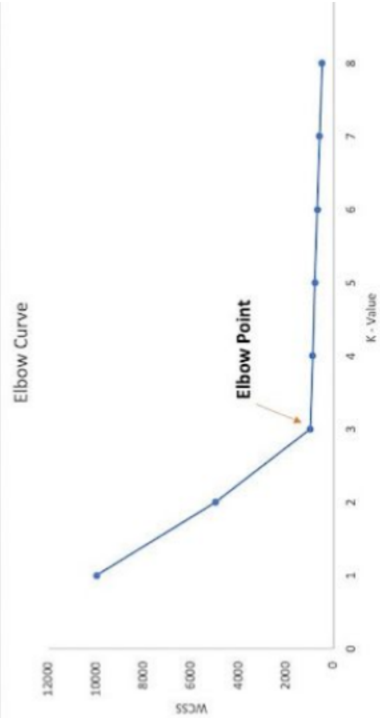
K-Means Clustering

- Once we have two or three dimensional plot, we can cluster the points using K-means clustering. A brief overview of the process is given below:
    - Choose K: the amount of clusters we expect to see in the data.
    - Initiate one point on the plot for each cluster and using euclidean distance assign the points closest to the cluster origin to the respective cluster.
    - Take the mean of the points in all clusters—this point is now the origin of the clusters.
    - Repeat the algorithm till the cluster origin positions do not change.



K-Means Clustering

- There are a few more things to be discussed with K-means clustering:
    - A very obvious issue are bad initialisation points. There is no way to solve this except running the algorithm multiple times. Adding up the variation of every cluster would give us a good idea on how well our data has been clustered.
    - Choosing the correct value of K can be done in one of two ways.
        - If we know the clusters we are looking for we will only run the algorithm with one K value, as any other values of K will be nonsensical.
        - If we are simply looking for clusters that could be open to interpretation, we could implement an elbow graph. This graph plots the reduction in variance as the value of K is increased. When K = number of points, $\sigma^2 = 0$, as every cluster has only one point, and therefore no variance. While reducing K will continuously reduce variance, when the variance stops dropping significantly we can say that the optimal number of clusters has been reached.



B Code

Outline

Data that we must examine include:

- Human Graded
- GPT Graded (No Markscheme)
- GPT Graded (Markscheme)

This notebook will explore this data through two seperate methods:

1. Regression Plots

- We have two options: Multiple 2D Plots, or a 3D Plot. 3D Plots are often hard to read and may not be able to provide much insight into the data we present. As such I will make regression plots for all permutations of our data, as well as calculating a r^2 and p-value for all plots. Conclusions will then be made depending on what is found.

2. Clustering Plots

- Once again, like above we can implement both 2D and 3D Clustering plots. However, instead of making a 3D Plot, I will reduce dimensionality using PCA or t-SNE. I do not believe the reduced cluster should be the only one we examine, instead we should—just like with the regression—create a cluster with all permutations of data. This will allow us to catch clusters that may otherwise be "contaminated" by the addition of more data. Obviously all of these will produced with their own elbow plots.

Thus, we have a total of:

- Regression Plots: 3
  - GPT M vs GPT NM
  - GPT M vs HUMAN
  - GPT NM vs HUMAN
- Clustering Plots: 4
  - GPT M vs GPT NM
  - GPT M vs HUMAN
  - GPT NM vs HUMAN
  - GPT M vs GPT NM vs HUMAN (Reduced)

7 plots.

In [1]:
```
#If you run this yourself, make sure all libraries are installed and the csv file is in the same
#import libraries
import numpy as np
import pandas as pd
import matplotlib as mlp
from matplotlib import pyplot as plt
mlp.rcParams['figure.dpi'] = 300
import scipy.stats
from sklearn.cluster import KMeans
from sklearn.manifold import TSNE
```

In [2]:
```
#import data as pandas df
EM = pd.read_csv("EM.csv")
QM = pd.read_csv("QM.csv")
CM = pd.read_csv("CM.csv")
```

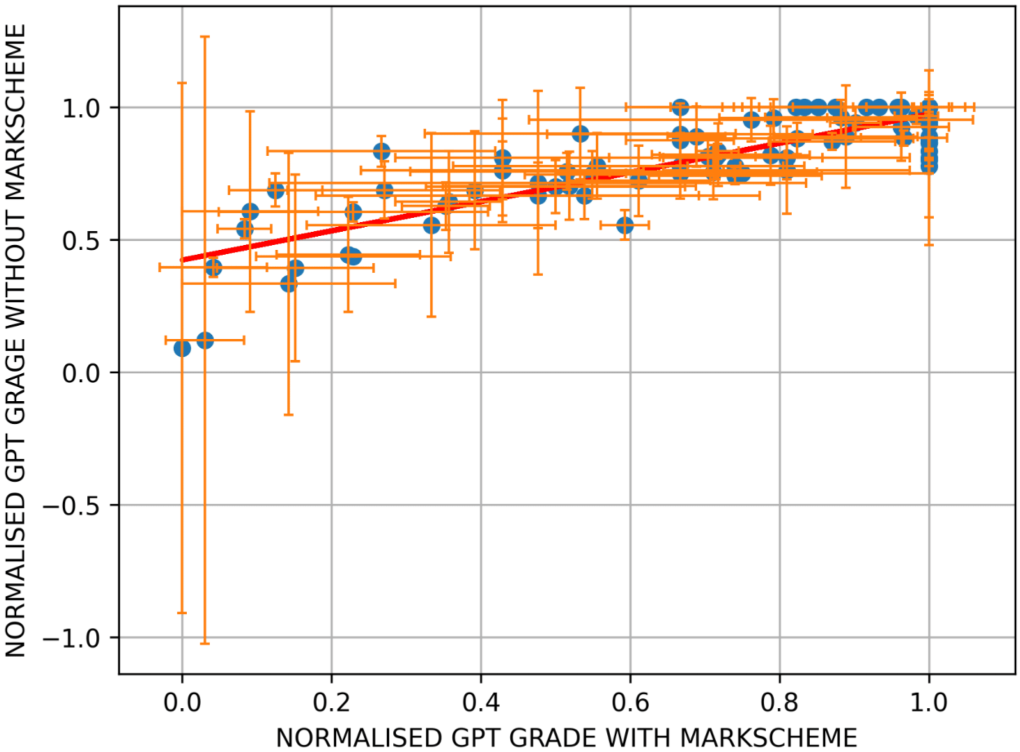
In [3]:
```
list(EM.columns) #checking header titles
```

Out[3]:
```
['Question',
 'Total_Marks',
 'Human_Grade',
 'GPT_Grade_NM',
 'GPT_Grade_M',
 'Normal_HG',
 'Normal_GPT_NM',
 'Normal_GPT_M',
 'std_dev_HG',
 'std_dev_GPT_NM',
 'std_dev_GPT_M',
 'Norm_std_dev_HG',
 'Norm_std_dev_GPT_NM',
 'Norm_std_dev_GPT_M']
```

In [4]:
```
data = pd.concat([EM, QM, CM]) #combining all data
```

Regression GPT Markscheme vs GPT No Markscheme

In [5]:
```
# Plot data and set labels
plt.plot(data['Normal_GPT_M'], data['Normal_GPT_NM'], 'o')
plt.xlabel('NORMALISED GPT GRADE WITH MARKSCHEME')
plt.ylabel('NORMALISED GPT GRAGE WITHOUT MARKSCHEME')

# Fit the trend line
z = np.polyfit(data['Normal_GPT_M'], data['Normal_GPT_NM'], 1)
p = np.poly1d(z)
plt.plot(data['Normal_GPT_M'],p(data['Normal_GPT_M']), color='red', linewidth=2)
plt.errorbar(data['Normal_GPT_M'],data['Normal_GPT_NM'], yerr=data['Norm_std_dev_GPT_NM'], xerr=(
# Show the plot
plt.grid()
plt.show()
```



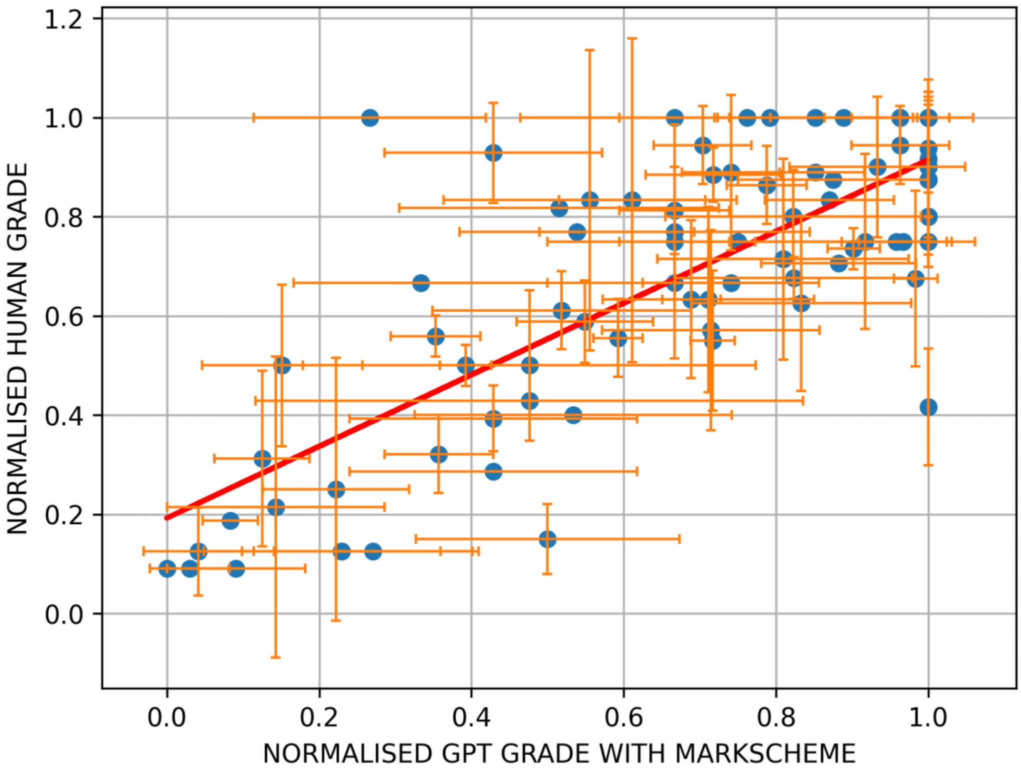
In [6]:
```
scipy.stats.pearsonr(data['Normal_GPT_M'], data['Normal_GPT_NM']) #(pearson coefficient, p-value,
```

Out[6]:
```
PearsonRResult(statistic=0.84815745899633, pvalue=5.199391569844949e-26)
```

13

Regression GPT Markscheme vs Human Graded

In [7]:
```
# Plot data and set labels
plt.plot(data['Normal_GPT_M'], data['Normal_HG'], 'o')
plt.xlabel('NORMALISED GPT GRADE WITH MARKSCHEME')
plt.ylabel('NORMALISED HUMAN GRADE')

# Fit the trend line
z = np.polyfit(data['Normal_GPT_M'], data['Normal_HG'], 1)
p = np.poly1d(z)
plt.plot(data['Normal_GPT_M'],p(data['Normal_GPT_M']), color='red', linewidth=2)
plt.errorbar(data['Normal_GPT_M'],data['Normal_HG'], yerr=data['Norm_std_dev_HG'], xerr=data['No
# Show the plot
plt.grid()
plt.show()
```



