**IFN 509: DATA EXPLORATION AND MINING**

**ASSESSMENT 2**

**Team Name: SCOTT’S TOTS**

**Group No. 5**

| **Student Name** | **Student Id** |
| --- | --- |
| **Jiao Wu** | ***N9334173*** |
| **Koralalage Samindi Ranika Wanigasekera** | ***N11226757*** |
| **Vishnu Shaji** | ***N11407697*** |

***Team Contribution***

|  | **Jiao Wu** | **Koralalage Samindi Ranika Wanigasekera** | **Vishnu Shaji** |
| --- | --- | --- | --- |
| **Jiao Wu** | **100 %** | **100 %** | **100%** |
| **Koralalage Samindi Ranika Wanigasekera** | **100 %** | **100 %** | **100 %** |
| **Vishnu Shaji** | **100 %** | **100 %** | **100 %** |

**Question 1 - Association mining**

1. **What pre-processing was required on the dataset before building the association mining model? What variables did you include in the analysis? Justify your choice.**

Pre-processing - we corrected the data type for column ‘date’ from object to datetime64 in the case that sequence analysis is required for the following steps. We dropped the ‘global\_num’ variable because it had missing values and because we don’t need two separate identifying variables. Converted the data into a list of transactions. Took out all transactions(Figure 1.1) that only had one location in them, because this meant that that particular patient had not travelled anywhere from that one place.

Variables included -

* ‘patient\_id’ - needed in order to identify unique transactions and to track the routes of unique individual patients
* ‘location’ - this is the variable that shows which locations Covid-19 positive patients travelled to, since our task is to track patient routes this variable is crucial



Figure 1.1. Generating transactional data for association mining

1. **Conduct association mining and answer the following: a.**
2. **What ‘min\_support’ and `min\_confidence’ thresholds were set for this mining exercise? Rationalise why these values were chosen.**

min\_support = 1.0101%

min\_confidence = 22.1786%

The min\_support and min\_confidence thresholds set for this mining exercise were 1.0101% and 22.1786%. (Figure.1.2a) These values were chosen through trial and error. Min\_support = 0.05 and min\_confidence = 0.5 resulted in one rule, which was not enough. min\_support was reduced to 0.01 which resulted in 6 rules, which is more than earlier but still not enough for analysis purposes. Based on considerations of association patterns, we have excluded rules that have excessively high support than its value and involve excessive items caused by excessively low support. This has led us to the obtained result. These two values were fine-tuned until we reached the chosen values, which works perfectly for our analysis purposes of this assignment.

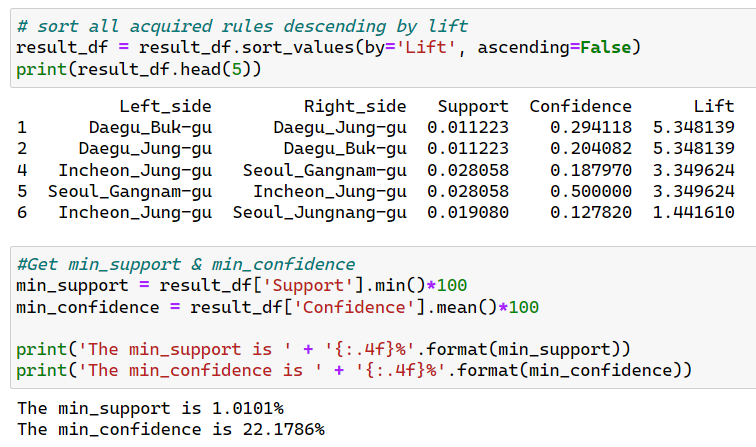


Figure 1.2a. Min\_support & min\_confidence

1. **. Report the top-5 (interesting) rules and interpret them.**

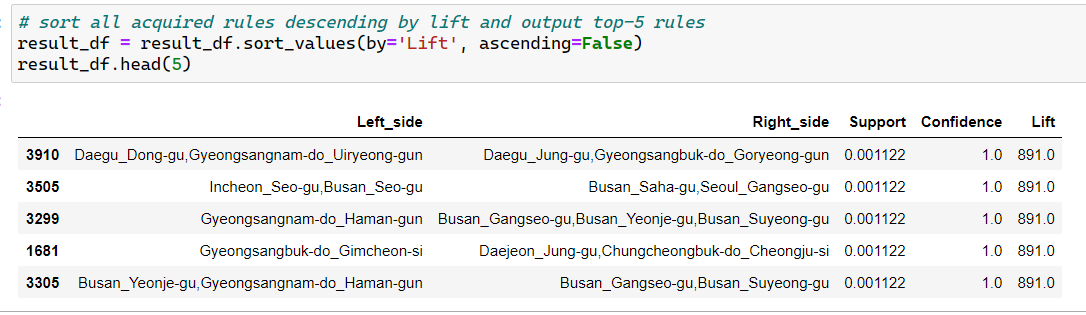


Figure 1.2b. Top5 rules

1). Daegu\_Dong-gu,Gyeongsangnam-do\_Uiryeong-gun ->> Daegu\_Jung-gu,Gyeongsangbuk-do\_Goryeong-gun

This rule has 100% confidence. A lift value of 891 and a support of 0.1122%. This rule suggests that positive patients who visited Daegu\_Dong-gu,Gyeongsangnam-do\_Uiryeong-gun also visited Daegu\_Jung-gu,Gyeongsangbuk-do\_Goryeong-gun

2). Incheon\_Seo-gu,Busan\_Seo-gu ->> Busan\_Saha-gu,Seoul\_Gangseo-gu

This rule has 100% confidence. A lift value of 891 and a support of 0.1122%. This rule suggests that positive patients who visited Incheon\_Seo-gu,Busan\_Seo-gu also visited Busan\_Saha-gu,Seoul\_Gangseo-gu

3). Gyeongsangnam-do\_Haman-gun ->> Busan\_Gangseo-gu,Busan\_Yeonje-gu,Busan\_Suyeong-gu

This rule has 100% confidence. A lift value of 891 and a support of 0.1122%. This rule suggests that positive patients who visited Gyeongsangnam-do\_Haman-gun also visited Busan\_Gangseo-gu,Busan\_Yeonje-gu,Busan\_Suyeong-gu

4). Gyeongsangbuk-do\_Gimcheon-si ->> Daejeon\_Jung-gu,Chungcheongbuk-do\_Cheongju-si

This rule has 100% confidence. A lift value of 891 and a support of 0.1122%. This rule suggests that positive patients who visited Gyeongsangbuk-do\_Gimcheon-si also visited Daejeon\_Jung-gu,Chungcheongbuk-do\_Cheongju-si

5). Busan\_Yeonje-gu,Gyeongsangnam-do\_Haman-gun ->> Busan\_Gangseo-gu,Busan\_Suyeong-gu

This rule has 100% confidence. A lift value of 891 and a support of 0.1122%. This rule suggests that positive patients who visited Busan\_Yeonje-gu,Gyeongsangnam-do\_Haman-gun also visited Busan\_Gangseo-gu,Busan\_Suyeong-gu

1. **List four most interesting routes taken by individuals who have tested positive for COVID19 and have travelled from Buk-gu City in Busan Province.**

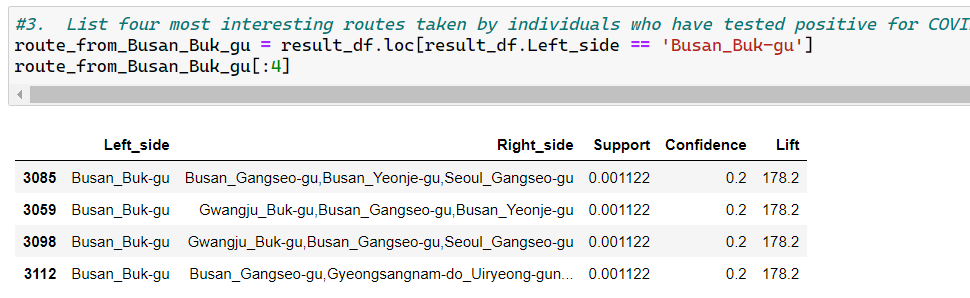


Figure 1.3. Top4 most interesting routes

1). Busan\_Buk-gu ->> Busan\_Gangseo-gu,Busan\_Yeonje-gu,Seoul\_Gangseo-gu

2). Busan\_Buk-gu ->> Gwangju\_Buk-gu,Busan\_Gangseo-gu,Busan\_Yeonje-gu

3). Busan\_Buk-gu ->> Gwangju\_Buk-gu,Busan\_Gangseo-gu,Seoul\_Gangseo-gu

4). Busan\_Buk-gu ->> Busan\_Gangseo-gu,Gyeongsangnam-do\_Uiryeong-gun..

