MIRNA ARIVALAGAN - 220142881

MACHINE LEARNING – SIT720

Trimester 2 2021

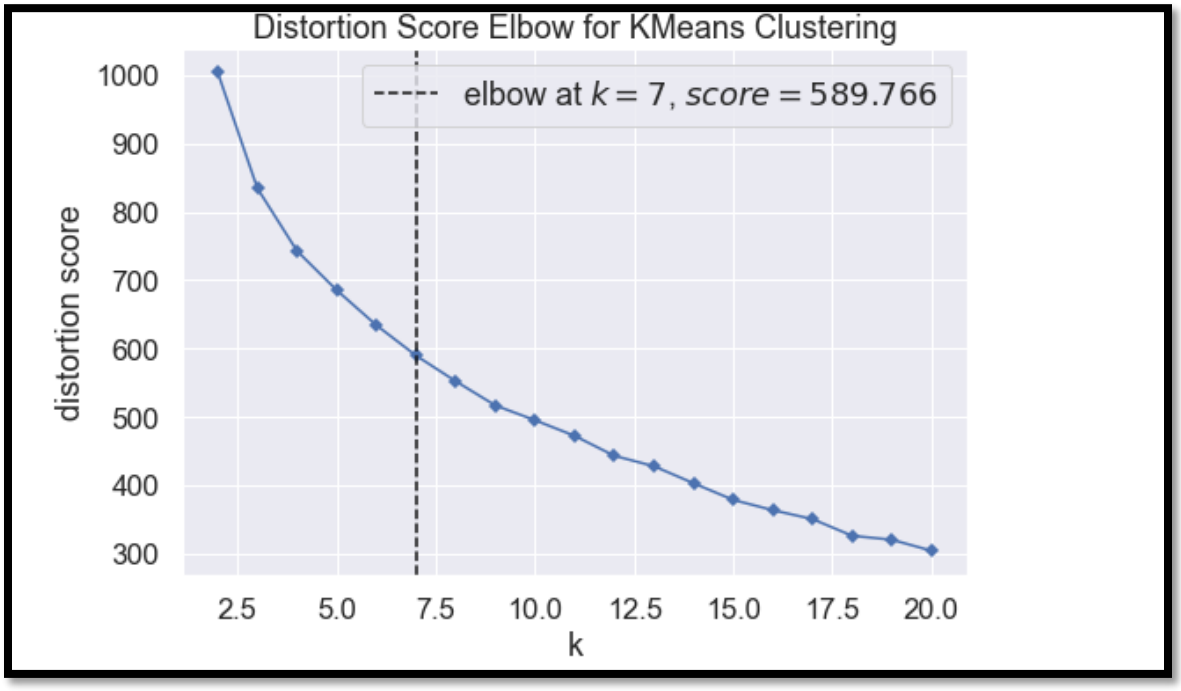
Contents

Part 1

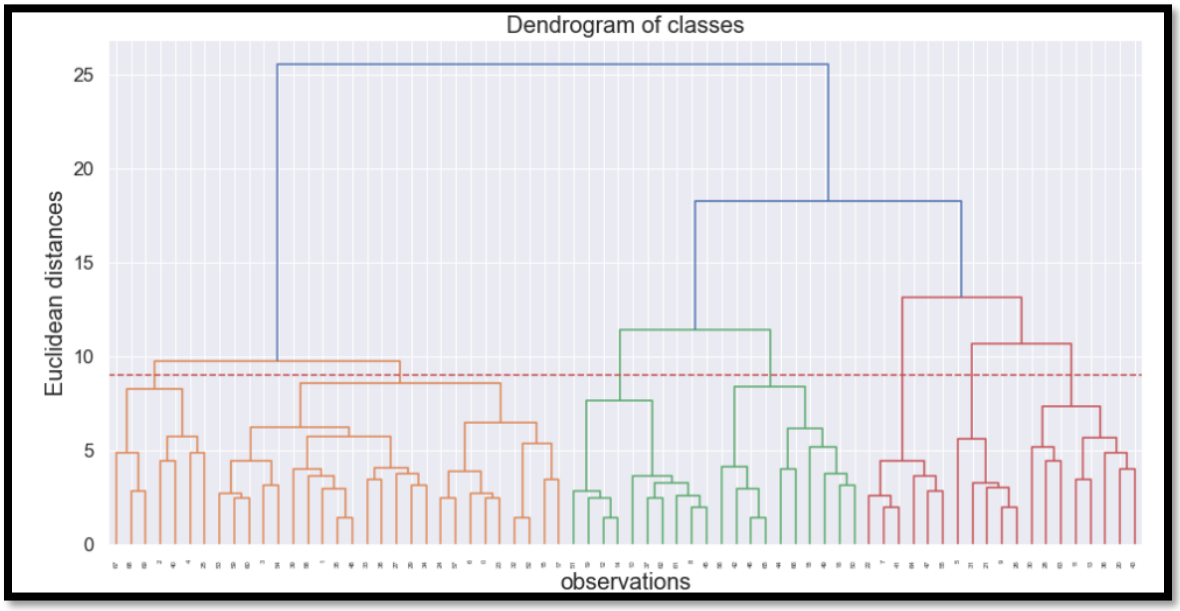
Question 1

Determine the number of subgroups from the dataset using attributes 3 to 205 i.e., exclude attributes 1, 2 and 206. Is this number same as number of classes presented by attribute 206? Explain and justify your findings.

To identify the number of subgroups from the dataset given, I passed the data set through a couple of different algorithms. The first method I used, was the elbow method. The elbow method works by running through the inputted range of clusters, as it is running through the range, the quality of the clusters improves, and once it reaches a point where the quality does not improve any further, the elbow method will identify the optimum number of clusters as the bend in the elbow when the average within cluster sum of squares of distance between data points is plotted. With our data set, the elbow method indicated that the ideal number of clusters was 7.



To validate this, I ran the same data set through a dendrogram using the linkage method of ward, which is the analysis of variance (ANOVA) to determine the distance between data points. This method starts off with having each data point as an individual cluster and works its way to merge with other data points that are similar and keeps merging until a single cluster is formed as the main cluster (Patlolla, 2018). With the dendrogram method, I have been able to identify a cut off point where there are 7 significant cluster. Do note though that the dendrogram is open to interpretation, as I could move a step up and end up with 3 big clusters. To sum it up, in both instances, I was able to detect 7 clusters which matches the number of classes that was present in the data set.