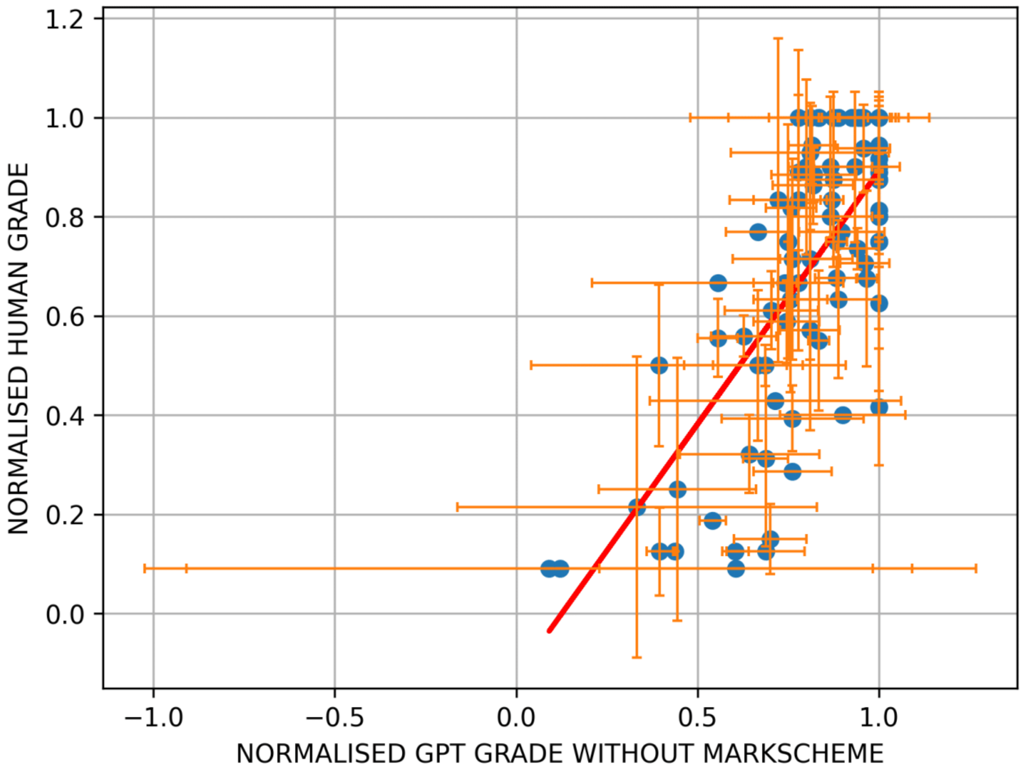
In [8]:
```
scipy.stats.pearsonr(data['Normal_GPT_M'], data['Normal_HG']) #(pearson coefficient, p-value)
```

Out[8]:
```
PearsonRResult(statistic=0.7875491758678023, pvalue=3.372892504233281e-20)
```

14

Regression GPT No Markscheme vs Human Graded

In [9]:
```
# Plot data and set labels
plt.plot(data['Normal_GPT_NM'], data['Normal_HG'], 'o')
plt.xlabel('NORMALISED GPT GRADE WITHOUT MARKSCHEME')
plt.ylabel('NORMALISED HUMAN GRADE')

# Fit the trend line
z = np.polyfit(data['Normal_GPT_NM'], data['Normal_HG'], 1)
p = np.poly1d(z)
plt.plot(data['Normal_GPT_NM'],p(data['Normal_GPT_NM']), color='red', linewidth=2)
plt.errorbar(data['Normal_GPT_NM'],data['Normal_HG'], yerr=data['Norm_std_dev_HG'], xerr=data['No
# Show the plot
plt.grid()
plt.show()
```



In [10]:
```
scipy.stats.pearsonr(data['Normal_GPT_NM'], data['Normal_HG']) #(pearson coefficient, p-value)
```

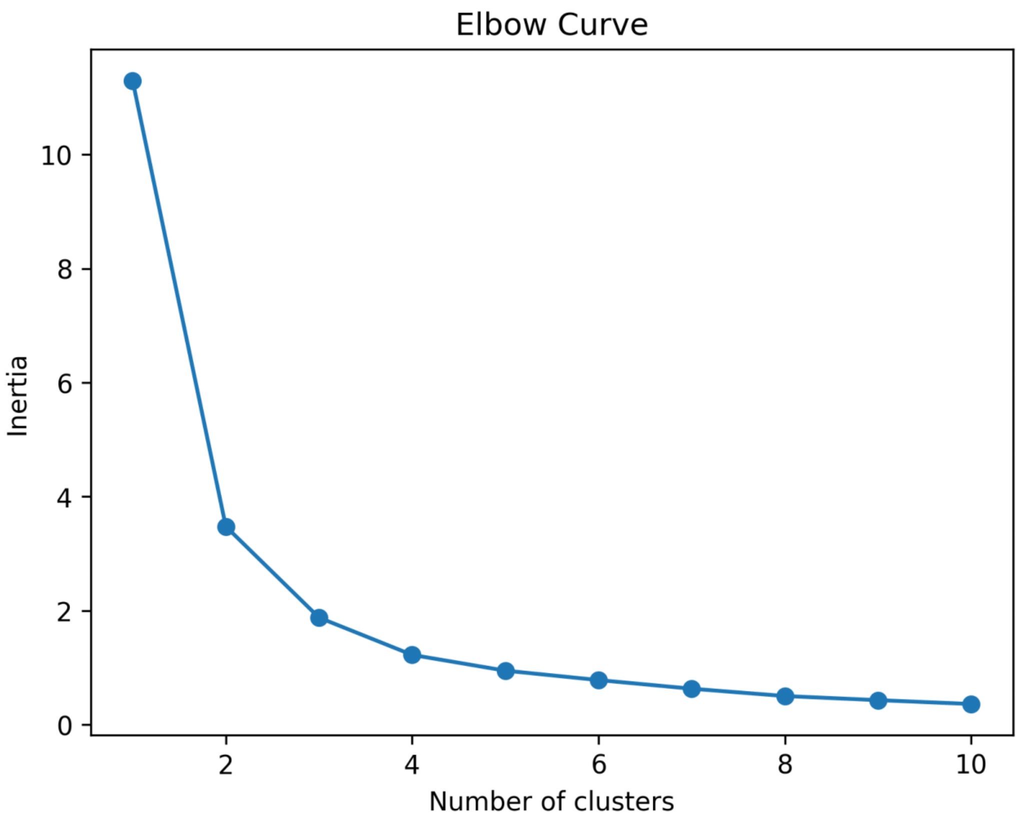
Out[10]:
```
PearsonRResult(statistic=0.722863983059397, pvalue=8.682354923167096e-16)
```

The order of correlation was [NM vs HG] < [M vs HG] < [NM vs M]

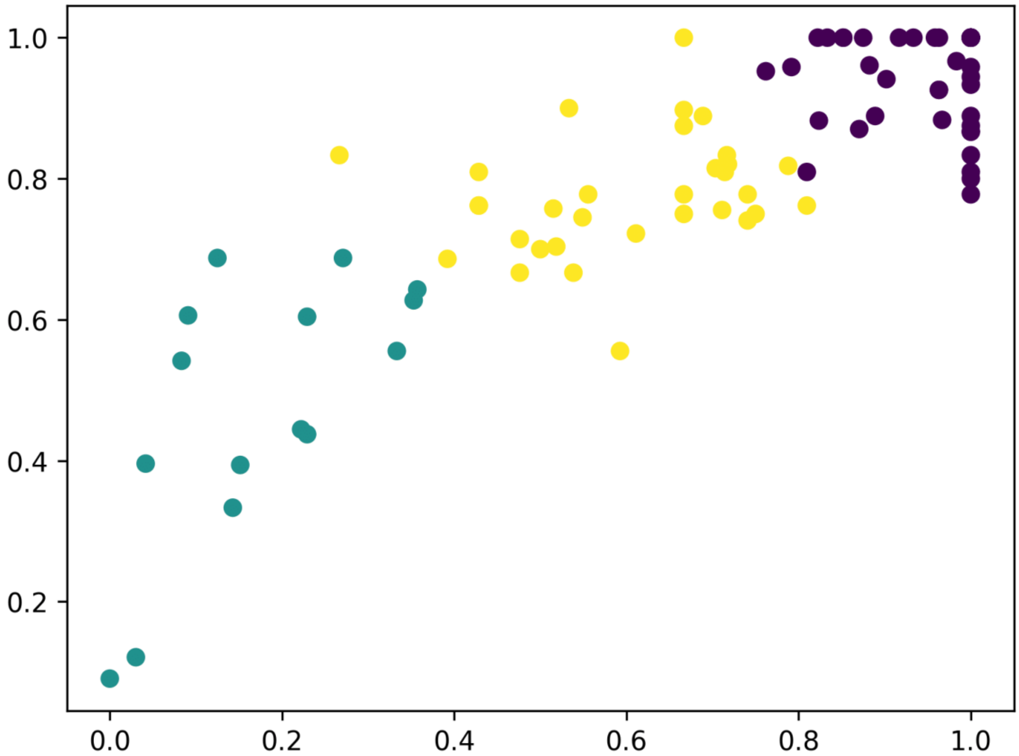
Clustering GPT Markscheme vs GPT No Markscheme

In [11]:
```
#create data structure to cluster
clust_data_1 = list(zip(data['Normal_GPT_M'], data['Normal_GPT_NM']))
```

In [12]:
```
#Array for variation within each cluster
inertias = []

#We will calculate the variation for 10 clusters
for i in range(1,11):
    kmeans = KMeans(n_clusters=i, n_init = 10)
    kmeans.fit(clust_data_1)
    inertias.append(kmeans.inertia_)

#plotting elbow curve
plt.plot(range(1,11), inertias, marker='o')
plt.title('Elbow Curve')
plt.xlabel('Number of clusters')
plt.ylabel('Inertia')
plt.show()
```



In [13]:
```
#Creating the classified data. Play around with n_clusters if you would like to see how data is c
kmeans = KMeans(n_clusters=3, n_init = 10)
kmeans.fit(clust_data_1)

#Plotting classified data
plt.scatter(data['Normal_GPT_M'], data['Normal_GPT_NM'], c=kmeans.labels_)
plt.show()
ques = np.arange(1,91,1)
print("1 = green, 2 = yellow, 0 = purple")
sort_1 = np.column_stack((kmeans.labels_,ques))
sort_1 = sort_1[sort_1[:,0].argsort()]
```

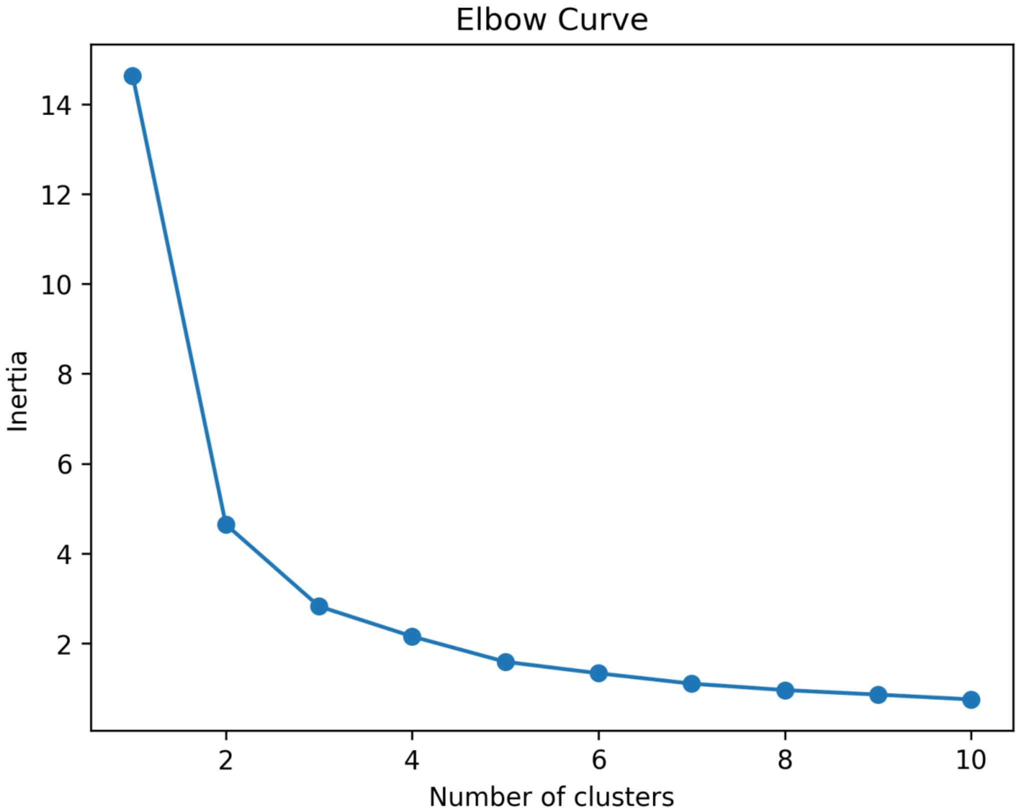



```
1 = green, 2 = yellow, 0 = purple
```