1. **Can you perform sequence analysis on this dataset? If yes, present your results. If not, rationalise why.**

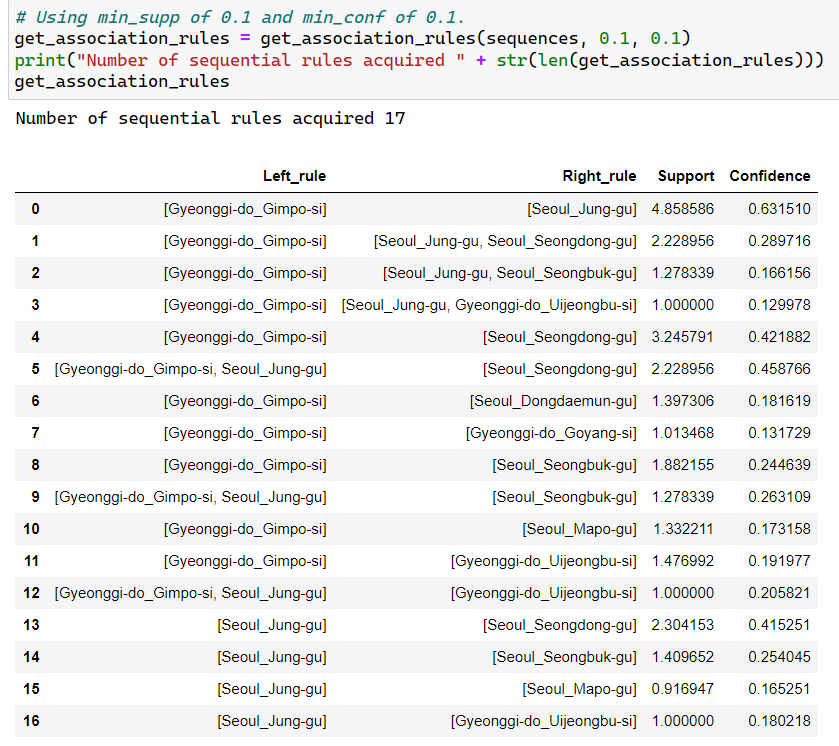


Figure 1.4. generating sequential rules

Yes, this dataset can undergo sequential analysis. We sorted the patient's actions by date and applied it to the location, resulting in the patient's travel route "trip". We then used SPMF to generate sequential rules. We set the min\_support threshold to 0.1 and the min\_confidence threshold to 0.1 to obtain association rules. We obtained a total of 17 sequential rules.

1. **In what ways can the results of this task be utilised by the relevant decision-makers?**

The information gained through analysing the dataset for the task can be used by relevant health institutions or any other institutions/departments responsible for tracking the growth of the Covid19 pandemic in this country. The results of this task clearly shows which cities/provinces were visited and from where; this can also be used to track patients if need be because if they know one location of the patient, it is likely that we already know where the patient might be travelling to next (at least a narrowed down list of provinces/cities they might travel to)

**Question 2 - Clustering**

1. **What pre-processing was required on the dataset (D2.csv) before building the clustering model and why?**

Had to create a separate data frame with just the numeric attributes in it, because that is what the task requires at this point. And had to also drop any rows that contained as Covid19 negative because the task considers only responses from Covid19 positive persons. We then dropped the attribute ‘covid19\_positive’ because it no longer brings value to the task.

1. **Build a clustering model to profile the characteristics of COVID positive individuals. Answer the followings:**
2. **What clustering algorithm have you used and why?**

K-Means clustering - K-means was chosen for this task on the basis of two considerations. First, we are currently only considering numeric values and no categorical values. It is common knowledge that K-Means work best for fully numeric datasets. The other reason is not having a certain attribute that we’re trying to predict. Instead, we’re using a selection of attributes and trying to find patterns and observations in the dataset.

1. **List the attributes used in this analysis.**

Height, weight, alcohol, contacts\_count, house\_count, worried

1. **What is the optimal number of clusters identified? How did you reach this optimal number?**

Optimal number of clusters identified (k) = 7

The elbow method and silhouette scores were both used when trying to decide the optimal number of clusters. The elbow method resulted in k being in the range of 2 - 8, and then the silhouette scores were used on all these values and this resulted in k = 7 being the highest silhouette score. The outcome of using the elbow method is shown in the image 1 below(Figure 2.2c).



Figure 2.2c. outcome

1. **Did you normalise/standardise the variables? What was its effect on the model – Does the variable normalisation/standardisation process enable a better clustering solution?**

Yes, the data was normalised. The normalisation was done because different attributes in the dataset lay in vastly different ranges (eg - the ‘height’ ranges anywhere from 100-240 while ‘worried’ ranges from 1-5). Yes, normalising did change the outcome of the task and gave better clusters. Without normalising the data, the optimal number of clusters was 2 (k=2) and the main division for the two clusters was based purely on the ‘weight’ (cluster 0 - lesser weights, cluster 1 - higher weights). But after normalising the data, the optimal number of clusters was 5 (k=5) which completely changed how the clusters turned out. Now patterns can be seen in not just the ‘weight’ attribute but also in attributes such as ‘alcohol’, ‘house\_count’ and ‘worried’.

**3.For the model with the optimal number of clusters, answer the following**

1. **Visualise the clusters using ‘pairplot’ and interpret the visualisation.**

Figure 2.3a. pairplot

This pairplot(Figure 2.3a) can be used to interpret the clustering results we have gained from this task. It can be used to identify certain patterns in the individual clusters.

When looking at the ‘weight’ attribute, it can be observed that subjects from Cluster 0 tend to have higher weights compared to the rest.

Observing the ‘house\_count’ and ‘contact-count’ attributes tells us that Cluster 4 tends to have higher numbers in both these attributes for most of the instances.

For the attribute ‘alcohol’ on the other hand, it is observed that Cluster 3 lies in the higher range much more than any of the other clusters.

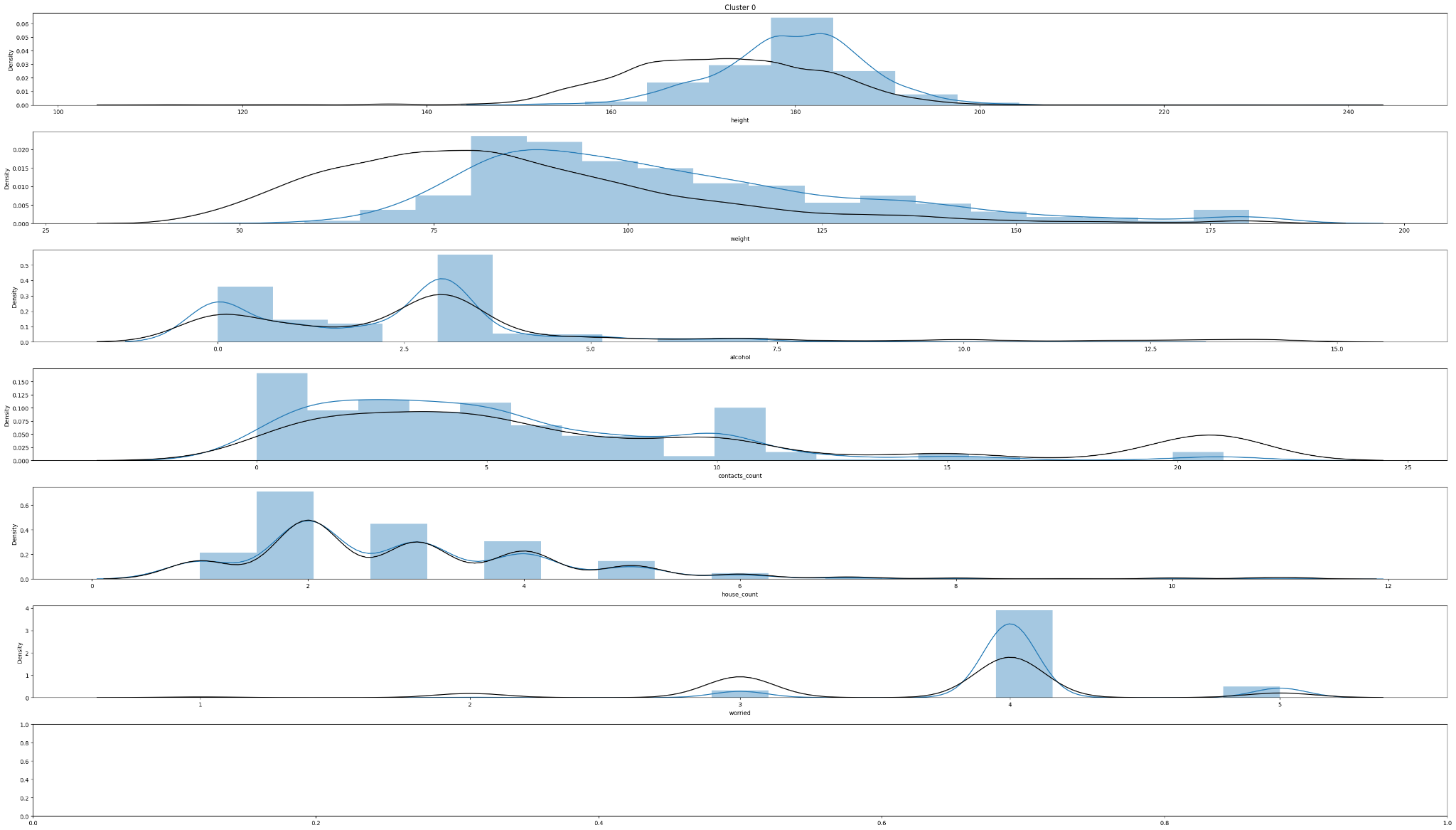
The attribute ‘height’ does not show any significant trends with the current clustering but it does have a few outlier points, that might be subjects on either end of the height spectrum, those who are either very much on the lower height range and very much on the higher range.