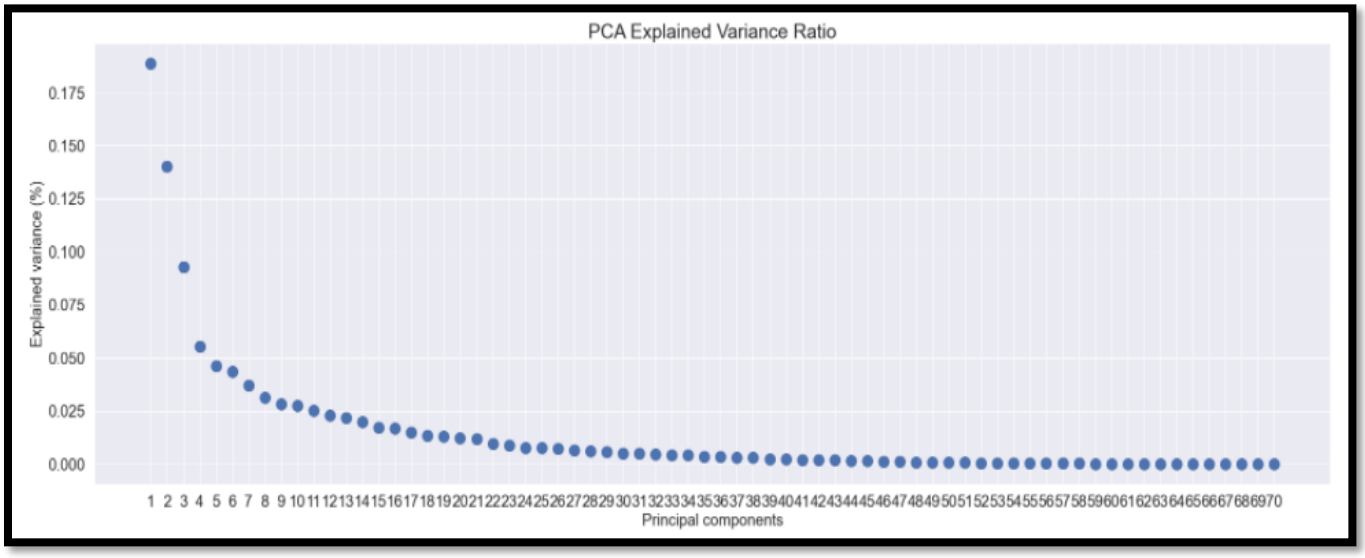
Question 2

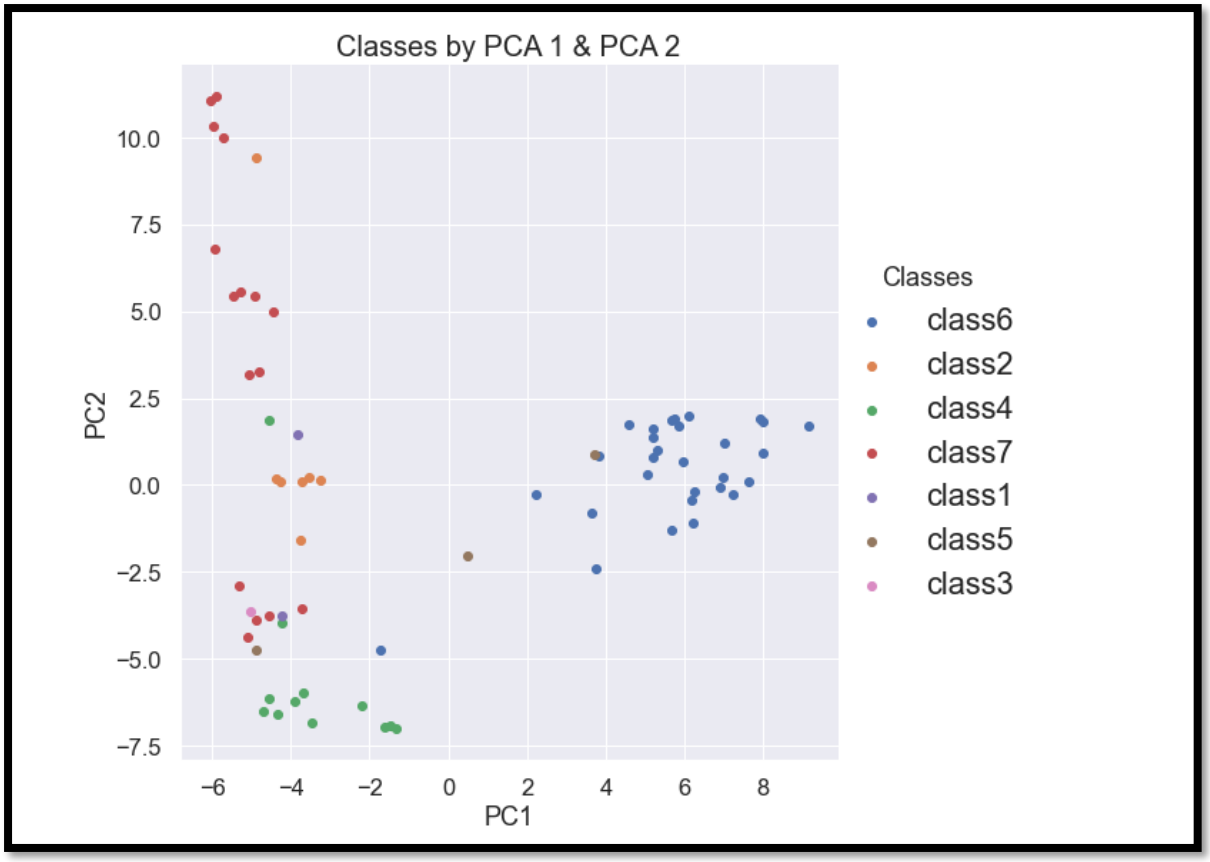
Is this data facing curse of dimensionality? If so, then how to solve this problem. Explain with a two-dimensional plot and report relevant loss of information.

Curse of dimensionality occurs when the data set being used has too many features, more specifically where the number of features is greater than the number of observations in the dataset used. When there are too many features, it's makes it more difficult for machine learning models to identify patterns in the data. It also causes the issue of sparsity where there is more distance between data points, thus reducing the models' capabilities in generating specific clusters which is the purpose of this assessment. (Yiu, 2019)

At a glance, we know there was a total of 70 observations and 203 features selected, so we can conclude that the dataset provided faces the curse of dimensionality. To address this issue, Principal component analysis (PCA) can be used to identify features that are the most meaningful to be fitted into the model. PCA works in a way that it finds the least number of features that contains the most variation of information in the dataset.

After fitting the data to the Sklearn PCA algorithm, we can observe that 70% of variation of information is captured in the first 10 features. The first 3 components capture just under 50% of the variation (21.19%), and the first 2 components captures 32.9% of variation. What this means is that when we use the data set to fit into a machine learning model, we are only losing about 30% of the information if we use the 1st 10 features instead of all 203 features, thus reduces the chance of overfitting the model.


PCA Explained Variance Ratio
Explained variance (%)
0.175
0.150
0.125
0.100
0.075
0.050
0.025
0.000
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70
Principal components


Classes by PCA 1 & PCA 2
PC2
10.0
7.5
5.0
2.5
0.0
−2.5
−5.0
−7.5
−6
−4
−2
0
2
4
6
8
PC1
Classes
class6
class2
class4
class7
class1
class5
class3

Question 3

After applying principal component analysis (PCA) on a given dataset, it was found that the percentage of variance for the first N components is X%. How is this percentage of variance computed?

After fitting the data into the PCA algorithm, we have found that the percentage for variance for the first 10 components is 70%. To obtain the percentage of variance, when running the data set through the PCA algorithm, it first identifies the strongest feature, thus having the highest eigen value, once the algorithm identifies the strongest feature, it then identifies the next strongest feature, but this feature should not be correlated to the first feature, once this is done, the algorithm then tries to identify the 3rd strongest feature that is uncorrelated with the first two features and repeats this process until it goes through all features. Once all the features variance is computed, the algorithm will calculate the sum of squares of each principal component, and these sums of squares divided by the sample size minus 1 to give us the percentage of variance for each feature.