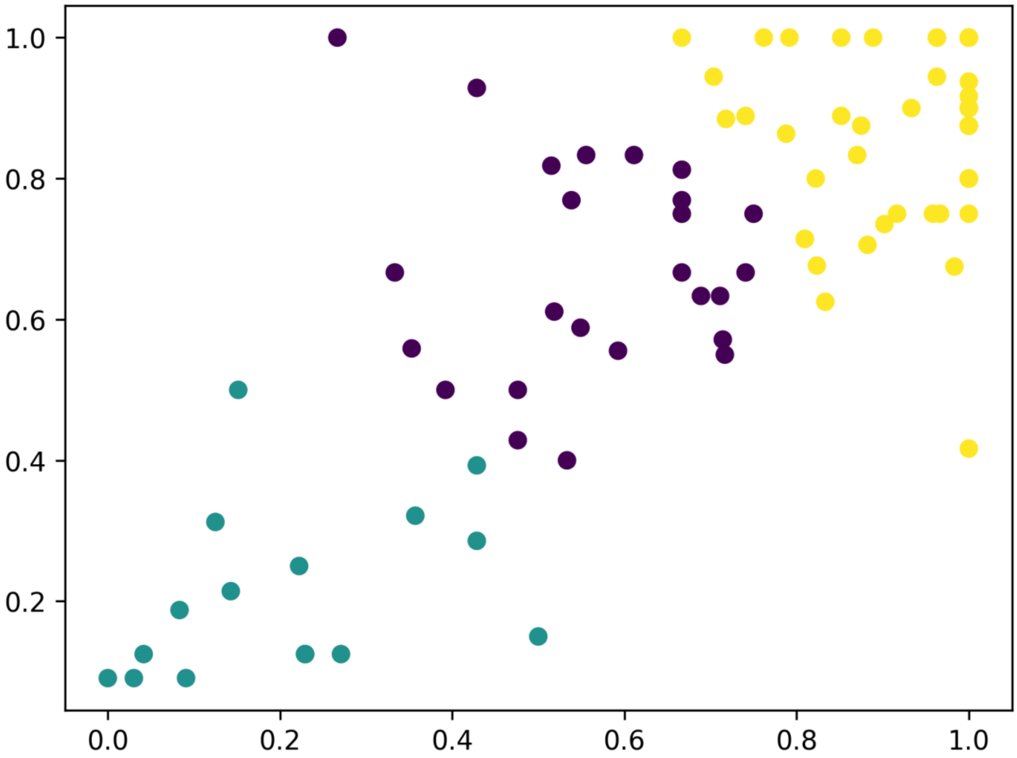
Clustering GPT Markscheme vs Human Graded

In [14]:
```
#create data structure to cluster
clust_data_2 = list(zip(data['Normal_GPT_M'], data['Normal_HG']))
```

In [15]:
```
#Array for variation within each cluster
inertias = []

#We will calculate the variation for 10 clusters
for i in range(1,11):
    kmeans = KMeans(n_clusters=i, n_init = 10)
    kmeans.fit(clust_data_2)
    inertias.append(kmeans.inertia_)

#plotting elbow curve
plt.plot(range(1,11), inertias, marker='o')
plt.title('Elbow Curve')
plt.xlabel('Number of clusters')
plt.ylabel('Inertia')
plt.show()
```



In [16]:
```
#Creating the classified data. Play around with n_clusters if you would like to see how data is c
kmeans = KMeans(n_clusters=3, n_init = 10)
kmeans.fit(clust_data_2)

#Plotting classified data
plt.scatter(data['Normal_GPT_M'], data['Normal_HG'], c=kmeans.labels_)
plt.show()
print("1 = green, 2 = yellow, 0 = purple")
sort_2 = np.column_stack((kmeans.labels_,ques))
sort_2 = sort_2[sort_2[:,0].argsort()]
```

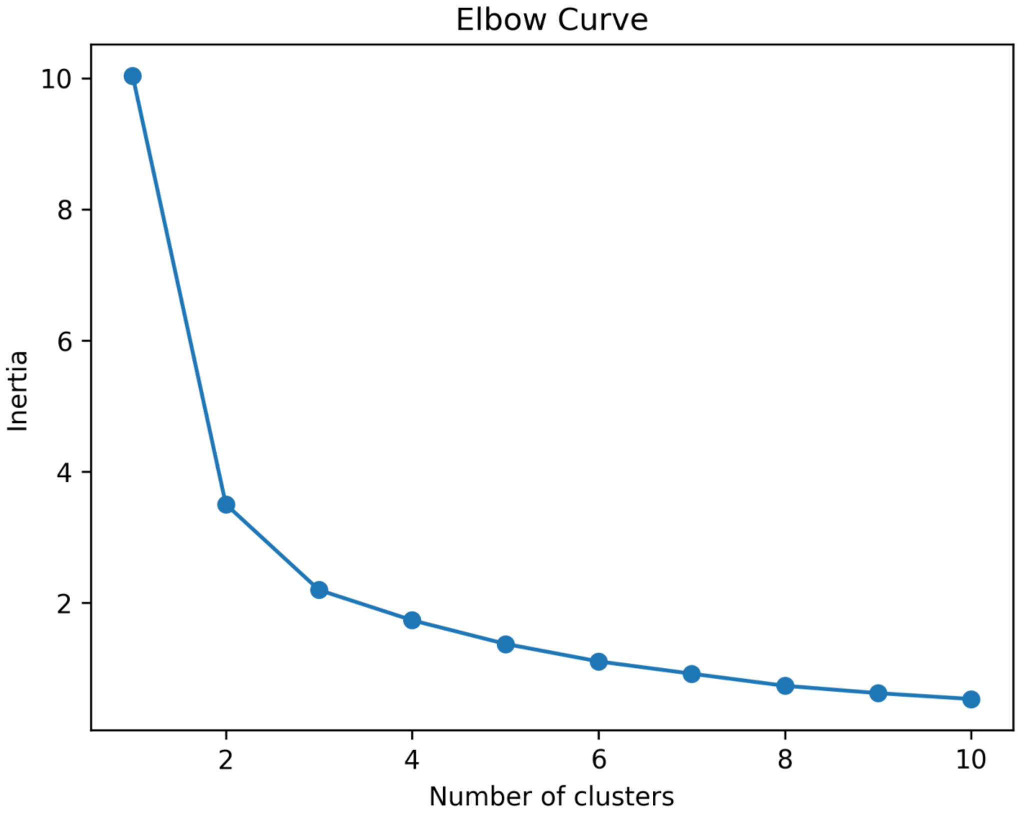



```
1 = green, 2 = yellow, 0 = purple
```

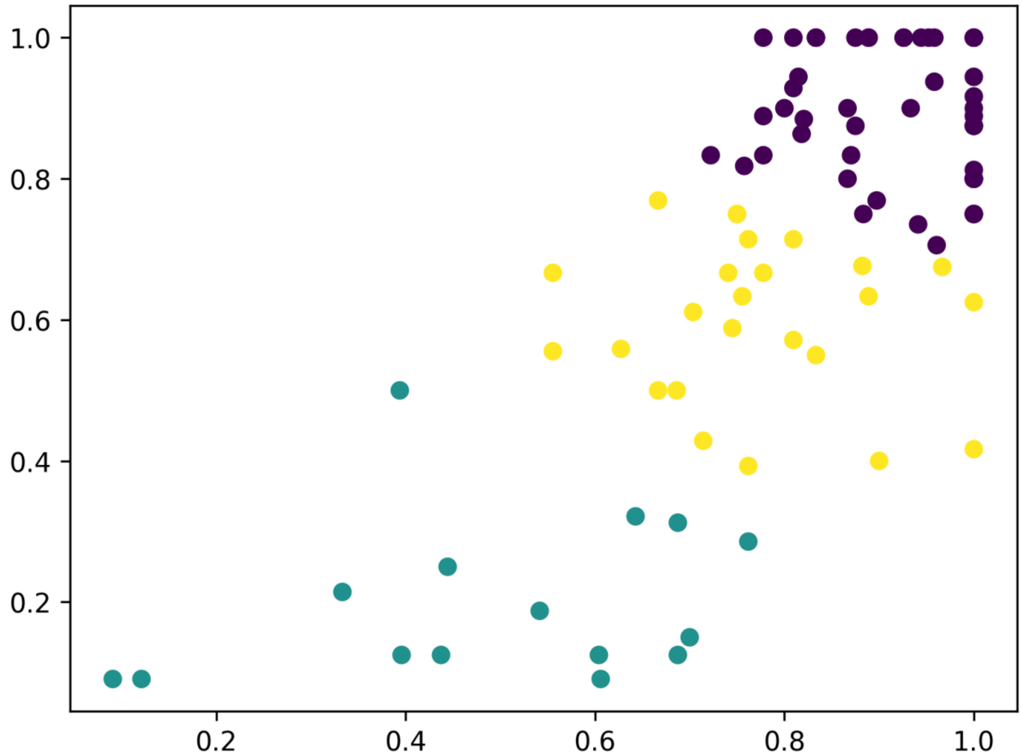
Clustering GPT No Markscheme vs Human Graded

In [17]:
```
#create data structure to cluster
clust_data_3 = list(zip(data['Normal_GPT_NM'], data['Normal_HG']))
```

In [18]:
```
#Array for variation within each cluster
inertias = []

#We will calculate the variation for 10 clusters
for i in range(1,11):
    kmeans = KMeans(n_clusters=i, n_init = 10)
    kmeans.fit(clust_data_3)
    inertias.append(kmeans.inertia_)

#plotting elbow curve
plt.plot(range(1,11), inertias, marker='o')
plt.title('Elbow Curve')
plt.xlabel('Number of clusters')
plt.ylabel('Inertia')
plt.show()
```



In [19]:
```
#Creating the classified data. Play around with n_clusters if you would like to see how data is c
kmeans = KMeans(n_clusters=3, n_init = 10)
kmeans.fit(clust_data_3)

#Plotting classified data
plt.scatter(data['Normal_GPT_NM'], data['Normal_HG'], c=kmeans.labels_)
plt.show()
print("1 = green, 2 = yellow, 0 = purple")
sort_3 = np.column_stack((kmeans.labels_,ques))
sort_3 = sort_3[sort_3[:,0].argsort()]
```




```
1 = green, 2 = yellow, 0 = purple
```

Implementing t-SNE

In [20]:
```
red_data = data[["Normal_GPT_NM", "Normal_GPT_M", "Normal_HG"]] #creating data with only normalis
```

In [21]:
```
#running t-SNE
tsne = TSNE(learning_rate = 500, n_components = 2)
x_tsne = tsne.fit_transform(red_data)
y = np.concatenate(np.stack((np.repeat(1,30), np.repeat(2,30), np.repeat(3,30)), axis = 0)) #crea
pd.DataFrame(x_tsne)
```

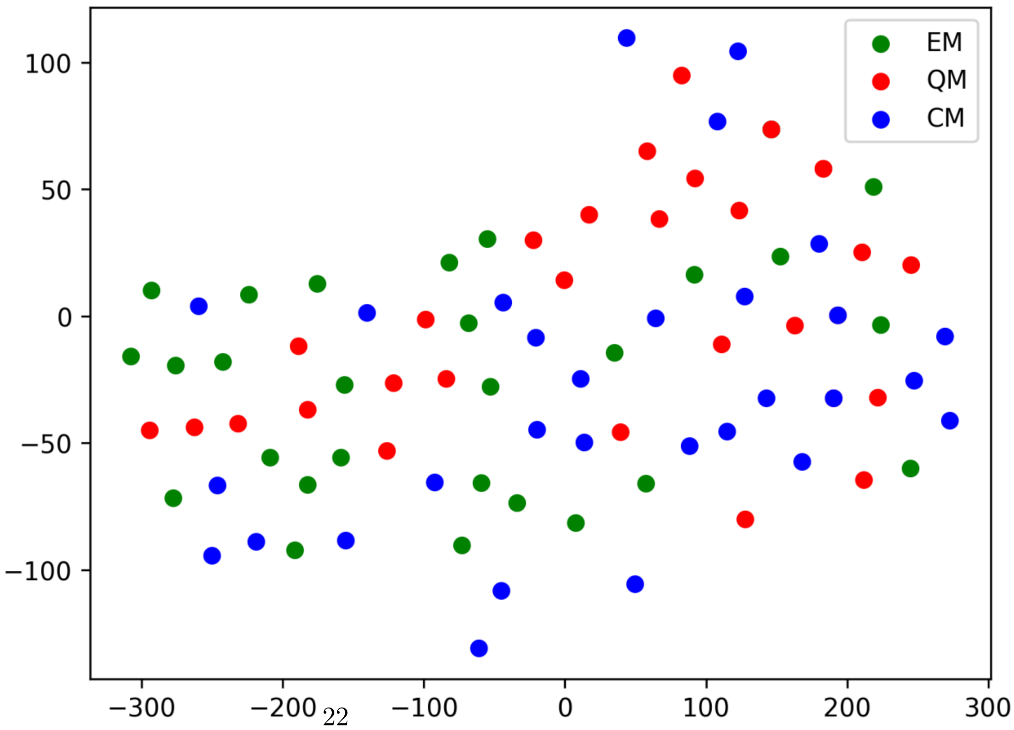
Out[21]:

	0	1
0	7.336886	-81.407669
1	146.095932	73.788910
2	-182.719269	-66.345970
3	35.052395	-14.296751
4	-293.113831	10.148754
...	...	...
85	-250.226654	-94.322876
86	-92.474510	-65.320023
87	-155.477402	-88.207253
88	88.283379	-51.022343
89	127.173264	7.746514

90 rows × 2 columns

In [22]:
```
plt.figure()
plt.scatter(x_tsne[np.where(y==1),0], x_tsne[np.where(y==1), 1], color='g', label = "EM")
plt.scatter(x_tsne[np.where(y==2),0], x_tsne[np.where(y==2), 1], color='r', label = "QM")
plt.scatter(x_tsne[np.where(y==3),0], x_tsne[np.where(y==3), 1], color='b', label = "CM")
plt.legend()
```

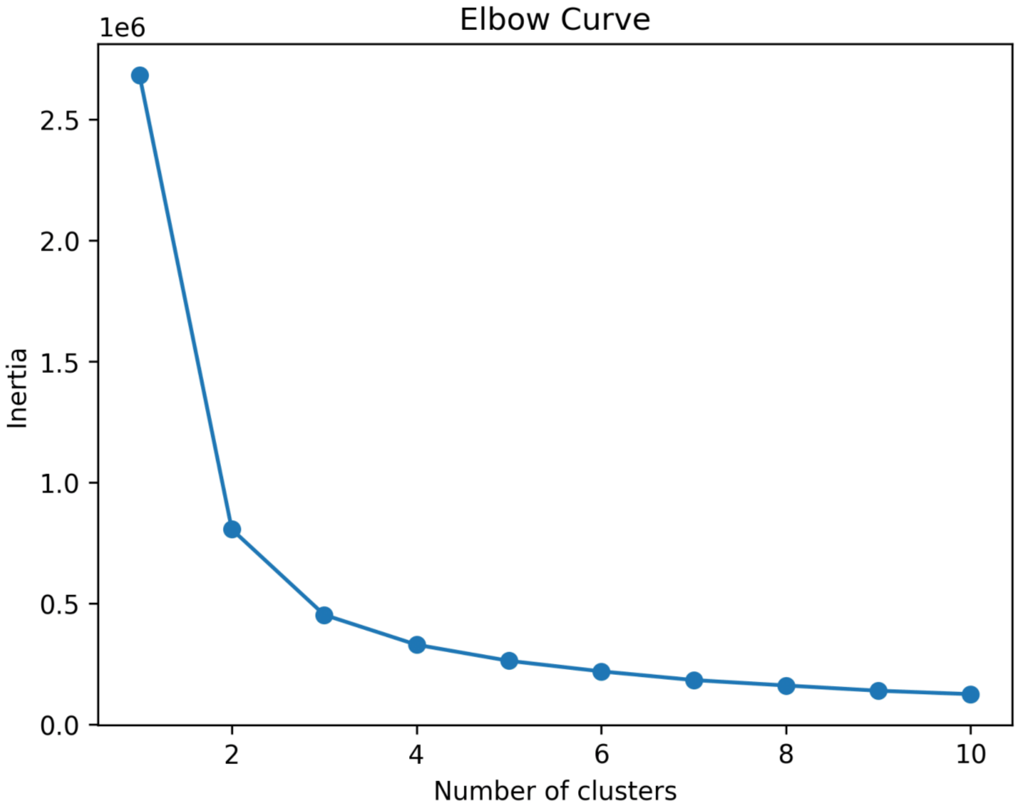
Out[22]: <matplotlib.legend.Legend at 0x15fa4dbd0>



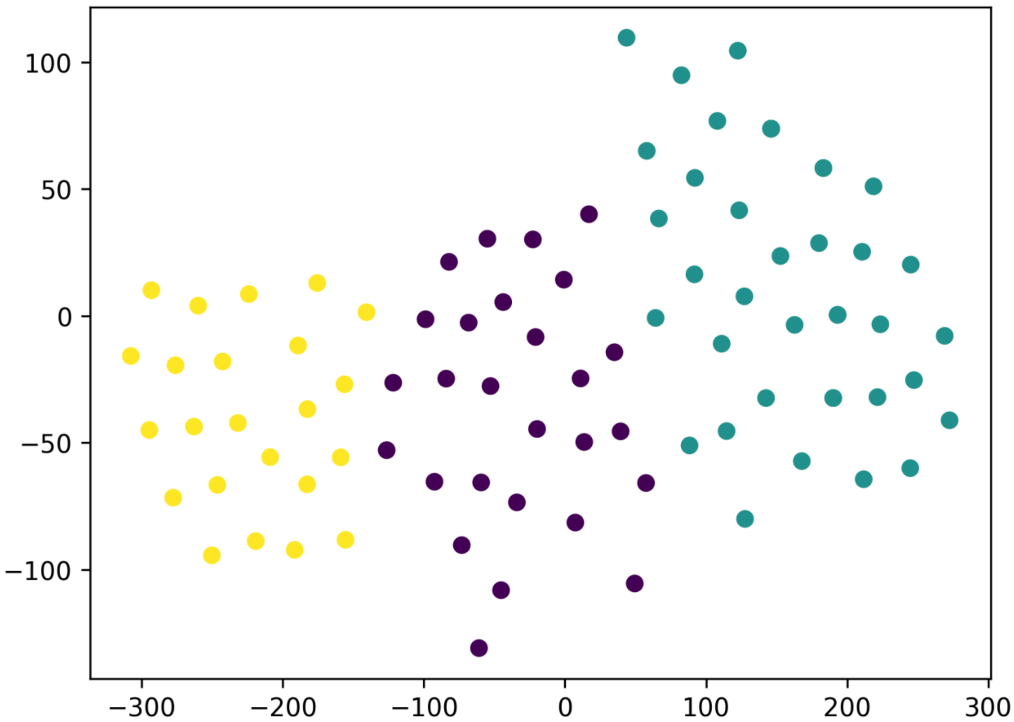
22

In [23]:
```
clust_data_4 = tuple(map(tuple, x_tsne))
```

In [24]:
```
#Array for variation within each cluster
inertias = []

#We will calculate the variation for 10 clusters
for i in range(1,11):
    kmeans = KMeans(n_clusters=i, n_init = 10)
    kmeans.fit(clust_data_4)
    inertias.append(kmeans.inertia_)

#plotting elbow curve
plt.plot(range(1,11), inertias, marker='o')
plt.title('Elbow Curve')
plt.xlabel('Number of clusters')
plt.ylabel('Inertia')
plt.show()
```



In [25]:
```
#Creating the classified data. Play around with n_clusters if you would like to see how data is c
kmeans = KMeans(n_clusters=3, n_init = 10)
kmeans.fit(clust_data_4)

#Plotting classified data
plt.scatter(x_tsne[np.where(y>0),0], x_tsne[np.where(y>0), 1], c=kmeans.labels_)
plt.show()
print("1 = green, 2 = yellow, 0 = purple")
sort_4 = np.column_stack((kmeans.labels_,ques))
sort_4 = sort_4[sort_4[:,0].argsort()]
```




```
1 = green, 2 = yellow, 0 = purple
```

In [ ]:

C Data Analysis Sheets

C.1 Data Preparation

Human Grades

		(Ryan)	Marks Awarded		(Faraaz)	Marks Awarded		(Average)	Marks Awarded	
Question	Total Marks	Solution 1:	Solution 2:	Solution 3:	Solution 1:	Solution 2:	Solution 3:	Solution 1:	Solution 2:	Solution 3:
1	8	6	8	5	7	8	7	6.5	8	6
2	5	4	5	5	4	4	4	4	4.5	4.5
3	14	6	4	4	5	4	5	5.5	4	4.5
4	11	10	5	9	9	6	9	9.5	5.5	9
5	11	1	1	1	1	1	1	1	1	1
6	5	4	5	4	4	4	4	4	4.5	4
7	6	4	4	4	6	6	4	5	5	4
8	6	6	2	4	6	1	4	6	1.5	4
9	15	11	8	8	13	11	11	12	9.5	9.5
10	7	1	4	3	2	3	3	1.5	3.5	3
	88									

GPT Grades (No Markscheme)

		Trial 1	Marks Awarded		Trial 2	Marks Awarded		Trial 3	Marks Awarded		(Average)	Marks Awarded	
Question	Total Marks	Solution 1:	Solution 2:	Solution 3:	Solution 1:	Solution 2:	Solution 3:	Solution 1:	Solution 2:	Solution 3:	Solution 1:	Solution 2:	Solution 3:
1	8	8	7	6	8	7	6	8	7	6	8	7	6
2	5	5	4	4	5	4	5	5	4	5	5	4	4.666666667
3	14	13	12	11	9	10	8	10	10	8	10.66666667	10.66666667	9
4	11	9	4	8	10	6	9	8	3	8	9	4.333333333	8.333333333
5	11	2	7	3	1	4	1	0	9	0	1	6.666666667	1.333333333
6	5	5	5	5	5	5	5	5	5	5	5	5	5
7	6	5	4	5	5	5	5	4	4	4	4.666666667	4.333333333	4.666666667
8	6	5	2	4	6	3	4	6	3	2	5.666666667	2.666666667	3.333333333
9	15	15	13	10	15	12	12	15	15	12	15	13.33333333	11.33333333
10	7	3	5	7	1	4	4	3	5	4	2.333333333	4.666666667	5

GPT Grades (Markscheme)

		Trial 1	Marks Awarded		Trial 2	Marks Awarded		Trial 3	Marks Awarded		(Average)	Marks Awarded	
Question	Total Marks	Solution 1:	Solution 2:	Solution 3:	Solution 1:	Solution 2:	Solution 3:	Solution 1:	Solution 2:	Solution 3:	Solution 1:	Solution 2:	Solution 3:
1	8	5	5	5	6	6	6	5	5	5	5.333333333	5.333333333	5.333333333
2	5	5	5	5	5	5	5	5	5	5	5	5	5
3	14	4	4	4	5	5	5	9	9	6	6	6	5
4	11	9	1	7	8	3	7	9	1	3	8.666666667	1.666666667	5.666666667
5	11	0	0	0	0	1	1	0	2	0	0	1	0.3333333333
6	5	5	5	5	5	5	5	5	4	5	5	4.666666667	5
7	6	2	3	3	4	4	4	4	4	5	3.333333333	3.666666667	4
8	6	6	1	3	6	1	1	6	2	2	6	1.333333333	2
9	15	10	10	10	12	11	9	15	10	13	12.33333333	10.33333333	10.66666667
10	7	1	1	1	2	4	6	0	5	3	1	3.333333333	3.333333333