It can also be noticed that in most of the plots above, the intra-cluster distance seems to be less with most of the data points belonging to each cluster seems to be lying very close to each other and the visualising clearly shows this by the showing us distinct clusters (but there is overlapping of clusters in certain plots, making it unclear to us exactly which data points might be lying behind these clusters)

1. **Characterize the nature of each cluster by giving it a descriptive label and a brief description. Hint: use cluster distribution.**

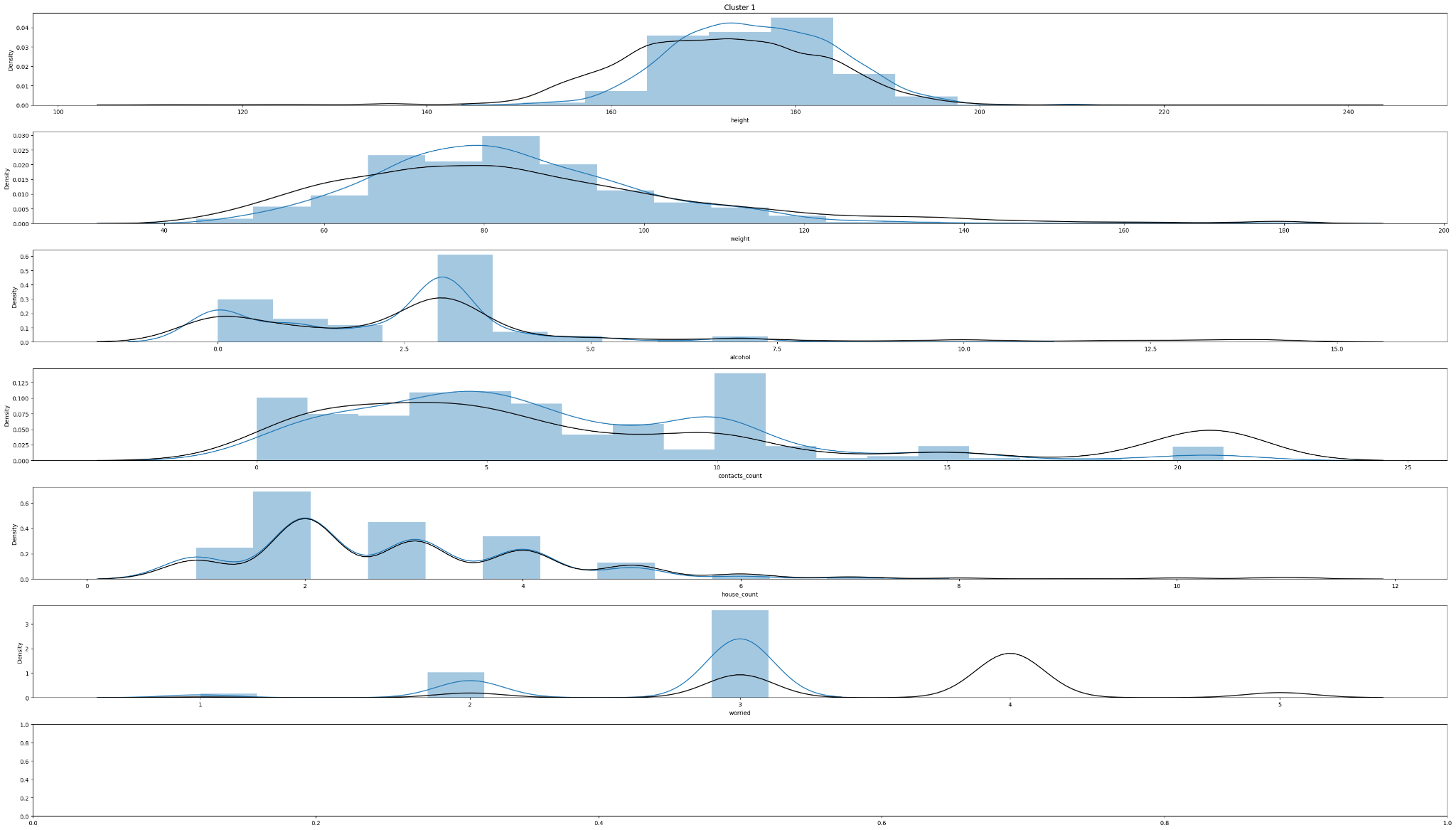
Blue line represents the distribution of the variable in the respective cluster

Black line represents the distribution of the variable in the whole dataset(Figure 2.3b)

Figure 2.3b1. pairplot

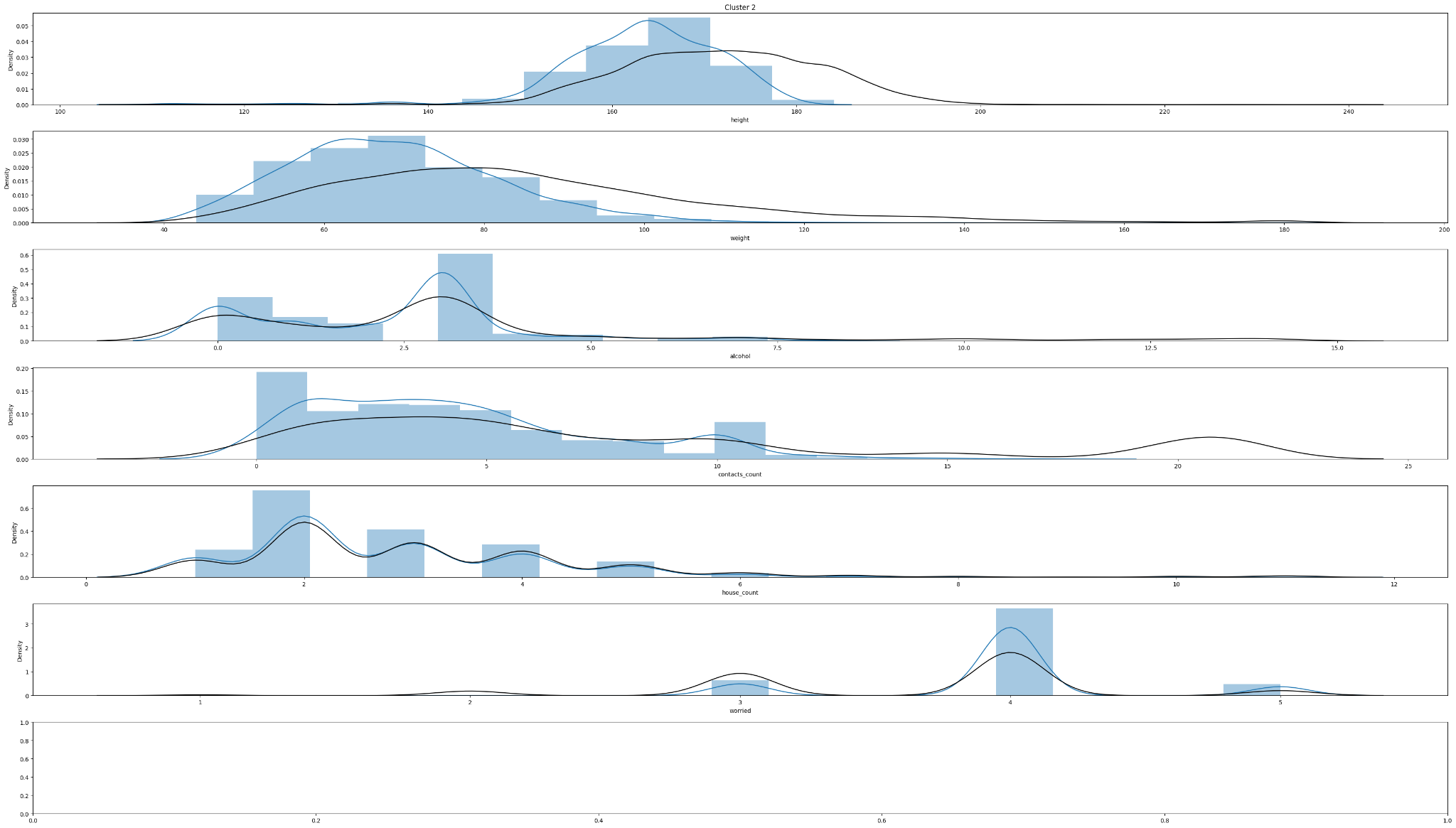
Cluster 0 - higher on the weight spectrum, slightly higher contact\_counts

The image above shows the distribution plots for Cluster 0. Cluster 0 seems to be following most of the trends in the dataset, but one noticeable fact is that for the attributes ‘height’ and ‘weight’, the cluster is showing a bit more right aligned trend than the rest of the data set. But in terms of the rest of the attributes, the data points in Cluster 0 are following the same trends as the rest of the data, but sometimes with higher values than the rest of the data (example - in ‘house\_count;, Cluster 0 has higher number in the 0-5 range, but follows almost the exact values in ‘contacts\_count’)