Part 2

Question 4

Create a machine learning (ML) model for predicting "weight" using all features except "NObeyesdad" and report observed performance. Explain your results based on following criteria:

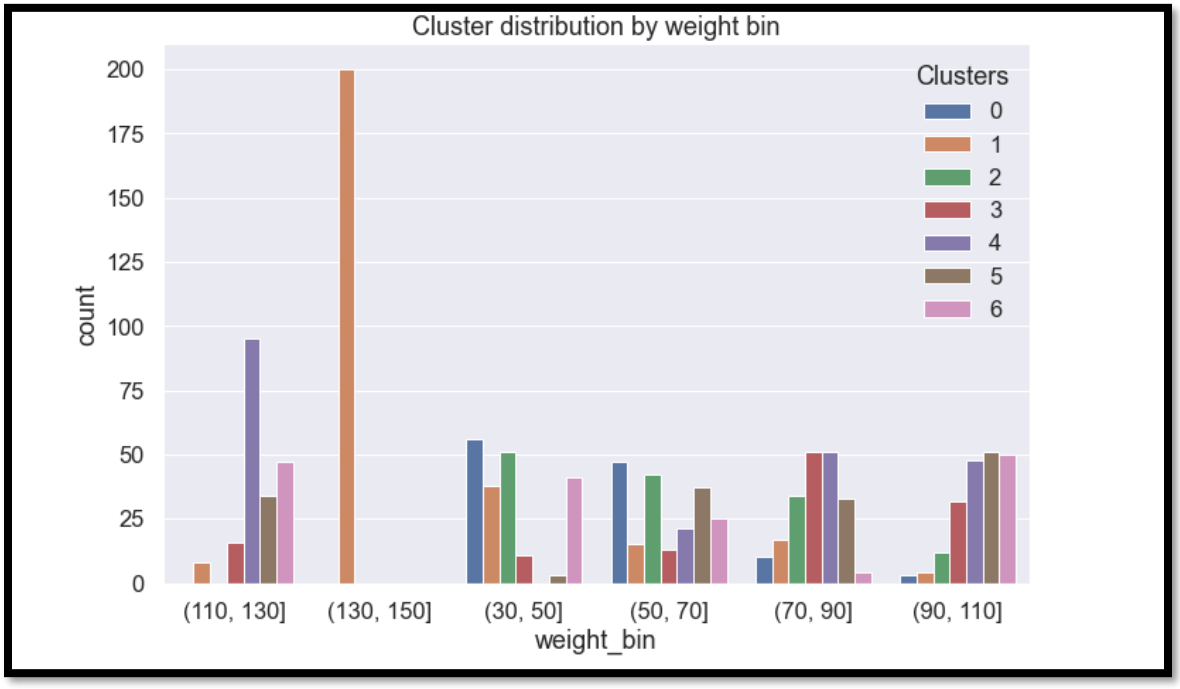
a. What model have you selected for solving this problem and why?

I have selected to use K means as a model to predict the weight. K means is an algorithm that partitions data into subgroups, with no overlap in data points. Reason why I picked this model is because I have seen how this model have been used to segment customers, so given the task that needs to be solved, I've approached this task in a way that the model groups together individuals based on their features to predict the weight range.

As K means uses distance-based measurements to determine similarities between data points, I believe this will work well when we pass through the features that are in our dataset as weights of people can be associated with lifestyle choices and often family history. Given our dataset can describe these associations as we have information about the individuals Alcohol consumption and physical activities information for example, the algorithm can identify these similarities in the data set and group together individuals that are similar which can then be used to determine an approximate weight range.

However, even after running the data set through the algorithm twice, once prior to reducing the dimensions and balancing the dataset, and the second time after, we were still not able to

predict weight accurately. You can observe that the clusters are still present in a wide range of weight groups.



b. Have you made any assumption for the target variable? If so, then why?

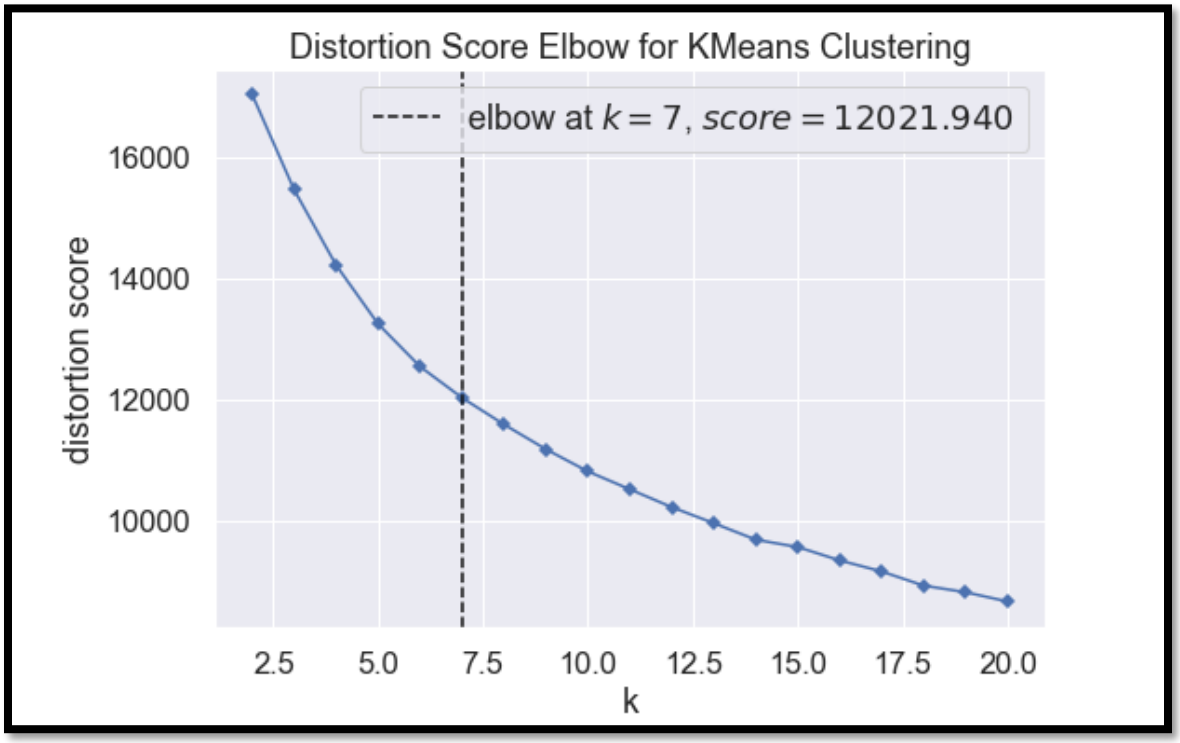
The target variable the weight bins that I have grouped together to predict the weight range. The first assumption that I have made was that there is a linear relationship with the weight and the number of physical activities an individual does, where the weight increases when there is lesser physical activity involved. I have also assumed that there is a linear relationship with weight and the individual's caloric food intake.

c. What have you done with text variables? Explain.

I have converted the text variables into numerical variables by one hot encoding these variables using pythons' pandas get dummies function. One hot encoding converted the variables into a binary vector representation, where each variable in each feature now represents a new feature. I have chosen to use this method of encoding instead of using a label encode, as the categorical variables that are present in the data set has no ordinal order to them. Encoding via a label encode represents some sort of order between the variables.

d. Have you optimised any model parameters? What is the benefit of this action?

I have passed the dataset through the K means algorithm to identify the optimum number of clusters within the range of 2 to 20 clusters. In doing so, I was able to plot an elbow graph that stipulated the best number of clusters in our dataset is 7 clusters.