EM Final Data

Question	Total_Marks	Human_Grade	GPT_Grade_NM	GPT_Grade_M	Normal_HG	Normal_GPT_NM	Normal_GPT_M	std_dev_HG	std_dev_GPT_NM	std_dev_GPT_M	Norm_std_dev_HG	Norm_std_dev_GPT_NM	Norm_std_dev_GPT_M
1	8	6.5	8	5.333333333	0.8125	1	0.6666666667	0.7071067812	0	0.5773502692	0.08838834765	0	0.07216878365
2	5	4	5	5	0.8	1	1	0	0	0	0	0	0
3	14	5.5	10.66666667	6	0.3928571429	0.7619047619	0.4285714286	0.7071067812	2.081665999	2.645751311	0.06629126074	0.1951561874	0.1889822365
4	11	9.5	9	8.666666667	0.8636363636	0.8181818182	0.7878787879	0.7071067812	1	0.5773502692	0.07856742013	0.1111111111	0.05248638811
5	11	1	1	0	0.09090909091	0.09090909091	0	0	1	0	0	1	0
6	5	4	5	5	0.8	1	1	0	0	0	0	0	0
7	6	5	4.666666667	3.333333333	0.8333333333	0.7777777778	0.5555555556	1.414213562	0.5773502692	1.154700538	0.3030457634	0.1237179148	0.1924500897
8	6	6	5.666666667	6	1	0.9444444444	1	0	0.5773502692	0	0	0.1018853416	0
9	15	12	15	12.33333333	0.8	1	0.8222222222	1.414213562	0	2.516611478	0.09428090416	0	0.1677740986
10	7	1.5	2.333333333	1	0.2142857143	0.3333333333	0.1428571429	0.7071067812	1.154700538	1	0.3030457634	0.4948716593	0.1428571429
1	8	8	7	5.333333333	1	0.875	0.6666666667	0	0	0.5773502692	0	0	0.07216878365
2	5	4.5	4	5	0.9	0.8	1	0.7071067812	0	0	0.1767766953	0	0
3	14	4	10.66666667	6	0.2857142857	0.7619047619	0.4285714286	0	1.154700538	2.645751311	0	0.1082531755	0.1889822365
4	11	5.5	4.333333333	1.666666667	0.5	0.3939393939	0.1515151515	0.7071067812	1.527525232	1.154700538	0.163178488	0.3525058227	0.1049727762
5	11	1	6.666666667	1	0.09090909091	0.6060606061	0.09090909091	0	2.516611478	1	0	0.3774917218	0.09090909091
6	5	4.5	5	4.666666667	0.9	1	0.9333333333	0.7071067812	0	0.5773502692	0.1414213562	0	0.1154700538
7	6	5	4.333333333	3.666666667	0.8333333333	0.7222222222	0.6111111111	1.414213562	0.5773502692	0.5773502692	0.3263569759	0.1332346775	0.09622504486
8	6	1.5	2.666666667	1.333333333	0.25	0.4444444444	0.2222222222	0.7071067812	0.5773502692	0.5773502692	0.2651650429	0.2165063509	0.09622504486
9	15	9.5	13.33333333	10.33333333	0.6333333333	0.8888888889	0.6888888889	2.121320344	1.527525232	0.5773502692	0.1590990258	0.1145643924	0.03849001795
10	7	3.5	4.666666667	3.333333333	0.5	0.6666666667	0.4761904762	0.7071067812	0.5773502692	2.081665999	0.1515228817	0.1237179148	0.2973808571
1	8	6	6	5.333333333	0.75	0.75	0.6666666667	1.414213562	0	0.5773502692	0.2357022604	0	0.07216878365
2	5	4.5	4.666666667	5	0.9	0.9333333333	1	0.7071067812	0.5773502692	0	0.1515228817	0.1237179148	0
3	14	4.5	9	5	0.3214285714	0.6428571429	0.3571428571	0.7071067812	1.732050808	1	0.07856742013	0.1924500897	0.07142857143
4	11	9	8.333333333	5.666666667	0.8181818182	0.7575757576	0.5151515152	0	0.5773502692	2.309401077	0	0.0692820323	0.2099455524
5	11	1	1.333333333	0.3333333333	0.09090909091	0.1212121212	0.0303030303	0	1.527525232	0.5773502692	0	1.145643924	0.05248638811
6	5	4	5	5	0.8	1	1	0	0	0	0	0	0
7	6	4	4.666666667	4	0.6666666667	0.7777777778	0.6666666667	0	0.5773502692	1	0	0.1237179148	0.1666666667
8	6	4	3.333333333	2	0.6666666667	0.5555555556	0.3333333333	0	1.154700538	1	0	0.3464101615	0.1666666667
9	15	9.5	11.33333333	10.66666667	0.6333333333	0.7555555556	0.7111111111	2.121320344	1.154700538	2.081665999	0.1871753244	0.1018853416	0.1387777333
10	7	3	5	3.333333333	0.4285714286	0.7142857143	0.4761904762	0	1.732050808	2.516611478	0	0.3464101615	0.3595159255

Human Grades

		(Ryan)	Marks Awarded		(Faraaz)	Marks Awarded		(Average)	Marks Awarded	
Question	Total Marks	Solution 1:	Solution 2:	Solution 3:	Solution 1:	Solution 2:	Solution 3:	Solution 1:	Solution 2:	Solution 3:
1	5	4	4	4	4	5	4	4	4.5	4
2	4	4	3	3	4	4	4	4	3.5	3.5
3	9	9	6	9	9	5	9	9	5.5	9
4	20	15	13	16	15	9	11	15	11	13.5
5	7	6	5	6	4	3	4	5	4	5
6	17	11	9	8	9	10	9	10	9.5	8.5
7	8	5	7	6	7	8	8	6	7.5	7
8	9	8	8	9	8	9	8	8	8.5	8.5
9	17	12	13	12	11	12	12	11.5	12.5	12
10	16	7	3	3	3	3	1	5	3	2
total marks	112									

GPT Grades (No Markscheme)

		Trial 1	Marks Awarded		Trial 2	Marks Awarded		Trial 3	Marks Awarded		(Average)	Marks Awarded	
Question	Total Marks	Solution 1:	Solution 2:	Solution 3:	Solution 1:	Solution 2:	Solution 3:	Solution 1:	Solution 2:	Solution 3:	Solution 1:	Solution 2:	Solution 3:
1	5	5	4	4	5	5	5	5	4	4	5	4.333333333	4.333333333
2	4	4	4	4	4	4	4	4	4	4	4	4	4
3	9	7	5	9	9	7	7	9	7	8	8.333333333	6.333333333	8
4	20	17	17	20	18	16	19	18	17	19	17.66666667	16.66666667	19.33333333
5	7	6	6	6	5	5	4	6	6	6	5.666666667	5.666666667	5.333333333
6	17	14	11	10	11	12	9	13	9	16	12.66666667	10.66666667	11.66666667
7	8	8	7	7	8	8	6	8	8	8	8	7.666666667	7
8	9	9	8	9	9	7	9	9	7	9	9	7.333333333	9
9	17	16	16	17	15	15	17	14	17	15	15	16	16.33333333
10	16	10	8	7	11	9	6	12	9	6	11	8.666666667	6.333333333

GPT Grades (Markscheme)

		Trial 1	Marks Awarded		Trial 2	Marks Awarded		Trial 3	Marks Awarded		(Average)	Marks Awarded	
Question	Total Marks	Solution 1:	Solution 2:	Solution 3:	Solution 1:	Solution 2:	Solution 3:	Solution 1:	Solution 2:	Solution 3:	Solution 1:	Solution 2:	Solution 3:
1	5	5	5	5	5	5	5	5	5	5	5	5	5
2	4	4	4	4	4	4	4	4	4	4	4	4	4
3	9	8	6	9	9	3	9	9	5	9	8.666666667	4.666666667	9
4	20	20	14	20	18	15	19	20	14	20	19.33333333	14.33333333	19.66666667
5	7	7	6	7	5	5	5	5	4	5	5.666666667	5	5.666666667
6	17	11	5	6	8	6	7	9	7	7	9.333333333	6	6.666666667
7	8	8	8	8	6	8	8	8	8	8	7.333333333	8	8
8	9	8	6	9	7	7	9	8	6	8	7.666666667	6.333333333	8.666666667
9	17	12	16	14	17	15	17	13	15	14	14	15.33333333	15
10	16	3	1	0	2	2	2	1	1	0	2	1.333333333	0.6666666667

QM Final Data

Question	Total_Marks	Human_Grade	GPT_Grade_NM	GPT_Grade_M	Normal_HG	Normal_GPT_NM	Normal_GPT_M	std_dev_HG	std_dev_GPT_NM	std_dev_GPT_M	Norm_std_dev_HG	Norm_std_dev_GPT_NM	Norm_std_dev_GPT_M
1	5	4	5	5	0.8	1	1	0	0	0	0	0	0
2	4	4	4	4	1	1	1	0	0	0	0	0	0
3	9	9	8.333333333	8.666666667	1	0.9259259259	0.962962963	0	1.154700538	0.5773502692	0	0.1283000598	0.06415002991
4	20	15	17.66666667	19.33333333	0.75	0.8833333333	0.9666666667	0	0.5773502692	1.154700538	0	0.02886751346	0.05773502692
5	7	5	5.666666667	5.666666667	0.7142857143	0.8095238095	0.8095238095	1.414213562	0.5773502692	1.154700538	0.2020305089	0.08247860988	0.1649572198
6	17	10	12.66666667	9.333333333	0.5882352941	0.7450980392	0.5490196078	1.414213562	1.527525232	1.527525232	0.08318903308	0.08985442539	0.08985442539
7	8	6	8	7.333333333	0.75	1	0.9166666667	1.414213562	0	1.154700538	0.1767766953	0	0.1443375673
8	9	8	9	7.666666667	0.8888888889	1	0.8518518519	0	0	0.5773502692	0	0	0.06415002991
9	17	11.5	15	14	0.6764705882	0.8823529412	0.8235294118	0.7071067812	1	2.645751311	0.04159451654	0.05882352941	0.1556324301
10	16	5	11	2	0.3125	0.6875	0.125	2.828427125	1	1	0.1767766953	0.0625	0.0625
1	5	4.5	4.333333333	5	0.9	0.8666666667	1	0.7071067812	0.5773502692	0	0.1414213562	0.1154700538	0
2	4	3.5	4	4	0.875	1	1	0.7071067812	0	0	0.1767766953	0	0
3	9	5.5	6.333333333	4.666666667	0.6111111111	0.7037037037	0.5185185185	0.7071067812	1.154700538	1.527525232	0.07856742013	0.1283000598	0.1697250257
4	20	11	16.66666667	14.33333333	0.55	0.8333333333	0.7166666667	2.828427125	0.5773502692	0.5773502692	0.1414213562	0.02886751346	0.02886751346
5	7	4	5.666666667	5	0.5714285714	0.8095238095	0.7142857143	1.414213562	0.5773502692	1	0.2020305089	0.08247860988	0.1428571429
6	17	9.5	10.66666667	6	0.5588235294	0.6274509804	0.3529411765	0.7071067812	1.527525232	1	0.04159451654	0.08985442539	0.05882352941
7	8	7.5	7.666666667	8	0.9375	0.9583333333	1	0.7071067812	0.5773502692	0	0.08838834765	0.07216878365	0
8	9	8.5	7.333333333	6.333333333	0.9444444444	0.8148148148	0.7037037037	0.7071067812	0.5773502692	0.5773502692	0.07856742013	0.06415002991	0.06415002991
9	17	12.5	16	15.33333333	0.7352941176	0.9411764706	0.9019607843	0.7071067812	1	0.5773502692	0.04159451654	0.05882352941	0.03396178054
10	16	3	8.666666667	1.333333333	0.1875	0.5416666667	0.08333333333	0	0.5773502692	0.5773502692	0	0.03608439182	0.03608439182
1	5	4	4.333333333	5	0.8	0.8666666667	1	0	0.5773502692	0	0	0.1154700538	0
2	4	3.5	4	4	0.875	1	1	0.7071067812	0	0	0.1767766953	0	0
3	9	9	8	9	1	0.8888888889	1	0	1	0	0	0.1111111111	0
4	20	13.5	19.33333333	19.66666667	0.675	0.9666666667	0.9833333333	3.535533906	0.5773502692	0.5773502692	0.1767766953	0.02886751346	0.02886751346
5	7	5	5.333333333	5.666666667	0.7142857143	0.7619047619	0.8095238095	1.414213562	1.154700538	1.154700538	0.2020305089	0.1649572198	0.1649572198
6	17	8.5	11.66666667	6.666666667	0.5	0.6862745098	0.3921568627	0.7071067812	3.785938897	0.5773502692	0.04159451654	0.2227022881	0.03396178054
7	8	7	7	8	0.875	0.875	1	1.414213562	1	0	0.1767766953	0.125	0
8	9	8.5	9	8.666666667	0.9444444444	1	0.962962963	0.7071067812	0	0.5773502692	0.07856742013	0	0.06415002991
9	17	12	16.33333333	15	0.7058823529	0.9607843137	0.8823529412	0	1.154700538	1.732050808	0	0.06792356108	0.1018853416
10	16	2	6.333333333	0.6666666667	0.125	0.3958333333	0.04166666667	1.414213562	0.5773502692	1.154700538	0.08838834765	0.03608439182	0.07216878365