Figure 2.3b2. Pairplot

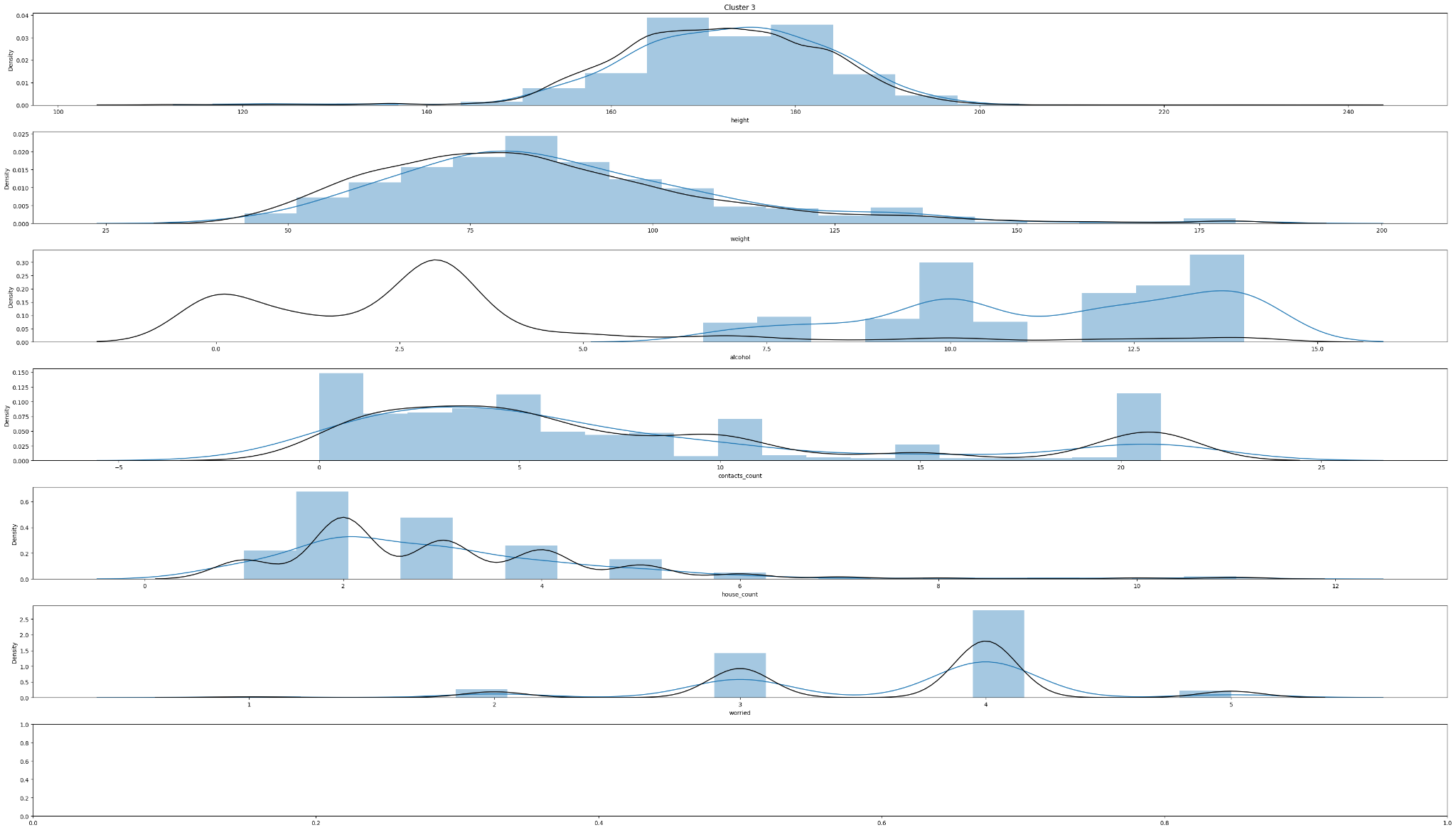
Cluster 1 - same house\_count trends as the rest of the data set, higher weight, height, contact\_count and low-mid levels of worry

The Cluster 1 follows most of the trends in the dataset for all variables, while following the exact trend for ‘house\_count’. It can also be seen that most of the subjects in Cluster 1 has mentioned that their level of ‘worried’ is 3, meaning average levels of worry (this still aligns with the trend, but higher count). The data set distribution shows subjects for ‘worried’ at 4 and 5 as well, but there’s none of that in Cluster 1, so we can come to the conclusion that subjects in Cluster 1 are low-mid worried but never highly worried about testing positive for Covid19.

Figure 2.3b3. Pairplot

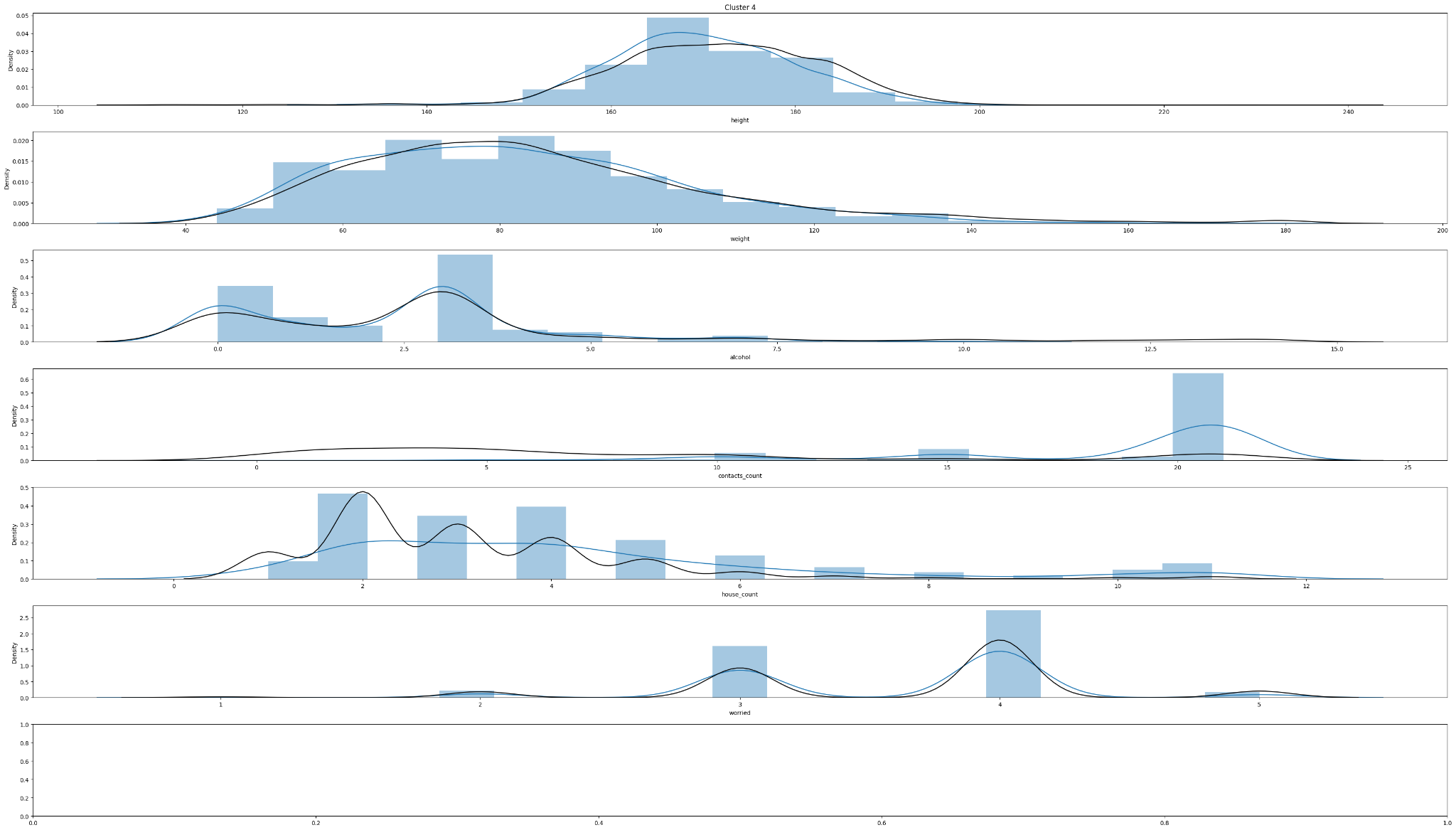
Cluster 2 - opposite trends in ‘weight’ and ‘height’ than the rest of the data

Most subjects in Cluster 2 show left trends in height and weight attributes as opposed to the right-ward trends shown in the overall dataset; this means that Cluster 2 has subjects with lower heights and weights than the average of the entire dataset for those attributes.

Figure 2.3b4. Pairplot

Cluster 3 - higher levels of alcohol, mid - high levels of worried

The data points in Cluster 3 follow similar trends to the general dataset in most attributes, except in ‘alcohol’. Cluster 3 follows an exact opposite trend in ‘alcohol’ - the average of the use of alcohol in the general dataset lies in the lower range of the spectrum, but the average of the use of alcohol in Cluster 3 lies on the higher range of the spectrum. It can also be observed that most data points in Cluster 3 have voted for ‘4’ for the ‘worried’ attribute (most of them on 3, 4, or 5, while still following the trends of the full data distributions), so we can conclude that the data points/subjects are mid-high levels of ‘worried’ about testing positive for Covid19.