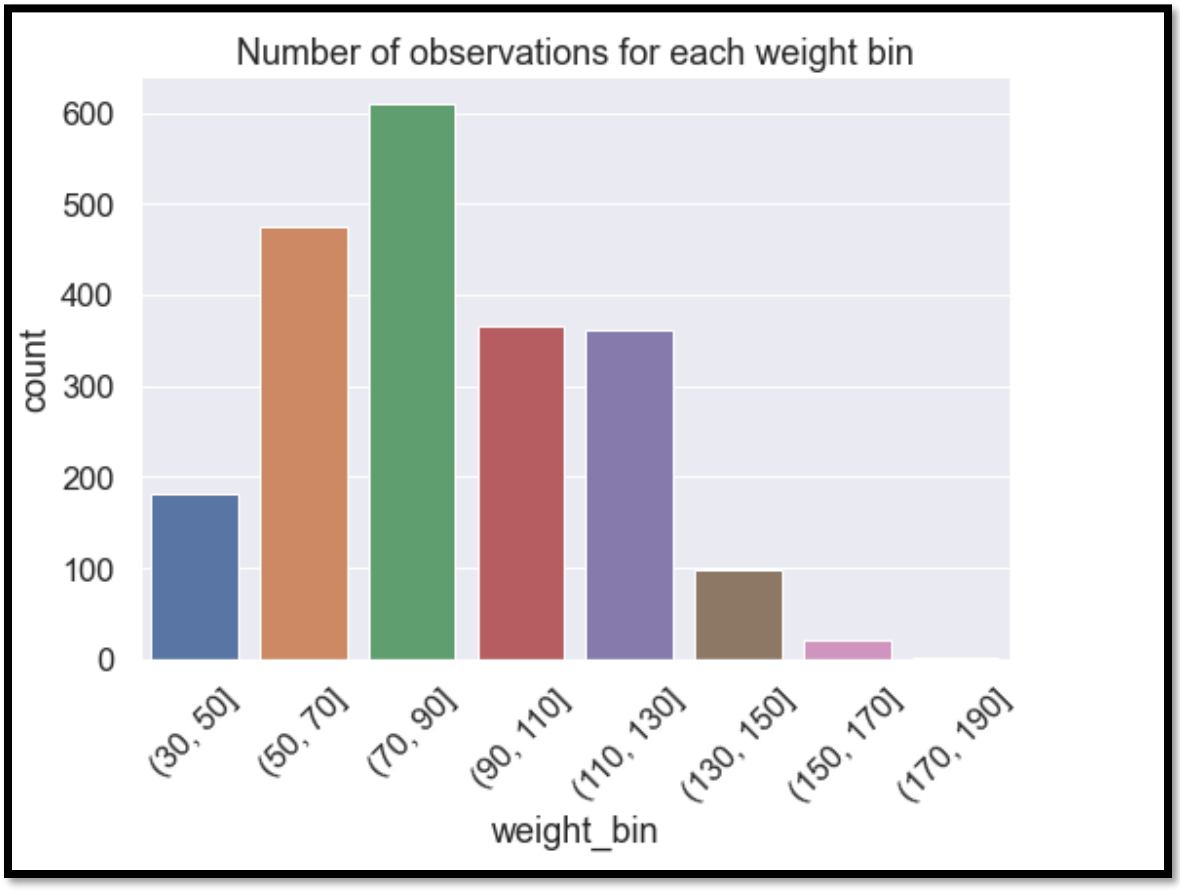


I have also ran the model a few times through a variety of number in the parameter n_init, and found 80 to have produced the best result. n-init represents, the number of initialization attempts for centroid clusters. I have also set the max_iter parameter to run a maximum of 500 times, and what this does it ensures that the algorithm has explored the entire feature space when determining the clusters.

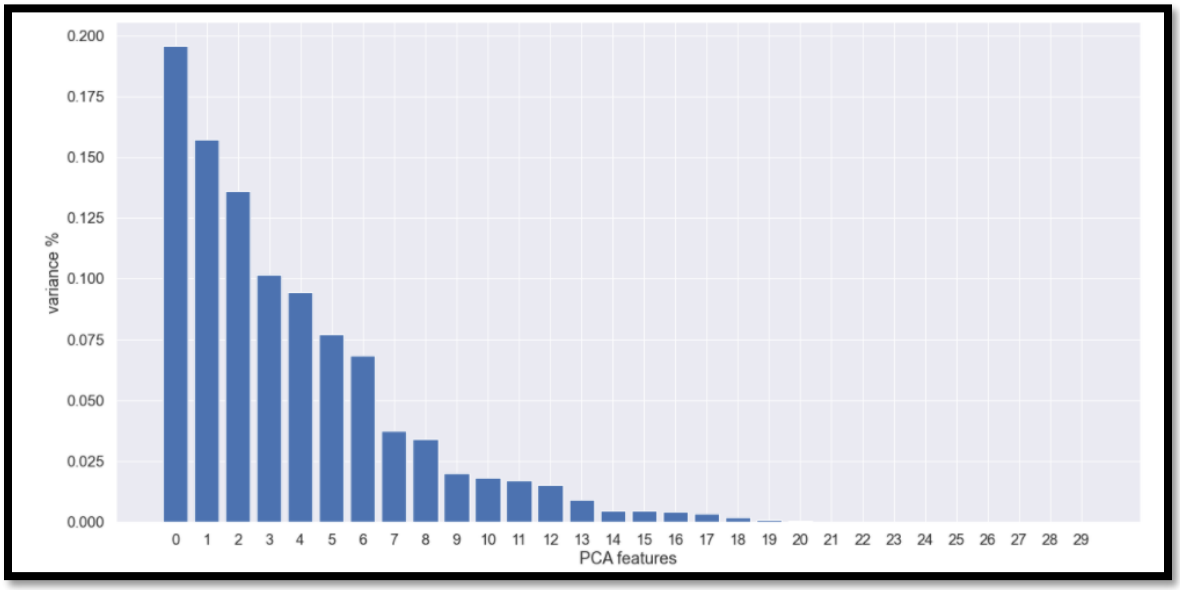
e. Have you applied any step for handling overfitting or underfitting issue? What is that?

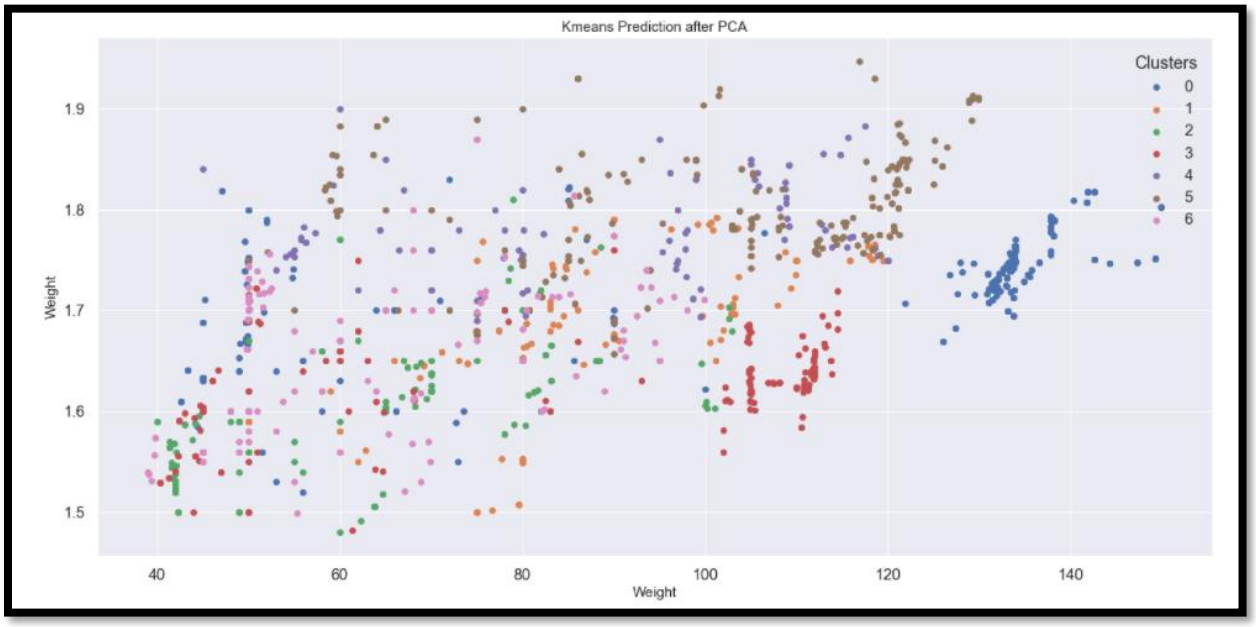
As you would see in the python solution supplied for this assessment, for this model, I initially ran the algorithm through the entire dataset, however due to having imbalanced classes with the weight distributions and too many features, the model didn't not predict accurately. It scored a low 13% silhouette score and 24% Mallows score.