Human Grades

		(Ryan)	Marks Awarded		(Faraaz)	Marks Awarded		(Average)	Marks Awarded	
Question	Total Marks	Solution 1:	Solution 2:	Solution 3:	Solution 1:	Solution 2:	Solution 3:	Solution 1:	Solution 2:	Solution 3:
1	4	2	3	3	3	3	3	2.5	3	3
2	6	6	6	6	6	6	6	6	6	6
3	6	5	3	6	6	2	6	5.5	2.5	6
4	18	12	15	11	12	15	9	12	15	10
5	9	7	9	9	9	9	9	8	9	9
6	16	2	2	2	2	2	2	2	2	2
7	13	10	12	10	10	11	10	10	11.5	10
8	10	10	1	4	10	2	4	10	1.5	4
9	7	6	7	7	7	7	7	6.5	7	7
10	8	6	8	7	6	8	7	6	8	7
total marks	97									

GPT Grades (No Markscheme)

		Trial 1	Marks Awarded		Trial 2	Marks Awarded		Trial 3	Marks Awarded		(Average)	Marks Awarded	
Question	Total Marks	Solution 1:	Solution 2:	Solution 3:	Solution 1:	Solution 2:	Solution 3:	Solution 1:	Solution 2:	Solution 3:	Solution 1:	Solution 2:	Solution 3:
1	4	4	3	4	4	3	4	4	3	4	4	3	4
2	6	5	6	4	5	4	6	5	6	4	5	5.333333333	4.666666667
3	6	6	6	6	6	6	6	6	6	6	6	6	6
4	18	13	16	9	13	16	10	14	15	11	13.33333333	15.66666667	10
5	9	7	9	9	7	9	9	7	7	9	7	8.333333333	9
6	16	10	7	13	10	7	10	9	7	10	9.666666667	7	11
7	13	12	9	8	13	11	8	10	12	10	11.66666667	10.66666667	8.666666667
8	10	8	6	10	9	8	10	8	7	7	8.333333333	7	9
9	7	6	7	6	7	7	7	4	3	7	5.666666667	5.666666667	6.666666667
10	8	8	8	8	8	8	8	8	7	8	8	7.666666667	8

GPT Grades (Markscheme)

		Trial 1	Marks Awarded		Trial 2	Marks Awarded		Trial 3	Marks Awarded		(Average)	Marks Awarded	
Question	Total Marks	Solution 1:	Solution 2:	Solution 3:	Solution 1:	Solution 2:	Solution 3:	Solution 1:	Solution 2:	Solution 3:	Solution 1:	Solution 2:	Solution 3:
1	4	4	4	4	3	3	4	3	2	4	3.333333333	3	4
2	6	6	5	6	6	5	6	6	6	6	6	5.333333333	6
3	6	6	6	6	6	6	6	6	6	6	6	6	6
4	18	11	14	10	14	16	11	15	17	11	13.33333333	15.66666667	10.66666667
5	9	7	9	9	6	9	7	7	8	7	6.666666667	8.666666667	7.666666667
6	16	2	3	2	2	2	5	7	6	6	3.666666667	3.666666667	4.333333333
7	13	6	8	7	10	10	5	10	10	9	8.666666667	9.333333333	7
8	10	4	6	7	1	3	3	3	6	6	2.666666667	5	5.333333333
9	7	3	7	3	4	7	6	2	7	7	3	7	5.333333333
10	8	8	6	6	8	6	7	7	7	8	7.666666667	6.333333333	7

CM Final Data

Question	Total_Marks	Human_Grade	GPT_Grade_NM	GPT_Grade_M	Normal_HG	Normal_GPT_NM	Normal_GPT_M	std_dev_HG	std_dev_GPT_NM	std_dev_GPT_M	Norm_std_dev_HG	Norm_std_dev_GPT_NM	Norm_std_dev_GPT_M
1	4	2.5	4	3.333333333	0.625	1	0.8333333333	0.7071067812	0	0.5773502692	0.1767766953	0	0.1443375673
2	6	6	5	6	1	0.8333333333	1	0	0	0	0	0	0
3	6	5.5	6	6	0.9166666667	1	1	0.7071067812	0	0	0.1178511302	0	0
4	18	12	13.33333333	13.33333333	0.6666666667	0.7407407407	0.7407407407	0	0.5773502692	2.081665999	0	0.03207501495	0.1156481111
5	9	8	7	6.666666667	0.8888888889	0.7777777778	0.7407407407	1.414213562	0	0.5773502692	0.1571348403	0	0.06415002991
6	16	2	9.666666667	3.666666667	0.125	0.6041666667	0.2291666667	0	0.5773502692	2.886751346	0	0.03608439182	0.1804219591
7	13	10	11.66666667	8.666666667	0.7692307692	0.8974358974	0.6666666667	0	1.527525232	2.309401077	0	0.1175019409	0.1776462367
8	10	10	8.333333333	2.666666667	1	0.8333333333	0.2666666667	0	0.5773502692	1.527525232	0	0.05773502692	0.1527525232
9	7	6.5	5.666666667	3	0.9285714286	0.8095238095	0.4285714286	0.7071067812	1.527525232	1	0.1010152545	0.2182178902	0.1428571429
10	8	6	8	7.666666667	0.75	1	0.9583333333	0	0	0.5773502692	0	0	0.07216878365
1	4	3	3	3	0.75	0.75	0.75	0	0	1	0	0	0.25
2	6	6	5.333333333	5.333333333	1	0.8888888889	0.8888888889	0	1.154700538	0.5773502692	0	0.1924500897	0.09622504486
3	6	2.5	6	6	0.4166666667	1	1	0.7071067812	0	0	0.1178511302	0	0
4	18	15	15.66666667	15.66666667	0.8333333333	0.8703703704	0.8703703704	0	0.5773502692	1.527525232	0	0.03207501495	0.08486251287
5	9	9	8.333333333	8.666666667	1	0.9259259259	0.962962963	0	1.154700538	0.5773502692	0	0.1283000598	0.06415002991
6	16	2	7	3.666666667	0.125	0.4375	0.2291666667	0	0	2.081665999	0	0	0.130104125
7	13	11.5	10.66666667	9.333333333	0.8846153846	0.8205128205	0.7179487179	0.7071067812	1.527525232	1.154700538	0.05439282932	0.1175019409	0.08882311834
8	10	1.5	7	5	0.15	0.7	0.5	0.7071067812	1	1.732050808	0.07071067812	0.1	0.1732050808
9	7	7	5.666666667	7	1	0.8095238095	1	0	2.309401077	0	0	0.3299144395	0
10	8	8	7.666666667	6.333333333	1	0.9583333333	0.7916666667	0	0.5773502692	0.5773502692	0	0.07216878365	0.07216878365
1	4	3	4	4	0.75	1	1	0	0	0	0	0	0
2	6	6	4.666666667	6	1	0.7777777778	1	0	1.154700538	0	0	0.1924500897	0
3	6	6	6	6	1	1	1	0	0	0	0	0	0
4	18	10	10	10.66666667	0.5555555556	0.5555555556	0.5925925926	1.414213562	1	0.5773502692	0.07856742013	0.05555555556	0.03207501495
5	9	9	9	7.666666667	1	1	0.8518518519	0	0	1.154700538	0	0	0.1283000598
6	16	2	11	4.333333333	0.125	0.6875	0.2708333333	0	1.732050808	2.081665999	0	0.1082531755	0.130104125
7	13	10	8.666666667	7	0.7692307692	0.6666666667	0.5384615385	0	1.154700538	2	0	0.08882311834	0.1538461538
8	10	4	9	5.333333333	0.4	0.9	0.5333333333	0	1.732050808	2.081665999	0	0.1732050808	0.2081665999
9	7	7	6.666666667	5.333333333	1	0.9523809524	0.7619047619	0	0.5773502692	2.081665999	0	0.08247860988	0.2973808571
10	8	7	8	7	0.875	1	0.875	0	0	1	0	0	0.125

C.2 Data Analysis

Cluster	Question	Subtopic	Actual Question	Solution
0	1	EM	1	1
0	11	EM	1	2
0	21	EM	1	3
0	3	EM	3	1
0	13	EM	3	2
0	4	EM	4	1
0	24	EM	4	3
0	7	EM	5	1
0	17	EM	7	2
0	27	EM	7	3
0	19	EM	9	2
0	29	EM	9	3
0	20	EM	10	2
0	30	EM	10	3
0	43	QM	3	2
0	44	QM	4	2
0	45	QM	5	2
0	55	QM	5	3
0	36	QM	6	1
0	56	QM	6	3
0	48	QM	8	2
0	71	CM	1	2
0	64	CM	4	1
0	84	CM	4	3
0	65	CM	5	1
0	67	CM	7	1
0	77	CM	7	2
0	87	CM	7	3
0	68	CM	8	1
0	78	CM	8	2
0	88	CM	8	3
0	69	CM	9	1
1	23	EM	3	3
1	14	EM	4	2
1	5	EM	5	1
1	15	EM	5	2
1	25	EM	5	3
1	18	EM	8	2
1	28	EM	8	3
1	10	EM	10	1
1	46	QM	6	2
1	40	QM	10	1
1	50	QM	10	2
1	60	QM	10	3
1	66	CM	6	1
1	76	CM	6	2
1	86	CM	6	3
2	2	EM	2	1
2	12	EM	2	2
2	22	EM	2	3
2	6	EM	6	1
2	16	EM	6	2
2	26	EM	6	3
2	8	EM	8	1
2	9	EM	9	1
2	31	QM	1	1
2	41	QM	1	2
2	51	QM	1	3
2	32	QM	2	1
2	42	QM	2	2
2	52	QM	2	3
2	33	QM	3	1
2	53	QM	3	3
2	34	QM	4	1
2	54	QM	4	3
2	35	QM	5	1
2	37	QM	7	1
2	47	QM	7	2
2	57	QM	7	3
2	38	QM	8	1
2	58	QM	8	3
2	39	QM	9	1
2	49	QM	9	2
2	59	QM	9	3
2	61	CM	1	1
2	81	CM	1	3
2	62	CM	2	1
2	72	CM	2	2
2	82	CM	2	3
2	63	CM	3	1
2	73	CM	3	2
2	83	CM	3	3
2	74	CM	4	2
2	75	CM	5	2
2	85	CM	5	3
2	79	CM	9	2
2	89	CM	9	3
2	70	CM	10	1
2	80	CM	10	2
2	90	CM	10	3

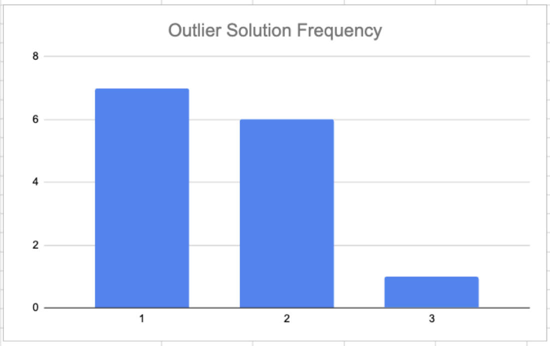
Notes

EM Q1 --> SOL 1 & SOL 2 & SOL 3
EM Q3 --> SOL 1 & SOL 2
EM Q4 --> SOL 1 & SOL 3
EM Q7 --> SOL 2 & SOL 3
EM Q9 --> SOL 2 & SOL 3
EM Q10 --> SOL 2 & SOL3
QM Q5 --> SOL 2 & SOL 3
QM Q6 --> SOL 1 & SOL 3
CM Q4 --> SOL 1 & SOL 3
CM Q7 --> SOL 1 & SOL 2 & SOL 3
CM Q8 --> SOL 1& SOL 2 & SOL 3
Outlier:
EM Q5 --> SOL1
QM Q3 --> SOL 2
QM Q4 --> SOL 2
QM Q8 --> SOL 2
CM Q1 --> SOL 2
CM Q5 --> SOL 1
CM Q9 --> SOL 1

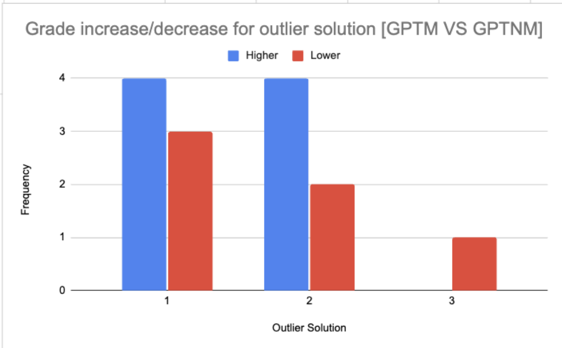
EM Q5 --> SOL 1 & SOL 2 & SOL 3
EM Q8 --> SOL 2 & SOL 3
QM Q10 --> SOL 1 & SOL 2 & SOL3
CM Q6 --> SOL 1 & SOL 2 & SOL 3
Outlier:
EM Q3--> SOL 3
EM Q4 -->SOL 2
EM Q10 --SOL 1
QM Q6 --> SOL2