Figure 2.3b5. Pairplot

Cluster 4 - low levels of alcohol, higher contact\_count and mid - high levels of ‘worried’

The average value Cluster 4 shows for ‘height’ is ever so slightly less than what the distribution of the entire dataset shows for ‘height’. It can also be observed that data points in Cluster 4 all lie on the 0-7.5 range in the ‘alcohol’ attribute, meaning almost all the data points in this cluster are in the lower 50% in terms of alcohol consumption.

**4. Now, build another clustering model by including the variable ‘Age’.**

**Use the best setting (e.g., variable standardisation, optimal K, etc) obtained in the previous models. Answer the followings:**

1. **What clustering algorithm have you used and why?**

K-Prototype - because the analysis now includes a categorical variable (‘age’). K-Prototypes algorithm is capable of handling numerical and categorical variables separately.

1. **List the attributes used in this analysis.**

Age, height, weight, alcohol, contacts\_count, house\_count, worried

1. **What difference do you see in this clustering interpretation when compared to the previous one (task 3)?**

Discuss the plot below on how it differs from the previous ones

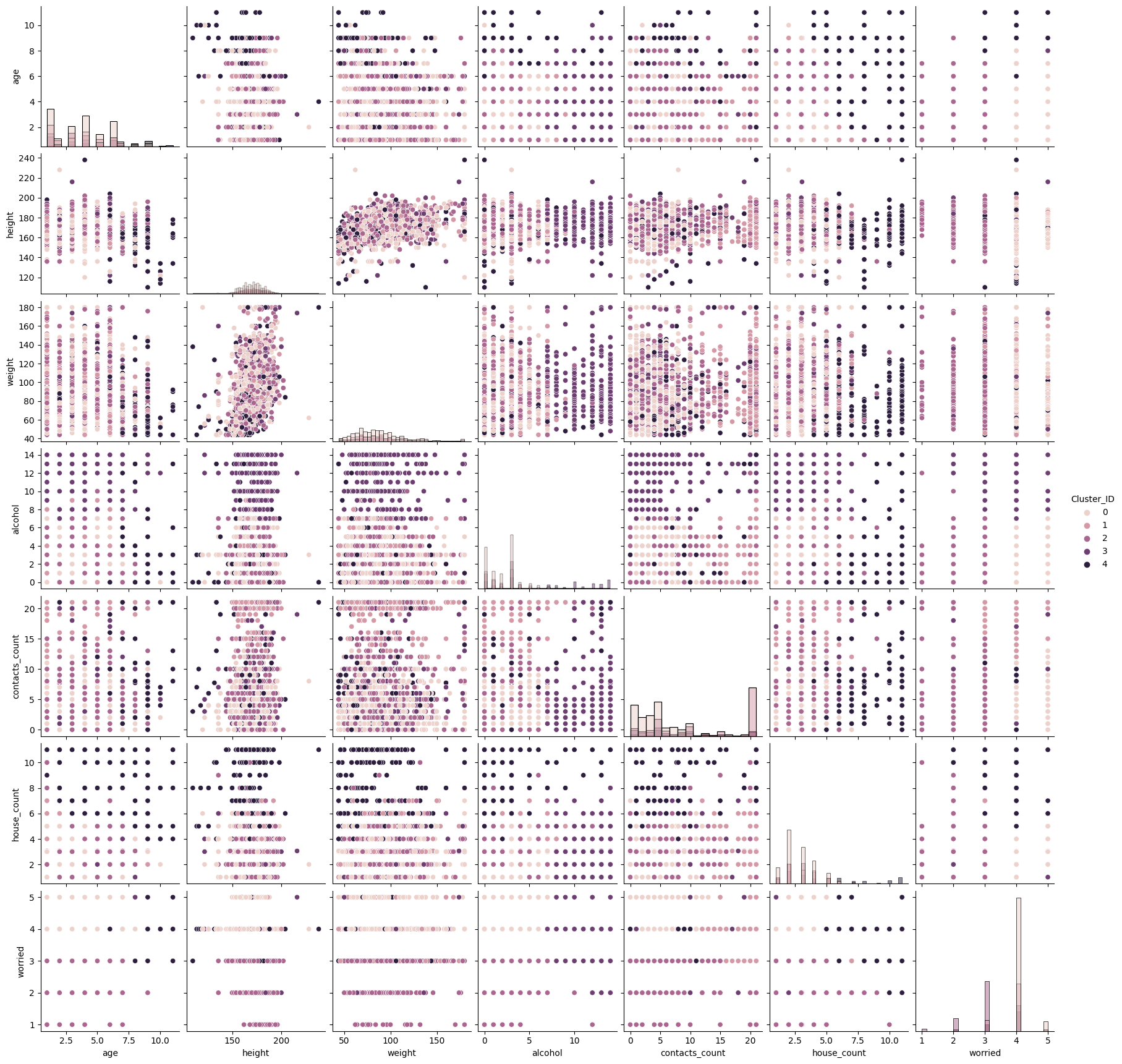


Figure 2.4. Pairplot

When looking at this second pairplot diagram after performing the KPrototype algorithm, it can be observed that certain plots in the above diagram have now changed from the previous question. Now, instead of Cluster 0 being the ones on the higher range of the ‘weight’ attribute, cluster 0 is now on the higher range of the ‘worried’ variable. Instead of Cluster 4 being on the higher ‘house\_count’s and ‘çontact\_count’s, it is now only higher on the ‘house\_count’ attribute. Instead of just cluster 3 being on the higher range of alcohol when performing K-Means, now both Cluster 2 and Cluster 3 are on the higher range of alcohol. From these interpretations it can be seen that the change in the input attributes and the change in the algorithm that is used, the performance and the interpretations of the clusters change.

5. These clustering results can be used to identify patterns in Covid19 positive patients and can be used to provide them with necessary treatments. As an example - if a certain cluster reports a very high number of contacts for its cluster data points, those patients and their contacts can be advised on the necessary steps to take in order to stop the virus from spreading further. Also if it is observed that a certain age percentile is more at risk of catching the virus, other people in those ages can be advised and be given better care as needed so that the virus can be either prevented or can be caught early on and be treated soon.

**Question 3 - part 1 - Decision Trees**

1. **What pre-processing was required on the dataset (D2.csv) before decision tree modelling? What distribution split between training and test datasets was used?**

The ‘contacts\_count’ and ‘house\_count’ variables were converted to int type to match the variable description that was provided. All numerical variables were checked to see if they contained any minus or invalid data; they were clean. The attributes were separated as X and Y, where X were all the input variables (all variables except the target variable) and Y was the target variable; in our case Y is ‘covid19\_positive’.

Distribution - 30% of the dataset were testing data and the other 70% of the dataset were training data. The data\_prep() function was used for the pre-processing steps. This function performs one hot encoding on the data, divides the data into X and Y and then uses the train\_test\_split() function from SKLeanr to split the data into train and test datasets. The split used was 30% test data and 70% train data.

1. **Build a decision tree using the default setting. Answer the followings:**
2. **What is the classification accuracy of training and test datasets?Training accuracy -**

1.0

Test accuracy - 0.66

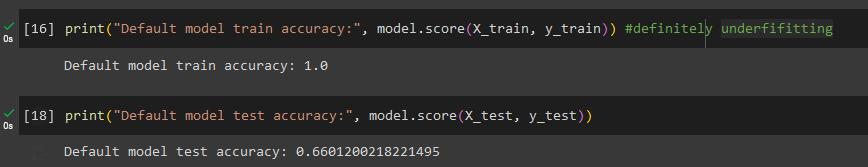


Figure 3.2a. Training accuracy & test accuracy

1. **What is the size of the tree (number of nodes and rules)?**

Number of nodes - 2009

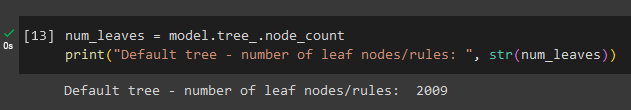


Figure 3.2b1. Get info from saved tree

Number of rules - 1005

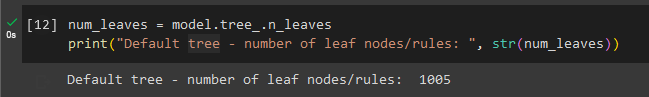


Figure 3.2b2. Get info from saved tree

1. **Which variable is used for the first split?**

Income\_high (<= 0.5)

1. **What are the 5 important variables (in the order) in building the tree?**

These 5 most important variables were found by using the ‘absolute()’ function in the Numpy package on the ‘feature\_importances’ in the default Decision tree model. Had to use this function to make sure that the absolute values were being used and that even the negative values were being considered alike.