I then redid the model, but I balanced the data set by taking a sample of 200 rows for each weight bin groups to avoid overfitting the models. Some slight variation in imbalanced classes is acceptable when building models, but in our scenario, the variations are quite large which can be observed below.



I have also used the same dataset and did some PCA analysis to identify features that are most meaningful to avoid over fitting the model. What I have used was a total of 10 features of the 29 that was present in the dataset after one hot encoding, I have been able to improve the Silhouette score from 13% to 19% and Fowlkes Mallows score from 24% to 37% by running PCA. This means that some of the features in the dataset were highly correlated which can cause overfitting. As you can see below, even after running PCA and balancing the dataset, the cluster distribution improved a little, but it's still present in most weight groups. We can conclude that unsupervised learning isn't suitable to predict weight, but we can use it to group together individuals of similar features and obtain the average weight of those groups.





Question 5

a. Report classification performance scores. Select scores that you think best for describing the model performance with appropriate justification:

I have developed 2 models for this question, the first model is using the K modes prototype algorithm and the second model uses the Agglomerative Clustering algorithm. Below are the

classification scores for the train & test splits for both models that has been developed. I've also used the Fowlkes Mallows scores alongside this as this metric is a more accurate metric to use for unsupervised learning.

Kmodes Prototype Algorithm

Testing Results

	precision	recall	f1-score	support
0	0.55	0.70	0.61	207
1	0.68	0.53	0.59	253
accuracy			0.60	460
macro avg	0.61	0.61	0.60	460
weighted avg	0.62	0.60	0.60	460

Training Results

	precision	recall	f1-score	support
0	0.59	0.65	0.62	660
1	0.65	0.58	0.61	719
accuracy			0.62	1379
macro avg	0.62	0.62	0.62	1379
weighted avg	0.62	0.62	0.62	1379

```
Test Fowlkes Mallows Score is 0.5275663732557019

Train Fowlkes Mallows Score is 0.527780065473605
```

Agglomerative Clustering Algorithm

Test Data Set Results

	precision	recall	f1-score	support
0	0.20	0.20	0.20	207
1	0.35	0.35	0.35	253
accuracy			0.28	460
macro avg	0.27	0.27	0.27	460
weighted avg	0.28	0.28	0.28	460

Training Results

	precision	recall	f1-score	support
0	0.70	0.95	0.81	660
1	0.93	0.64	0.76	719
accuracy			0.79	1379
macro avg	0.82	0.79	0.78	1379
weighted avg	0.82	0.79	0.78	1379

```
Test Fowlkes Mallows Score is 0.5971606460380143

Train Fowlkes Mallows Score is 0.676613346581987
```

As unsupervised learning methods do not work well using labelled target variables, we should use the classification report cautiously, as for example the clustering numbers allocated by these unsupervised learning algorithms do not indicate an order and does not tie back to the target variable that we are trying to predict, which in this instance is either class 1 or class 0, where class 1 is classified as obese, and class 0 is classified as not obese. From the sklearn classification report, the metric that I have used to assess my model is the F1-scores, as this shows the harmonic average of the precision and recall metric, it tells us what percentage of positive predictions are correct.

In our instance, the K modes prototype algorithm worked quite well in both the training and testing results. The F1 score was about 60% in both train and test rounds, this means we only have about 40% of false positives and negatives. However when we use this metric for the agglomerative clustering model, the training score, was high about 80% but the test F1 score for this model was low at about 20-35%. This indicates that there is overfitting in this instance. However as mentioned earlier, we need to use these classification scores with a grain of salt, as the model we have built is an unsupervised learning model, where classifications do not work.

To overcome this classification issue, I have also opted to use the Fowlkes Mallows Score, as this method measures the similarity of two clusters as a set of points and like the F1 score, it

measures the geometric mean between precision and recall. For the K modes prototype model, the Fowlkes mallows score both training and testing scored 52%, so I would say this in not bad but not great performance either. When we look at the agglomerative clustering technique, we can see that the training data set performed high at 67% but the test data set scored slightly lower at 59%, which I suspect is caused due to overfitting the data.

Between the 2 models, I would say the K modes prototype model is the better model to use to predict the classes as the difference in scores for both the classification scores method and Fowlkes Mallows Score, they scored close to each other in both the train and test data sets.

b. Have you taken any step to check generalisability of the model? What is that and how it ensures generalisability:

Yes I have, in both my models that I have supplied, I have split the data into test and train data sets. By splitting and testing the datasets, this ensures that the model works in real life datasets. I have applied 75%:25% split to my dataset. Once I have split the dataset, I have dropped the Class column in both sets prior to fitting it to the model, this ensures that we are not overfitting the data, and we can use this feature as our ground truth to compare with the prediction labels to get a picture of how accurate the model is.

For my first K modes prototype model, as described in question 4 (a) – you will see that in both instances, the train and test produced results that are really close to each other, so I would say this model is suitable and is generalisable. However, this isn't the case with the agglomerative clustering model that I have developed, there is a significant difference in results between the train and test datasets, so in this instance, this model is not generalisable. I believe that the K modes model has performed well because the K modes prototype algorithm was specifically designed to handle mixed data types which is the case with the dataset that was used.

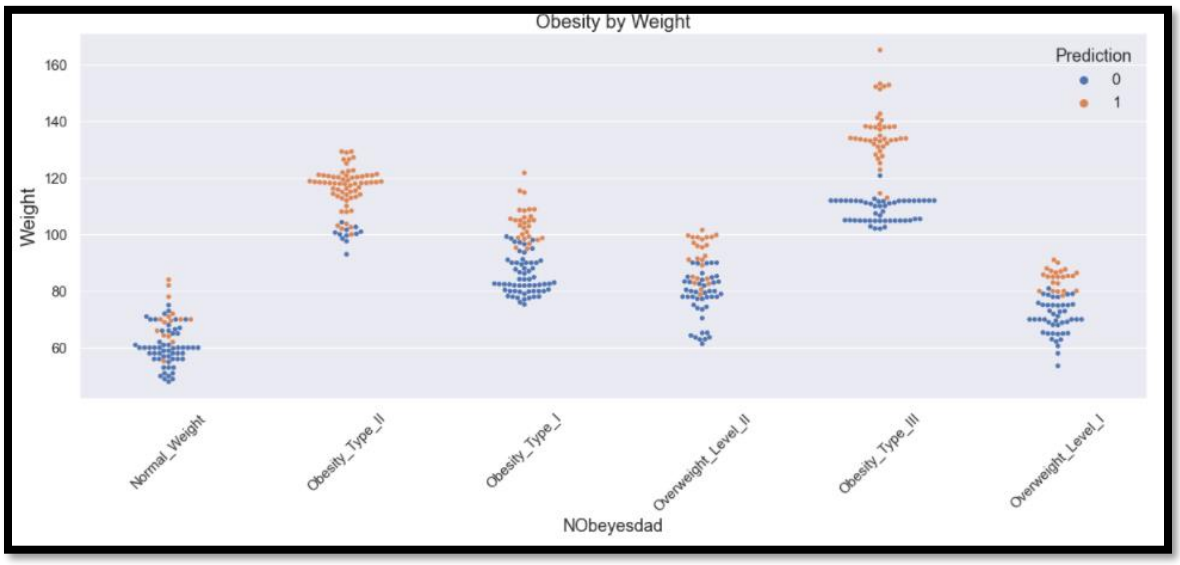
c. Can you design and develop any other model for solving this problem? If so, then why have you used the reported one? Give your justification:

I have designed a second model to solve this problem. I have decided to use agglomerative clustering as the selected algorithm. Reason why I have chosen this is because this algorithm works with a bottom-up approach, where it first create single data point clusters and then works it way up to create more clusters. As for this task, we must predict either class 1 or class 0, I believe this algorithm would work well to identify 2 key groups. I also decided to use this algorithm, as it does not actually identify the number of clusters, it is up to us to decide where the cut off point is, or pass the number of clusters into the algorithm, I thought this will work well since we already know we want to identify 2 different clusters.

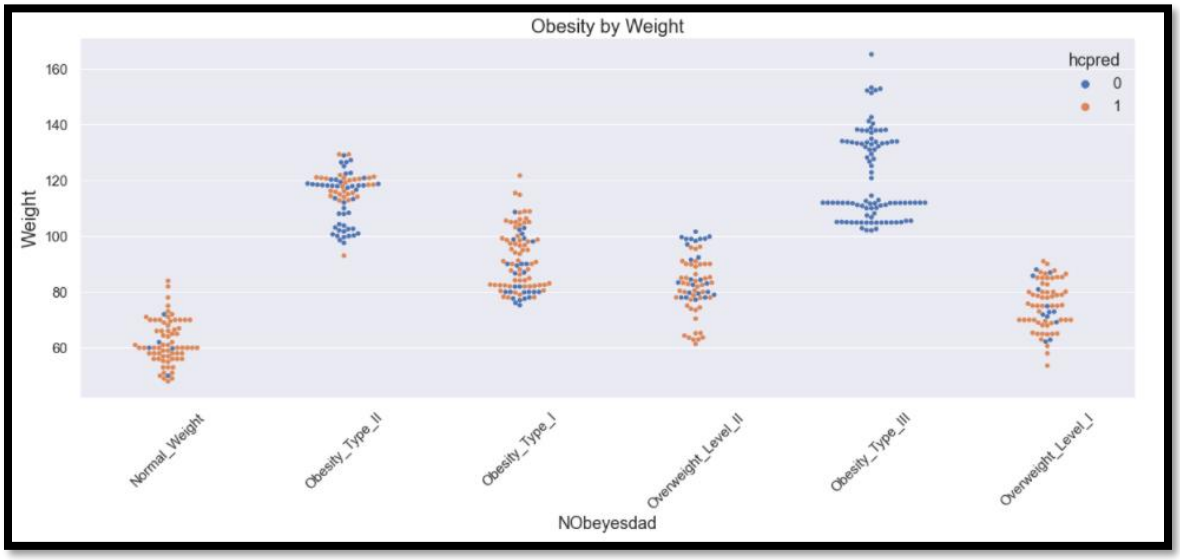
Between the 2 models, the K modes Prototype model seemed to have performed and allocated the predictions a little better than the agglomerative clustering method. This can be observed

below where we know the Nobeyesdad groups of Normal Weight, Overweight 1 and Overweight 2 has been assigned as 0 as our new class, and the Obesity Type 1-3 has been assigned as 1 in our new class.

With the Kmodes algorithm, from the visual below, you can observe that the distribution of prediction is not bad, where Obesity type 2 and type 3, most of the predictions were set to 1, which is accurate, Obesity type 1, the model had allocated most of it as not obese with a prediction of 0. I think this could be due to the weights being on the lower range. Most of the normal weight and overweight classes has also got a prediction of 0 which is accurate.



However, when you compare the distributions of predictions for the agglomerative clustering model, it is obvious that it is not as accurate as the K modes model, from the image below, you can see an entire group of Obesity type 3 has been classified as class 0, where 0 is our non-obese group that we have assigned. In saying that though, it is important to know that unsupervised learning methods is not designed to predict certain target variables.



Question 6

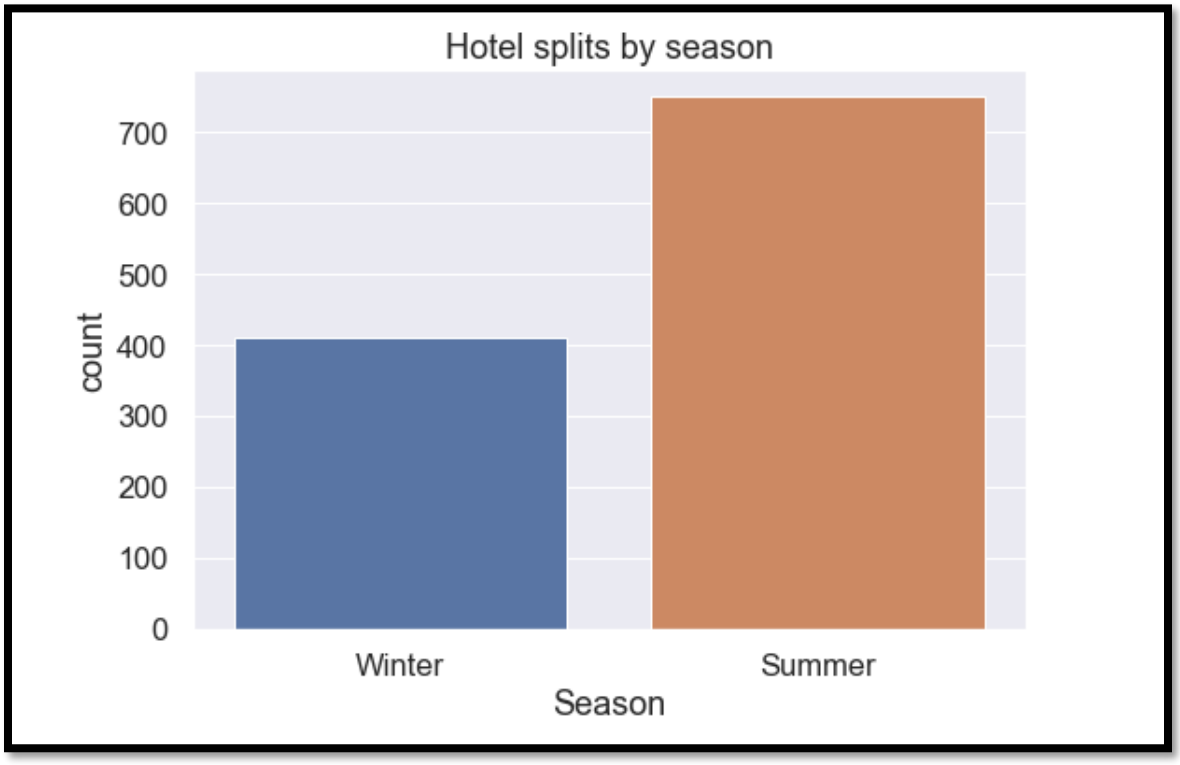
Suppose that a company has a number (>=500) of resorts around the globe.

a. Identify a list of features (>=5) that can be used to describe these resorts.