EM Q2 --> SOL 1 & SOL 2 & SOL 3
EM Q6 --> SOL 1 & SOL 2 & SOL 3
QM Q1 --> SOL 1 & SOL 2 & SOL 3
QM Q2 --> SOL 1 & SOL 2 & SOL 3
QM Q3 --> SOL 1 & SOL 3
QM Q4 --> SOL 1 & SOL 3
QM Q7 --> SOL 1 & 2 & 3
QM Q8 --> SOL 1 & SOL 3
QM Q9 --> SOL 1 & SOL 2 & SOL 3
CM Q1 --> SOL 1 & SOL 3
CM Q2 --> SOL 1 & SOL 2 & SOL 3
CM Q3 --> SOL 1 & SOL 2 & SOL 3
CM Q5 --> SOL 2 & SOL 3
CM Q9 --> SOL 2 & SOL 3
CM Q10 --> SOL 1 & SOL 2 & SOL 3
Outlier
EM Q8 --> SOL 1
EM Q9 --> SOL 1
QM Q5 --> SOL 1
CM Q4 --> SOL 2

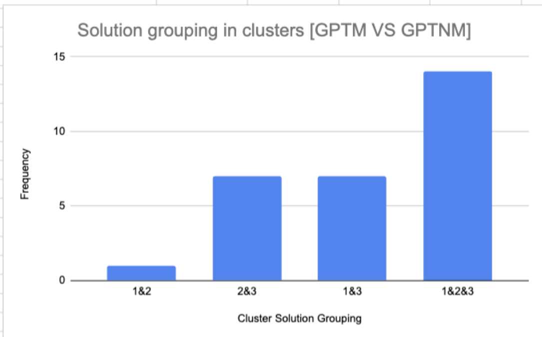
Outlier Solution	Frequency	
		2 = green, 1= yellow, 0 = purple
1	7	
2	6	
3	1	



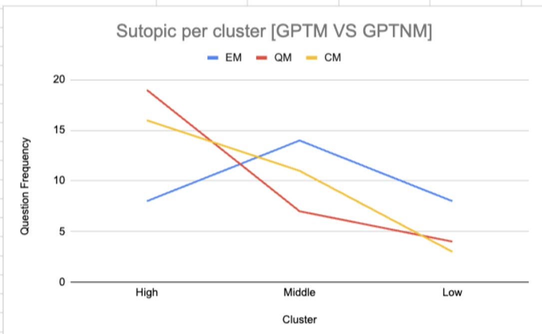
Outlier	Higher	Lower
1	4	3
2	4	2
3	0	1



Cluster Solution Permutation	Frequency
1&2	1
2&3	7
1&3	7
1&2&3	14

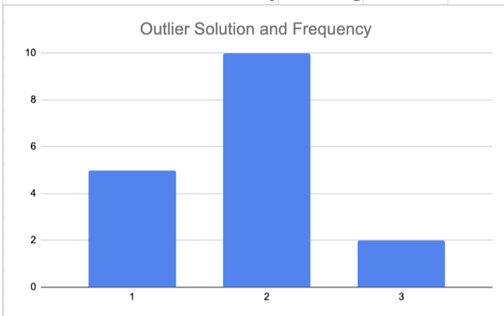


Cluster	EM	QM	CM
High	8	19	16
Middle	14	7	11
Low	8	4	3



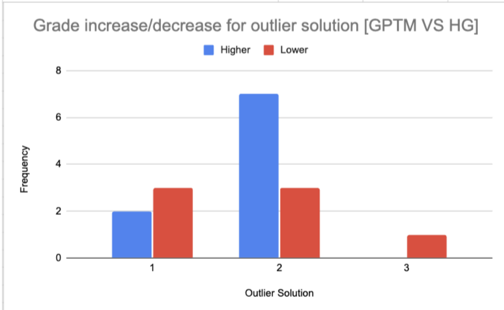
Cluster	Question	Subtopic	Actual Question	Solution	Notes
0	1	EM	1	1	
0	21	EM	1	3	
0	24	EM	4	3	
0	7	EM	7	1	
0	17	EM	7	2	
0	27	EM	7	3	
0	28	EM	8	3	
0	19	EM	9	2	
0	29	EM	9	3	
0	20	EM	10	2	
0	30	EM	10	3	
0	43	QM	3	2	
0	44	QM	4	2	
0	45	QM	5	2	EM Q1 --> SOL1 & SOL3 EM Q7 --> SOL 1 & SOL 2 & SOL 3 EM Q9 --> SOL 2 & SOL 3 EM Q10 --> SOL 2& SOL 3 QM Q6 --> SOL 1 & SOL 2 & SOL 3 CM Q4 --> SOL 1 & SOL 3 CM Q7 --> SOL 1 & SOL 3 CM Q8 --> SOL 1 & SOL 3 Outlier: EM Q4 --> SOL 3 EM Q8 --> SOL 3 QM Q3 --> SOL 2 QM Q4 --> SOL 2 QM Q5 --> SOL 2 CM Q1 --> SOL 2 CM Q9 --> SOL 1
0	36	QM	6	1	
0	46	QM	6	2	
0	56	QM	6	3	
0	71	CM	1	2	
0	64	CM	4	1	
0	84	CM	4	3	
0	67	CM	7	1	
0	87	CM	7	3	
0	68	CM	8	1	
0	88	CM	8	3	
0	69	CM	9	1	
1	11	EM	1	2	
1	2	EM	2	1	
1	12	EM	2	2	
1	22	EM	2	3	
1	4	EM	4	1	
1	6	EM	6	1	
1	16	EM	6	2	
1	26	EM	6	3	
1	8	EM	8	1	
1	9	EM	9	1	
1	31	QM	1	1	
1	41	QM	1	2	
1	51	QM	1	3	
1	32	QM	2	1	
1	42	QM	2	2	
1	52	QM	2	3	
1	33	QM	3	1	
1	53	QM	3	3	
1	34	QM	4	1	
1	54	QM	4	3	
1	35	QM	5	1	
1	55	QM	5	3	
1	37	QM	7	1	
1	47	QM	7	2	
1	57	QM	7	3	
1	38	QM	8	1	

Outlier Solution	Frequency		1= green, 0= yellow, 2 = purple
1	5		
2	10		
3	2		



Outlier	Higher	Lower
1	2	3
2	7	3
3	0	1

*rare case EM Q4, Q8 --> SOL 1, 2, 3 are all outliers.,

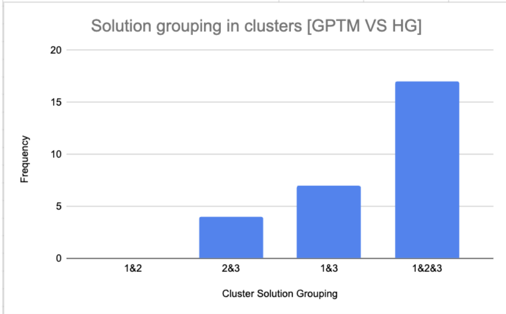


Cluster Solution Permutation	Frequency
1&2	0
2&3	4
1&3	7
1&2&3	17

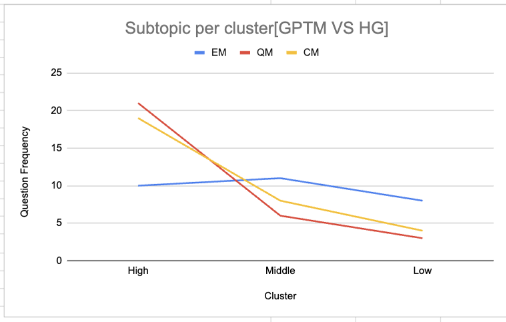
1	48	QM	8	2
1	58	QM	8	3
1	39	QM	9	1
1	49	QM	9	2
1	59	QM	9	3
1	61	CM	1	1
1	71	CM	1	2
1	81	CM	1	3
1	62	CM	2	1
1	72	CM	2	2
1	82	CM	2	3
1	63	CM	3	1
1	73	CM	3	2
1	83	CM	3	3
1	74	CM	4	2
1	65	CM	5	1
1	75	CM	5	2
1	85	CM	5	3
1	77	CM	7	2
1	79	CM	9	2
1	89	CM	9	3
1	70	CM	10	1
1	80	CM	10	2
1	90	CM	10	3
2	13	EM	3	2
2	23	EM	3	3
2	14	EM	4	2
2	5	EM	5	1
2	15	EM	5	2
2	25	EM	5	3
2	18	EM	8	2
2	10	EM	10	1
2	40	QM	10	1
2	50	QM	10	2
2	60	QM	10	3
2	66	CM	6	1
2	76	CM	6	2
2	86	CM	6	3
2	78	CM	8	2

EM Q2 --> SOL 1 & SOL 2 & SOL 3
EM Q6 --> SOL 1 & SOL 2 & SOL 3
QM Q1 --> SOL 1 & SOL 2 & SOL 3
QM Q2 --> SOL 1 & SOL 2 & SOL 3
QM Q3 --> SOL 1 & SOL 3
QM Q4 --> SOL 1 & SOL 3
QM Q5 --> SOL 1 & SOL 3
QM Q7 --> SOL 1 & SOL 2 & SOL 3
QM Q8 --> SOL 1 & SOL 2 & SOL 3
QM Q9 --> SOL 1 & SOL 2 & SOL 3
CM Q1 --> SOL 1 & SOL 2 & SOL 3
CM Q2 --> SOL 1 & SOL 2 & SOL 3
CM Q3 --> SOL 1 & SOL 2 & SOL 3
CM Q5 --> SOL 1 & SOL 2 & SOL 3
CM Q9 --> SOL 2 & SOL 3
CM Q10 --> SOL 1 & SOL 2 & SOL 3
Outlier:
EM Q1 --> SOL 2
EM Q4 --> SOL 1
EM Q8 --> SOL 1
EM Q9 --> SOL 1
CM Q4 --> SOL 2
CM Q7 --> SOL 2

EM Q3 --> SOL 2 & SOL 3
EM Q5 --> SOL 1 & SOL 2 & SOL 3
QM Q10 --> SOL 1 & SOL 2 & SOL 3
CM Q6 --> SOL 1 & SOL 2 & SOL 3
Outlier:
EM Q4 --> SOL 2
EM Q8 --> SOL 2
EM Q10 --> SOL 1
CM Q8 --> SOL 2



Cluster	EM	QM	CM
High	10	21	19
Middle	11	6	8
Low	8	3	4

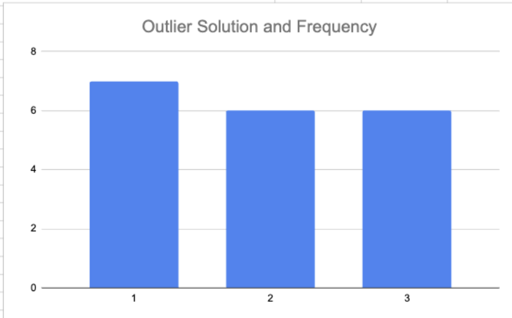


Cluster	Question	Subtopic	Actual Question	Solution
0	1	EM	1	1
0	11	EM	1	2
0	2	EM	2	1
0	12	EM	2	2
0	22	EM	2	3
0	4	EM	4	1
0	24	EM	4	3
0	6	EM	6	1
0	16	EM	6	2
0	26	EM	6	3
0	7	EM	7	1
0	17	EM	7	2
0	8	EM	8	1
0	9	EM	9	1
0	31	QM	1	1
0	41	QM	1	2
0	51	QM	1	3
0	32	QM	2	1
0	42	QM	2	2
0	52	QM	2	3
0	33	QM	3	1
0	53	QM	3	3
0	34	QM	4	1
0	37	QM	7	1
0	47	QM	7	2
0	57	QM	7	3
0	38	QM	8	1
0	48	QM	8	2
0	58	QM	8	3
0	49	QM	9	2
0	59	QM	9	3
0	81	CM	1	3
0	62	CM	2	1
0	72	CM	2	2
0	82	CM	2	3
0	63	CM	3	1
0	83	CM	3	3
0	74	CM	4	2
0	65	CM	5	1
0	75	CM	5	2
0	85	CM	5	3
0	67	CM	7	1
0	77	CM	7	2
0	68	CM	8	1
0	69	CM	9	1
0	79	CM	9	2
0	89	CM	9	3
0	70	CM	10	1
0	80	CM	10	2
0	90	CM	10	3
1	13	EM	3	2
1	23	EM	3	3