* Weight
* Height
* Income\_high
* Contacts\_count
* House\_count

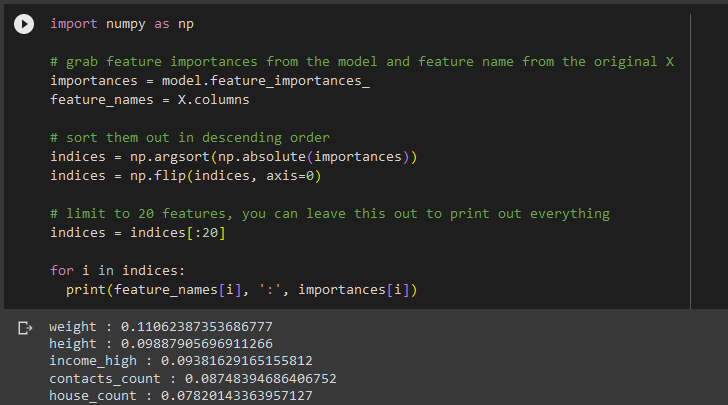


Figure 3.2d. 5 variables

1. **What parameters have been used in building the tree? Detail them.**

Class\_weight - none, criterion - gini, max\_depth - none, max\_features - none, max\_leaf\_nodes - none, min\_samples\_leaf - 1, min\_sample\_split - 2

3. **Build another decision tree tuned with GridSearchCV. Answer the followings:**

1. **What is the classification accuracy of training and test datasets?**

Training accuracy - 0.74

Testing accuracy - 0.70

1. **What is the size of the tree (i.e. the number of nodes and rules)?**

Number of nodes - 187

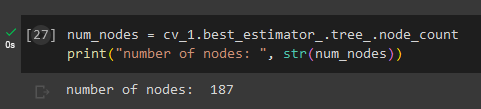


Figure 3.3b1. Number of nodes

Number of rules - 94

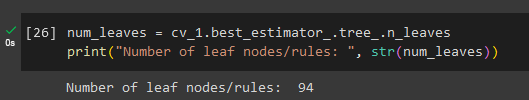


Figure 3.3b2. Number of rules

1. **Which variable is used for the first split?**

Income\_high (<= 0.5)

1. **What are the 5 important variables (in the order) in building the tree?**

Income\_high

Worried

Weight

Contacts\_count

Height

1. **Report if you see any evidence of model overfitting**

No - no significant signs of overfitting. There is only a 0.4% difference between the training accuracy and the testing accuracy. This difference is a very small amount that can be disregarded without considering it to be a sign of overfitting.

**4. What differences do you observe between these two decision tree models (with and without fine-tuning)? How do they compare performance-wise? Produce the ROC curve for both DTs. Explain why those changes may have happened.**

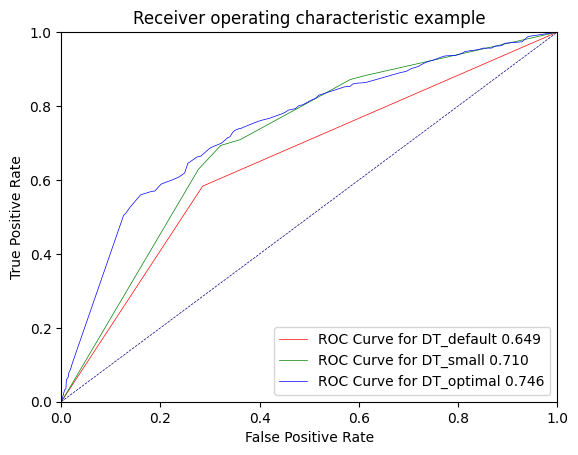


Figure 4. Receiver operating characteristic examples

The curve with the most area under the curve in a ROC graph is the best performing model out of the ones being compared. The diagonal line on the ROC curve is where the area under the curve is at 50% and it represents models that are no better than random chance, anything above that can be a good performing model. In our case, even the default model performs well but the optimal model (result of GridSearchCV) is the one with the highest are under the curve, so that is the better performing model.

These changes might have happened due to fine-tuning the relevant parameters/ hyperparameters in order to get the best performing model through GridSearchCV

**5. Using the better model, can you identify which individuals could potentially be "COVID positive"? Can you provide the general characteristics of those individuals?**

Utilising these results to identify Covid19 positive patients can be done in two ways. One is to use the feature\_importances of the better performing model (in our case, the model from GridSearchCV).

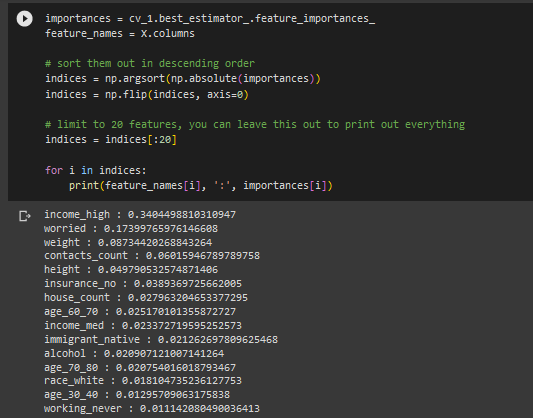


Figure 5.1. GridSearchCV

When looking at the above screenshot of the feature\_importances, it can be observed that data points with higher income, higher contact count, higher count of people in the household, and in the ages of 60-70 are more likely to test positive for the virus.

The second method of using the results to identify positive patients is to utilise the resultant decision tree itself. The below snippet of the optimal decision tree found through GridSearchCV will explain how to do this –

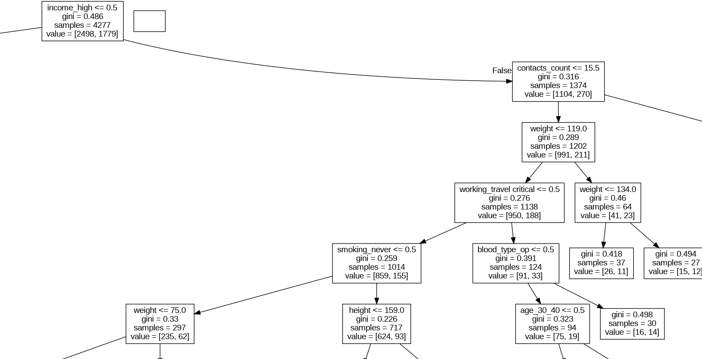


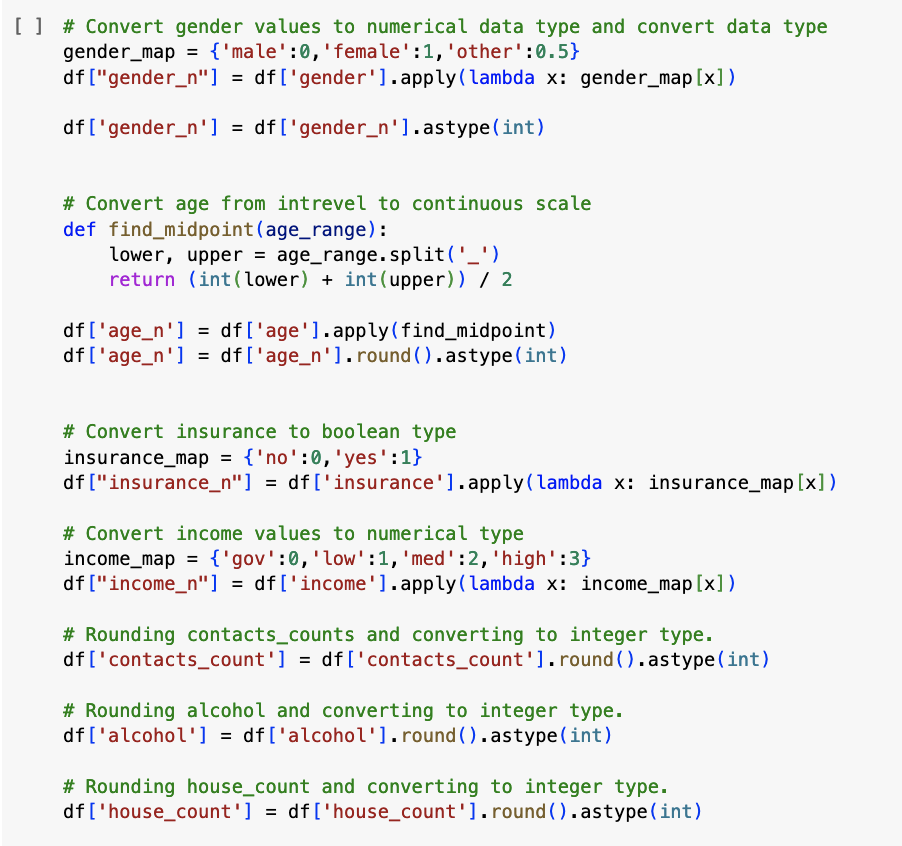
Figure 5.2. Decision tree

Income\_high <= 0.5 (False) -> contact\_count <= 15.5 (True) -> weight <= 119 (True) -> working\_travel critical <= 0.5 (False) -> blood\_type\_op <= 0.5 (False)

If you follow the sequence of rules that is mentioned you will get to 27 samples that have been reported as COvid19 positive by the decision tree.