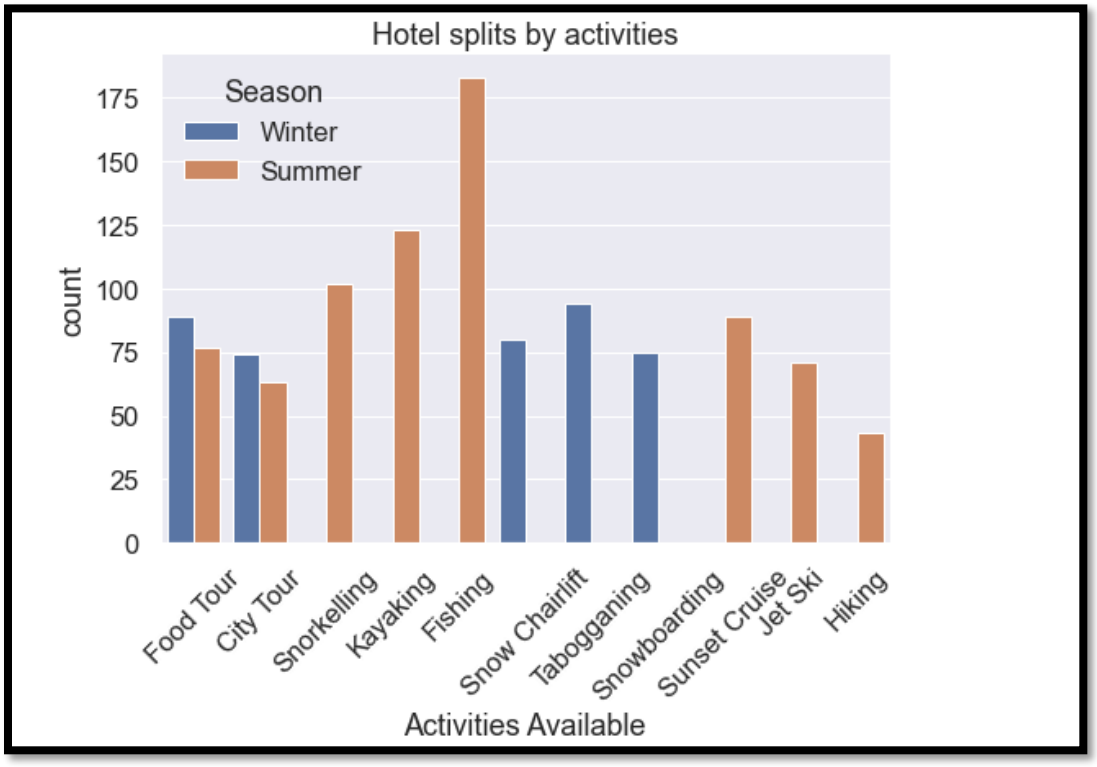
b. Create a dataset (rows>=500) and explain all variables. You can generate data either synthetically or collecting from similar datasets. Submit your created dataset. In addition, please provide links in case you have collected the dataset.

I have used a dataset that I have found online for this task and manipulated it to fit our assessment task. What I have done was used the list of hotel names, city names, ratings and review count from the data set that I have found and then added additional features to the data set.

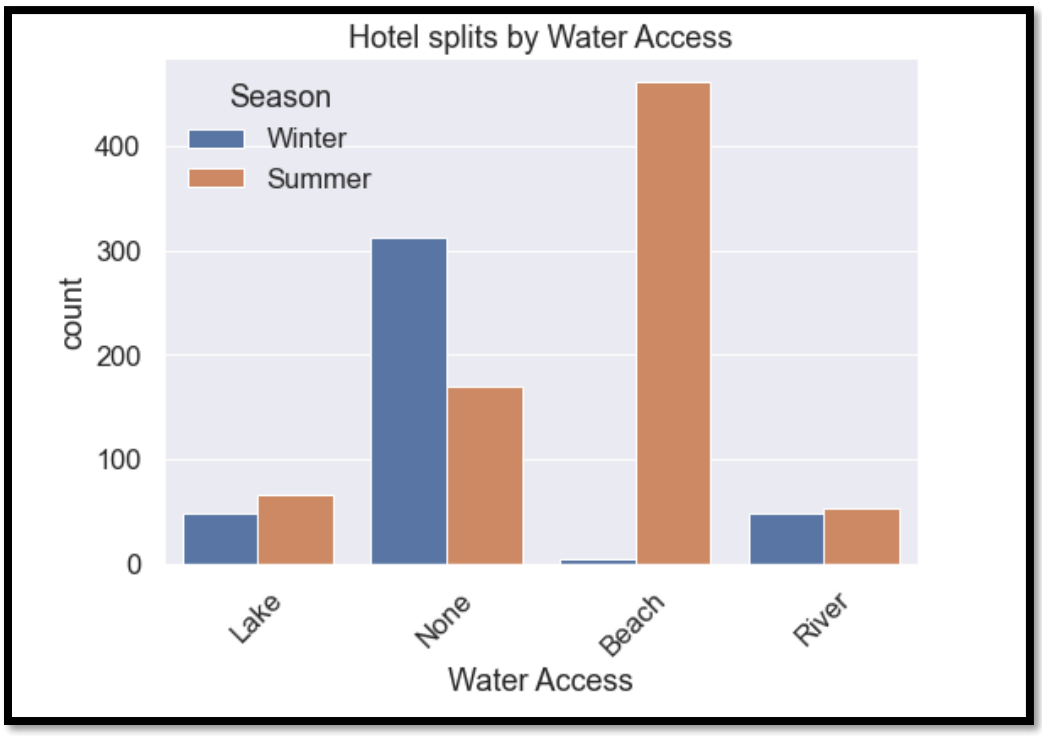
I have designed the dataset in a way that it can be used to provide a recommendation of hotels to book when visiting and travelling in Australia. As there are 2 main holiday seasons in Australia, I have added an additional feature named Season which has Winter or Summer as the option. So if the customer selects Summer, it will bring up a list of hotels that is suitable.



I have also added as features in the data set to include Hotel Inclusions such as Free Wi-Fi, Indoor or Outdoor Pools, Spa and Fireplace. There are 2 features for these hotel inclusions which is listed as inclusion 1 and 2. I have also included an activities available feature which has options such as Food Tours, City Tours, Snorkelling, Jet Ski, Kayaking, Fishing, Snow Chairlift and Tobogganing.



Also present in my data set are 2 seasonal indicator features such as Water Access where Beach, Lake and River are the options and another binary feature for Snow Access.



There is a total of 1163 observations and 10 features in my dataset.

Hotel name	city_name	Rati	reviews cou	Inclusion 1	Inclusion 2	Activities Availab	Water Acce	Snow Acce	Seasor
InterContinental Adelaide	Adelaide	4	2915	Free Wifi	Spa	Food Tour	Lake	No	Winter
Holiday Inn Express Adelaide City Centre	Adelaide	4.5	386	Free Wifi	Indoor Pool	Food Tour	Lake	No	Winter
Franklin Apartments	Adelaide	4	1004	Free Wifi	Indoor Pool	Food Tour	Lake	No	Winter
Adabco Boutique Hotel	Adelaide	4.5	1695	Free Wifi	Indoor Pool	Food Tour	Lake	No	Winter
The Chancellor on Currie	Adelaide	4	1116	Indoor Pool	Spa	Food Tour	Lake	No	Winter
Adina Apartment Hotel Adelaide Treasury	Adelaide	4.5	2860	Free Wifi	Spa	Food Tour	Lake	No	Winter

c. Build a ML model that can help a customer to select appropriate set of resorts based on the season of travel. Present and describe the performance of your model.