Notes

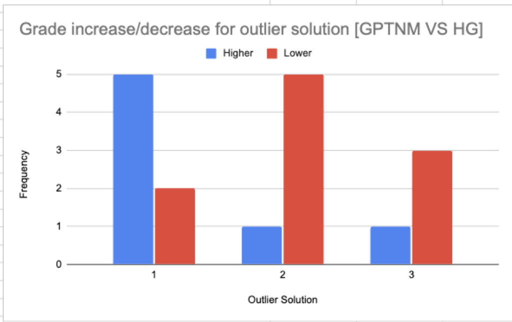
EM Q1 --> SOL 1 & SOL 2
EM Q2 --> SOL 1 & SOL 2 & SOL 3
EM Q4 --> SOL 1 & SOL 3
EM Q6 --> SOL 1 & SOL 2 & SOL 3
EM Q7 --> SOL 1 & SOL 2
QM Q1 --> SOL 1 & SOL 2 & SOL 3
QM Q2 --> SOL 1 & SOL 2 & SOL 3
QM Q3 --> SOL 1 & SOL 3
QM Q7 --> SOL 1 & SOL 2 & SOL 3
QM Q8 --> SOL 1 & SOL 2 & SOL 3
QM Q9 --> SOL 2 & SOL 3
CM Q2 --> SOL 1 & SOL 2 & SOL 3
CM Q3 --> SOL 1 & SOL 3
CM Q5 --> SOL 1 & SOL 2 & SOL 3
CM Q7 --> SOL 1 & SOL 2
CM Q9 --> SOL 1 & SOL 2 & SOL 3
CM Q10 --> SOL 1 & SOL 2 & SOL 3
Outlier:
EM Q8 --> SOL 1
EM Q9 --> SOL 1
QM Q4 --> SOL 1
CM Q1 --> SOL 3
CM Q4 --> SOL 2
CM Q8 --> SOL 1

Outlier Solution	Frequency		
			0 = green, 1= yellow, 2 = purple
1	7		
2	6		
3	6		



Outlier	Higher	Lower
1	5	2
2	1	5
3	1	3

*rare case EM,CM Q8 --> SOL 1, 2, 3 are all outliers.,

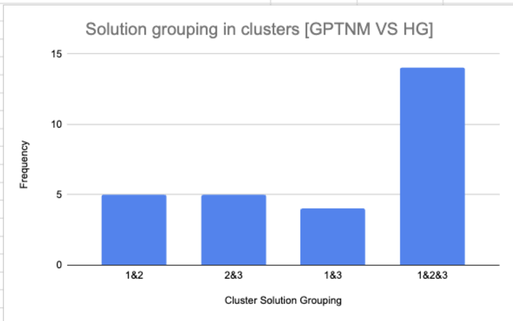


Cluster Solution Permutation	Frequency
1&2	5
2&3	5
1&3	4
1&2&3	14

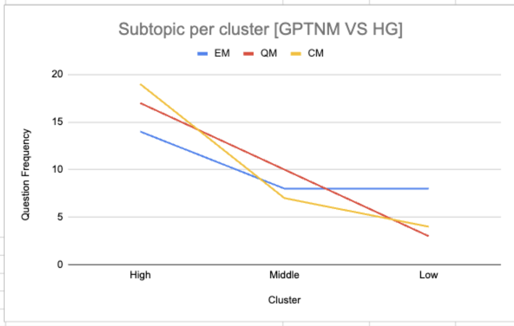
1	14	EM	4	2
1	5	EM	5	1
1	15	EM	5	2
1	25	EM	5	3
1	18	EM	8	2
1	10	EM	10	1
1	40	QM	10	1
1	50	QM	10	2
1	60	QM	10	3
1	66	CM	6	1
1	76	CM	6	2
1	86	CM	6	3
1	78	CM	8	2
2	21	EM	1	3
2	3	EM	3	1
2	27	EM	7	3
2	28	EM	8	3
2	19	EM	9	2
2	29	EM	9	3
2	20	EM	10	2
2	30	EM	10	3
2	43	QM	3	2
2	44	QM	4	2
2	54	QM	4	3
2	35	QM	5	1
2	45	QM	5	2
2	55	QM	5	3
2	36	QM	6	1
2	46	QM	6	2
2	56	QM	6	3
2	39	QM	9	1
2	61	CM	1	1
2	71	CM	1	2
2	73	CM	3	2
2	64	CM	4	1
2	84	CM	4	3
2	87	CM	7	3
2	88	CM	8	3

EM Q3 --> SOL 2 & SOL 3
EM Q5 --> SOL 1 & SOL 2 & SOL 3
QM Q10 --> SOL 1 & SOL 2 & SOL 3
CM Q6 --> SOL 1 & SOL 2 & SOL 3
Outlier:
EM Q4 --> SOL 2
EM Q8 --> SOL 2
EM Q10 --> SOL 1
CM Q8 --> SOL 2

EM Q9 --> SOL 2 & SOL 3
EM Q10 --> SOL 2 & SOL 3
QM Q4 --> SOL 2 & SOL 3
QM Q5 --> SOL 1 & SOL 2
QM Q6 - SOL 1 & SOL 2 & SOL 3
CM Q1 --> SOL 1 & SOL 2
CM Q4 --> SOL 1 & SOL 3
Outlier:
EM Q1 --> SOL 3
EM Q3 --> SOL 1
EM Q7 --> SOL 3
EM Q8 --> SOL 3
QM Q3 --> SOL 2
QM Q9 --> SOL 1
CM Q3 --> SOL 2
CM Q7 --> SOL 3
CM Q8 --> SOL 3



Cluster	EM	QM	CM
High	14	17	19
Middle	8	10	7
Low	8	3	4



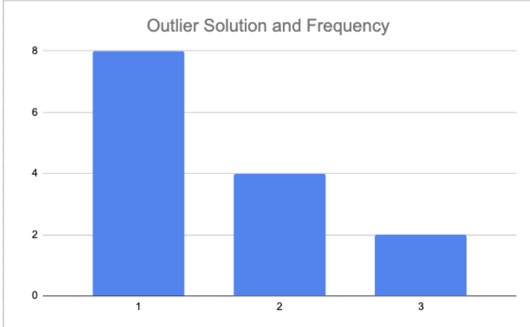
Cluster	Question	Subtopic	Actual Question	Solution
0	1	EM	1	1
0	11	EM	1	2
0	21	EM	1	3
0	4	EM	4	1
0	24	EM	4	3
0	7	EM	7	1
0	17	EM	7	2
0	27	EM	7	3
0	19	EM	9	2
0	29	EM	9	3
0	43	QM	3	2
0	44	QM	4	2
0	35	QM	5	1
0	45	QM	5	2
0	55	QM	5	3
0	36	QM	6	1
0	48	QM	8	2
0	39	QM	9	1
0	61	CM	1	1
0	71	CM	1	2
0	64	CM	4	1
0	65	CM	5	1
0	67	CM	7	1
0	77	CM	7	2
0	87	CM	7	3
0	68	CM	8	1
0	69	CM	9	1
1	2	EM	2	1
1	12	EM	2	2
1	22	EM	2	3
1	6	EM	6	1
1	16	EM	6	2
1	26	EM	6	3
1	8	EM	8	1
1	9	EM	9	1
1	31	QM	1	1
1	41	QM	1	2
1	51	QM	1	3
1	32	QM	2	1
1	42	QM	2	2
1	52	QM	2	3
1	33	QM	3	1
1	53	QM	3	3
1	34	QM	4	1
1	54	QM	4	3
1	37	QM	7	1
1	47	QM	7	2
1	57	QM	7	3
1	38	QM	8	1
1	58	QM	8	3

Notes

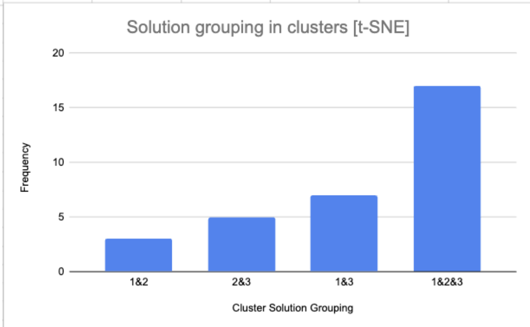
EM Q1 --> SOL 1 & SOL 2 & SOL 3
EM Q4 --> SOL 1 & SOL 3
EM Q7 --> SOL 1 & SOL 2 & SOL 3
EM Q9 --> SOL 2 & SOL 3
QM Q5 --> SOL 1 & SOL 2 & SOL 3
CM Q1 --> SOL 1 & SOL 2
CM Q7 --> SOL 1 & SOL 2 & SOL 3
Outlier:
QM Q3 --> SOL 2
QM Q4 --> SOL 2
QM Q6 --> SOL 1
QM Q8 --> SOL 2
QM Q9 --> SOL 1
CM Q4 --> SOL 1
CM Q5 --> SOL 1
CM Q8 --> SOL 1
CM Q9 --> SOL 1

1 = green, 2= yellow, 0 = purple

Outlier Solution	Frequency
1	8
2	4
3	2



Cluster Solution	Frequency
1&2	3
2&3	5
1&3	7
1&2&3	17



Cluster	EM	QM	CM
High	8	17	15
Middle	10	8	9
Low	12	5	6

1	49	QM	9	2	EM Q2 --> SOL 1 & SOL 2 & SOL 3 EM Q6 --> SOL 1 & SOL 2 & SOL 3 QM Q1 --> SOL 1 & SOL 2 & SOL 3 QM Q2 --> SOL 1 & SOL 2 & SOL 3 QM Q3 --> SOL 1 & SOL 3 QM Q4 -->.SOL 1 & SOL 3 QM Q7 --> SOL 1 & SOL 2 & SOL 3 QM Q8 --> SOL 1 & SOL 3 QM Q9 --> SOL 2 & SOL 3 CM Q2 --> SOL 1 & SOL 2 & SOL 3 CM Q3 --> SOL 1 & SOL 2 & SOL 3 CM Q5 --> SOL 2 & SOL 3 CM Q9 --> SOL 2 & SOL 3 CM Q10 --> SOL 1 & SOL 2 & SOL 3 Outlier: EM Q8 --> SOL 1 EM Q9 --> SOL 1 CM Q1 --> SOL 3 CM Q4 --> SOL 2
1	59	QM	9	3	
1	81	CM	1	3	
1	62	CM	2	1	
1	72	CM	2	2	
1	82	CM	2	3	
1	63	CM	3	1	
1	73	CM	3	2	
1	83	CM	3	3	
1	74	CM	4	2	
1	75	CM	5	2	
1	85	CM	5	3	
1	79	CM	9	2	
1	89	CM	9	3	
1	70	CM	10	1	
1	80	CM	10	2	
1	90	CM	10	3	
2	3	EM	3	1	EM Q3 --> SOL 1 & SOL 2 & SOL 3 EM Q4 --> SOL 1 & SOL 2 EM Q5 --> SOL 2 & SOL 3 EM Q8 --> SOL 2 & SOL 3 EM Q10 --> SOL 1 & SOL 2 & SOL 3 QM Q6 --> SOL 1 & SOL 2 QM Q10 --> SOL 1 & SOL 2 & SOL 3 QM Q6 --> SOL 1 & SOL 2 & SOL 3 QM Q8 --> SOL 2 & SOL 3 Outlier: CM Q4 --> SOL 3
2	13	EM	3	2	
2	23	EM	3	3	
2	5	EM	4	1	
2	14	EM	4	2	
2	15	EM	5	2	
2	25	EM	5	3	
2	18	EM	8	2	
2	28	EM	8	3	
2	10	EM	10	1	
2	20	EM	10	2	
2	30	EM	10	3	
2	46	QM	6	2	
2	56	QM	6	3	
2	40	QM	10	1	
2	50	QM	10	2	
2	60	QM	10	3	
2	84	CM	4	3	
2	66	CM	6	1	
2	76	CM	6	2	
2	86	CM	6	3	
2	78	CM	8	2	
2	88	CM	8	3