**Predictive Modelling using Regression**

1. **What pre-processing was required on the dataset before regression modelling? What distribution split between training and test datasets was used?**

The pre-processing done this dataset was slightly different than what's done with other 2 models, while other models used one-hot encoded variables for the features, ordinal features in this model were either label encoded with LabelEncode method or mapped to numerical values then scaled with Standard Scaler.   
Figure 3.2.1 Preprocessing the data for regression

Some additional preprocessing was also done to ensure data integrity, they are as follows:

1. Dropping blood\_type because a significant amount (33%) of data is missing and hence, it doesn't provide any value to the analysis.
2. One Hot Encode Nominal variables such as race, immigrant, smoking, working
3. Scale the numerical features appropriately.



Figure 3.2.2 Additional Preprocessing (One-hot-encoding)

**2. Build a regression model using the default regression method with all inputs. Build another regression model tuned with GridSearchCV. Now, choose a better model to answer the followings:**

1. **Explain why you chose that model.**

The results indicate that the train accuracy of both models was comparable, suggesting that they performed similarly in predicting the target variable. However, when examining the test accuracy, the default model, model, demonstrated slightly superior performance compared to the GridSearchCV-tuned model, cv.

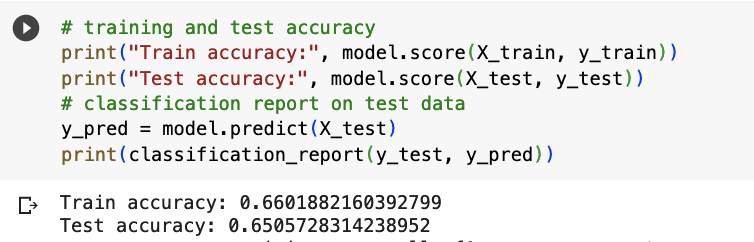
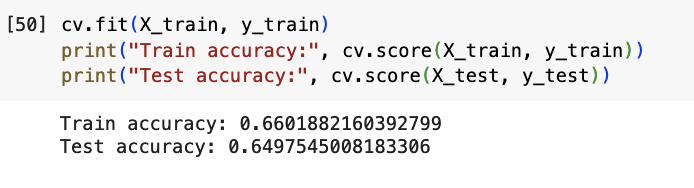


Figure 3.2.3 Test and Train accuracy of 2 models

1. **Name the regression function used.**

The objective of this analysis is to use logistic regression for classifying patients as either positive or negative for COVID-19 based on the binary variable "covid19\_positive," where 1 represents a positive diagnosis and 0 represents a negative diagnosis.

1. **Did you apply standardisation of variables? Why would you standardise the variables for regression mining?**

Yes, standardisation of variables was applied. The input variables were standardised for several reasons. Firstly, the variables had different scales, with height, weight, contact count, house count, alcohol and ‘worried’ having larger scales compared to the binary variables (0, 1).

Secondly, prediction algorithms tend to perform better with smaller scales as it improves their speed and efficiency. Lastly, standardisation ensures that data points can be easily compared by placing them on the same scale, which is essential for finding relationships and predicting new values.

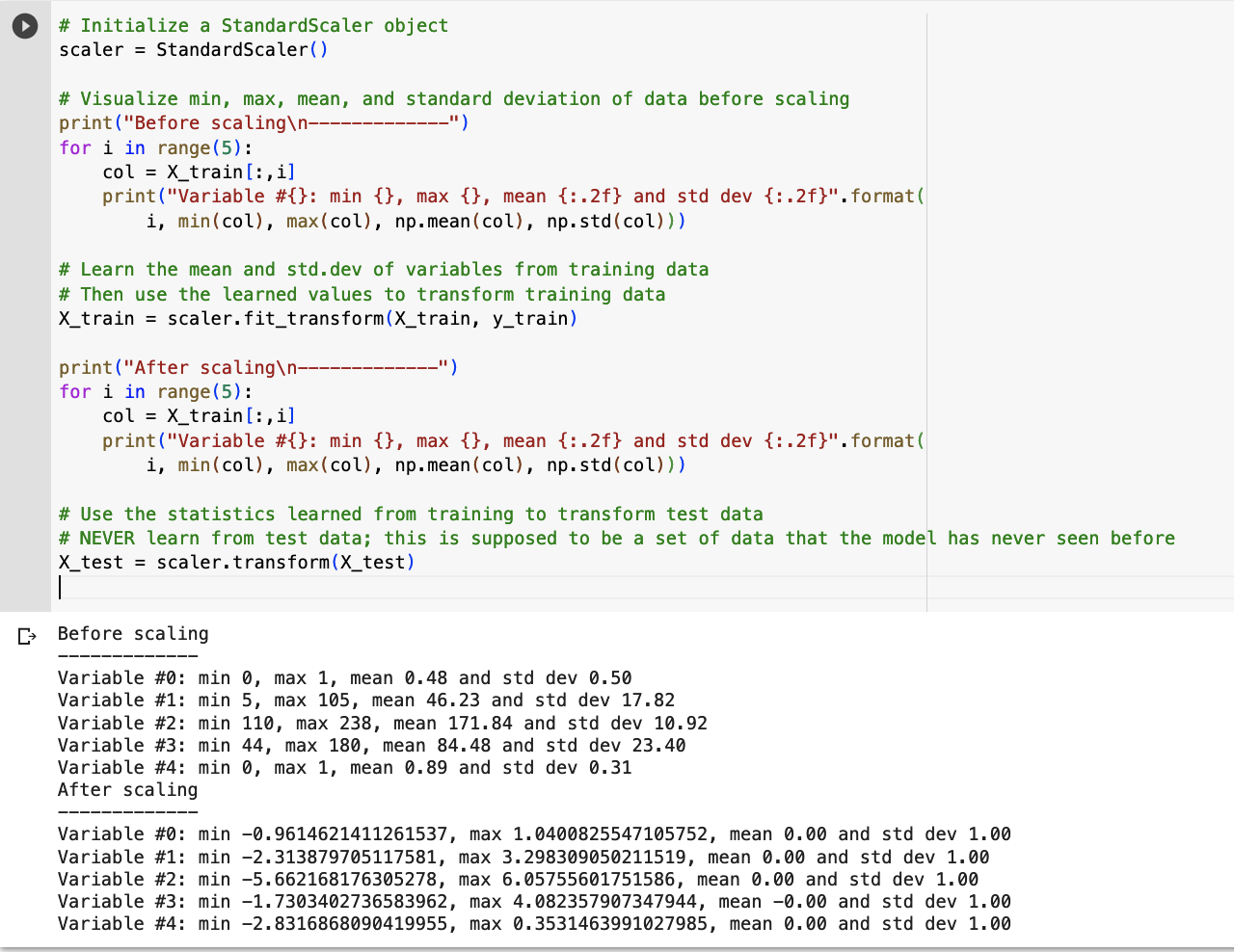


Figure 3.2.4 Scaling the features

1. **Report the variables included in the regression mode**

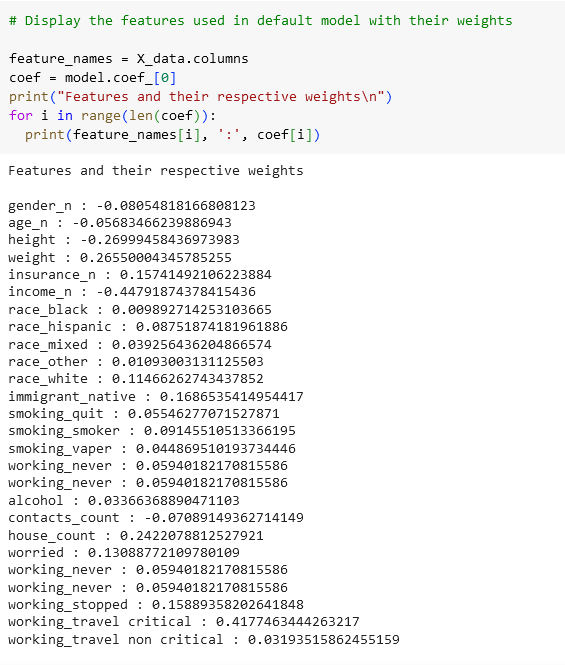


Figure 3.2.5 Features used in default model

The input variables used in the default model are: height, weight, insurance, income, race (black, Hispanic, mixed, other, white), immigrant (native), smoking (quit, smoker, vaper), working (never, stopped, travel critical, travel non-critical), alcohol, contacts count, house count, and worry.

Some of these variables were transformed into binary dummies during the preprocessing stage, resulting in separate variables for each nominal category, such as smoking, race, working and so on.

1. **Report the top-5 important variables (in the order) in the model.**

The feature importance indicates the impact of each variable in the model. According to Figure , the 'income\_n' variable had the greatest impact in the default model (model).