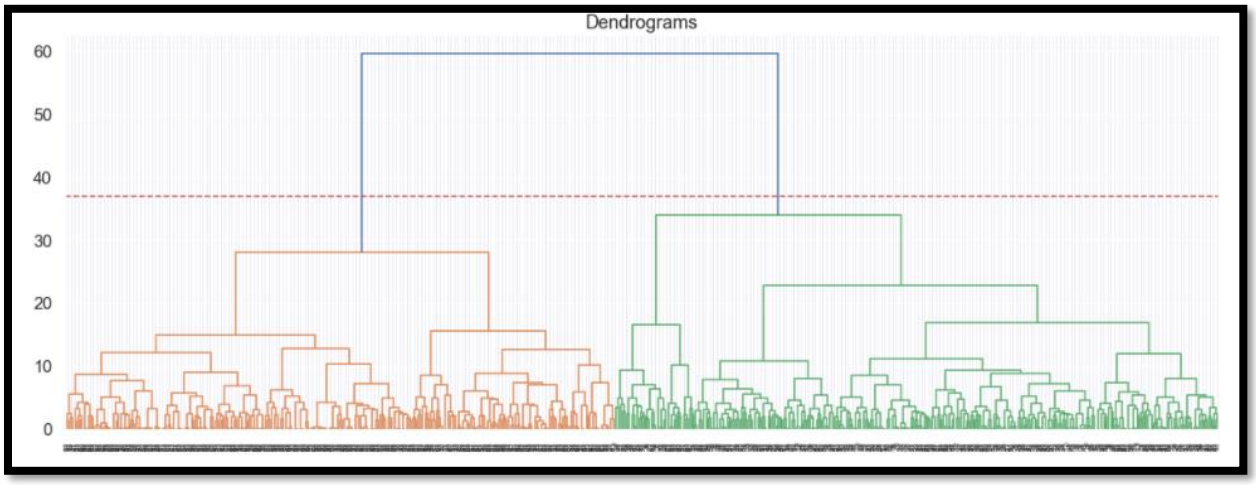
I have designed a model that provides a list of recommended hotels based on if the customer chooses Summer or Winter travel as their travel season option. Prior to building the model, I ensure that we have equal sample sizes across both seasons, so I took a sample of 400 records each from each season.

I then scaled the numerical data that is present in the dataset which was Ratings and Reviews count, and scaled the text variables using pandas get dummies to one hot encode these variables.

The algorithm that I have decided to use is agglomerative clustering, as like our previous task in question 4, I have 2 seasons to predict here, so I figured this algorithm would work well to identify two main clusters, which should relate to the two main seasons. I also chose this method as there is similarities between the types of activities and water access based on the season of travel, which I thought the algorithm should be able to derive patterns from and cluster accordingly.

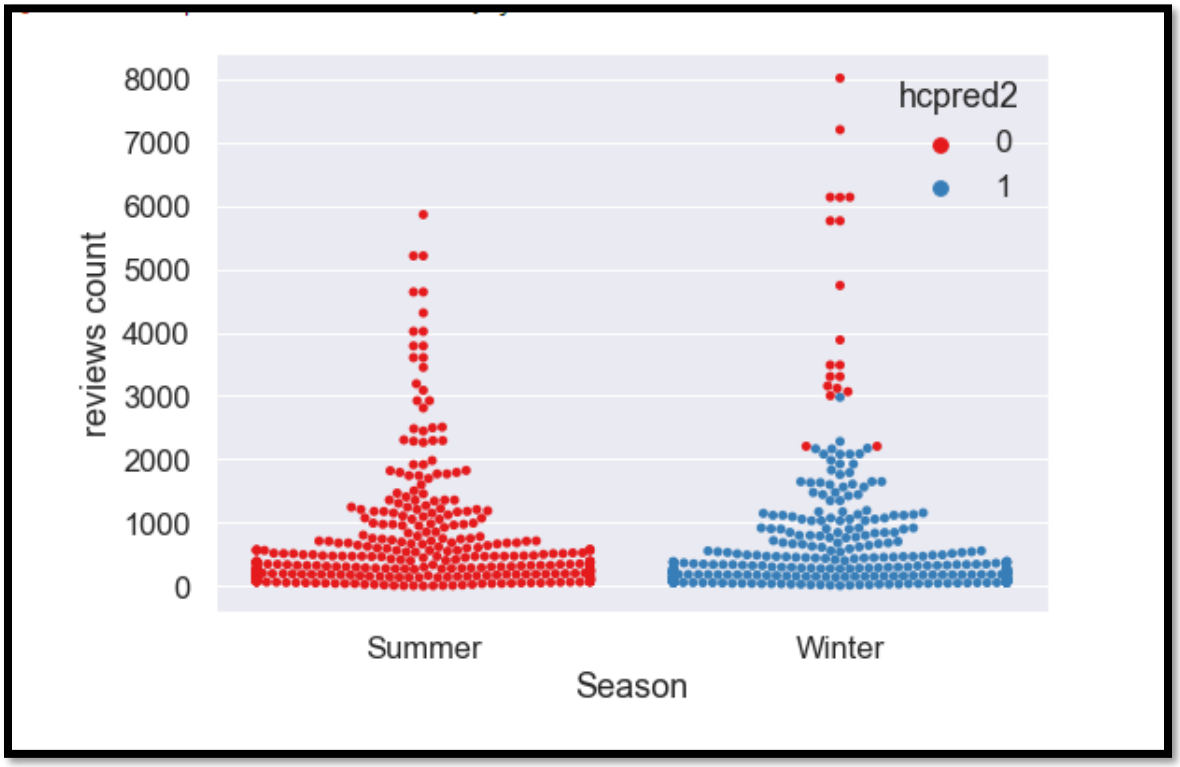
As you can see below, prior to passing the data set through the algorithm, I produced a dendrogram chart to visualise the distributions. There are 2 main clusters there that can be

observed.



I have used the Fowlkes Mallows score to evaluate the performance of the model. This scored a high 95%. And you can see the allocation of the predictions from the visualisation below, we can observe that all of the summer data points have been allocated a prediction of 0, which matches the class of 0 that we have assigned to summer, and most of the winter data points have been allocated a prediction of 1 which matches our classification of 1 for Winter, there was about 5% of winter data points that have been incorrectly classified as Summer.

Although even here, we need to take the classification with care, as unsupervised learning models is not designed to predict target labels. It's also important to note that further testing needs to be conducted with actual real life data set as I believe this model has performed well due to having a made-up dataset fitted into it.



d. Why do we need a ML model for this problem?

Given that unsupervised learning algorithm works by measuring distance in data points, and grouping together clusters that are similar in distance, I believe using unsupervised learning is appropriate to use when building a recommendation system such as the hotel recommendations that I have designed here. As most hotels and resorts, generally have similar features such as indoor or outdoor pools, spas or fireplaces in the rooms, these features once encoded should produce data points that are close to each other where the algorithm can then identify patterns in the data and group clusters accordingly. This model works like a recommendation system provided by Netflix for example, where it is recommending similar movies based on what has been watched and rated on previously.

References

Patlolla, C. R., 2018. *Understanding the concept of Hierarchical Clustering Technique.* [Online]
Available at: https://towardsdatascience.com/understanding-the-concept-of-hierarchical-clustering-technique-c6e8243758ec
[Accessed 30 August 2021].

Yiu, T., 2019. *The Curse of Dimesionality - Why High Dimensional Data Can Be So Troublesome.* [Online]
Available at: https://towardsdatascience.com/the-curse-of-dimensionality-50dc6e49aa1e
[Accessed 30 August 2021].