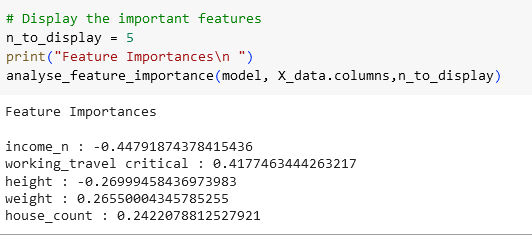


Figure 3.2.6 : Feature importance of the default model

1. **What is the classification accuracy on training and test datasets?**

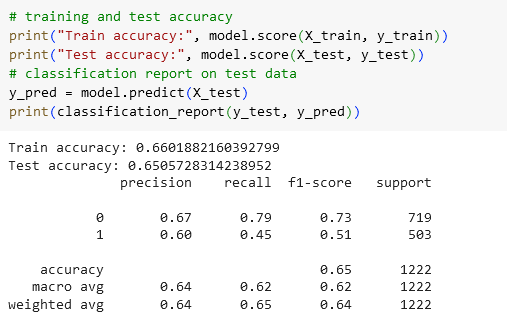


Figure 3.2.7 : Accuracy on training and test datasets using the default model

1. **Report any sign of overfitting in this model.**

Based on the provided data, the training accuracy (0.660) is slightly higher than the test accuracy (0.650). This suggests that the model may have some degree of overfitting.

To confirm if overfitting is occurring, it would be helpful to compare the model's performance on additional evaluation metrics and explore its learning curve.

**3. Build another regression model on the reduced variables set. Perform dimensionality**

**reduction with Recursive feature elimination. Tune the model with GridSearchCV to**

**find the best parameter setting. Answer the followings:**

1. **Was dimensionality reduction useful to identify a good feature set for building an accurate model?**

After applying Recursive Feature Elimination with Cross-Validation (RFECV) and tuning the model using GridSearchCV, we successfully reduced the number of features from 26 to 17. This feature elimination process aimed to select the most relevant features that significantly impact the model's performance.

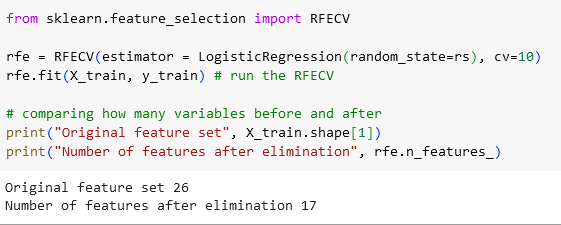


Figure 3.2.8: Eliminating least important features using RFECV

1. **What is the classification accuracy on training and test datasets?**

The following figure shows the accuracy of the RFECV mode (rfe\_cv).

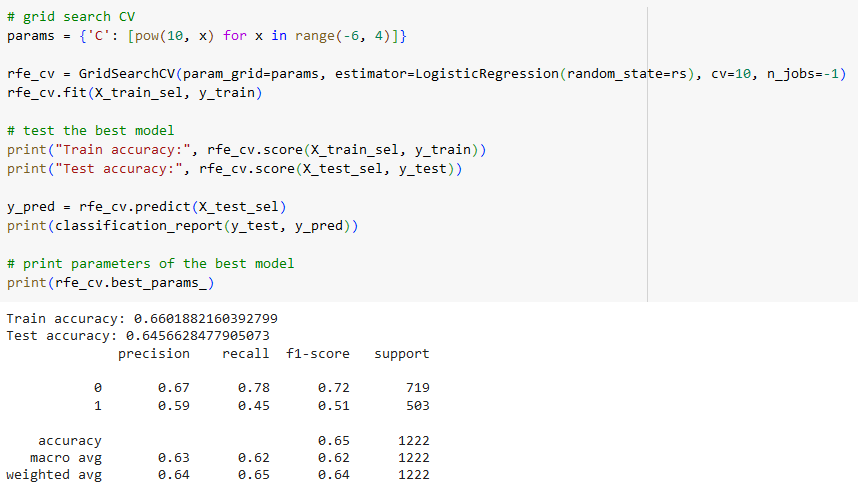


Figure 3.2.9: Train and test accuracy of RFECV

1. **Report any sign of overfitting.**

The training accuracy of 0.660 suggests that the model was able to correctly classify or predict the target variable for approximately 66% of the instances in the training dataset. However, the test accuracy of 0.645 indicates that the model's performance slightly dropped when applied to new, unseen data. This suggests that the model might be slightly overfitting to the training data and not generalising well to new observations.

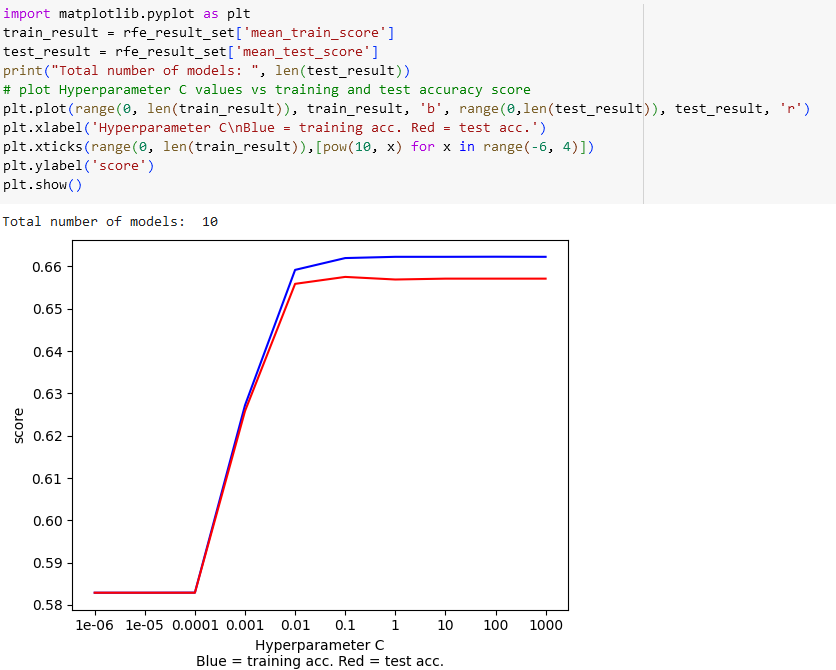


Figure 3.2.10 Performance graph of RFECV model

To assess if overfitting was present, an additional test was conducted. The results revealed a high accuracy for both the training and test datasets. The figure displayed a noticeable similarity between the accuracies of the two datasets. Furthermore, the regularisation hyperparameter C was set to 0.1, indicating that the model demonstrated stability beyond this value. As a result, there is no indication of overfitting in the model.

1. **Report the top-3 important variables (in the order) in the model.**

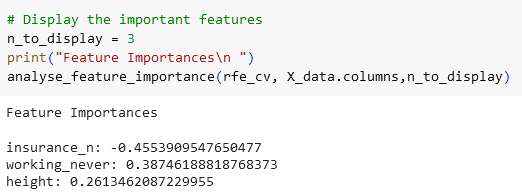


Figure 3.2.11 Top3 feature of the model RFE and tuned with GridSearchCV

The feature with the highest absolute value of importance is "insurance\_n," indicating that it plays a significant role in predicting the target variable. On the other hand, "working\_never" also has a relatively high absolute importance, suggesting its considerable influence. Finally, "height" has a lower absolute importance compared to the other features, indicating a relatively weaker relationship with the target variable.

**4. Produce the ROC curve for all different regression models. Using the best regression**

**model, can you identify which individuals could potentially be "COVID positive"?**

**Can you provide the general characteristics of those individuals?**

Three models were constructed in total:

1. The default logistic regression model (model).
2. The logistic regression model with grid search (cv).
3. The logistic regression model with feature selection using RFE and grid search (rfe\_cv).

The ROC curve, depicted in Figure c2-15, demonstrates that the regression model tuned with GridSearchCV (rfe\_cvv) exhibited slightly higher ROC performance.

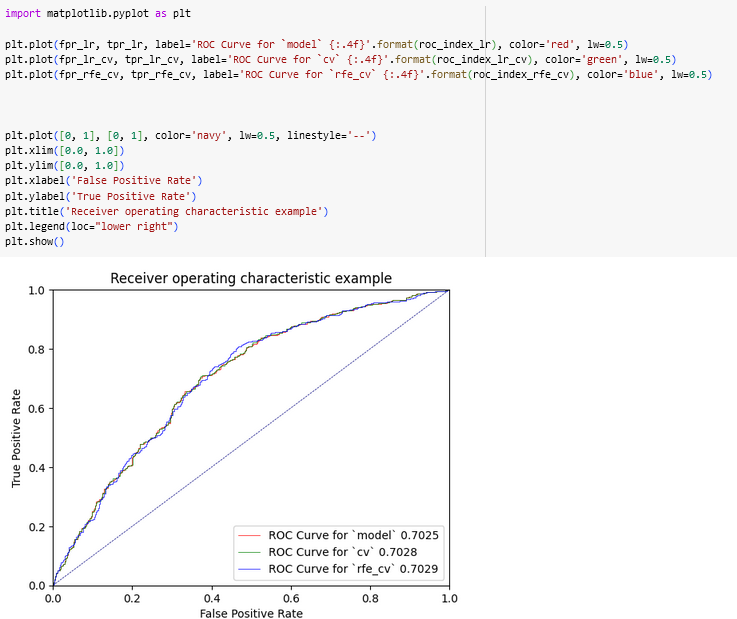


Figure 3.2.12 ROC curve of all regression models

We can identify potential patients based on the provided feature importances.Certain characteristics are more influential in identifying individuals who could potentially be "COVID positive." The negative impact of the feature "insurance\_n" suggests that individuals without insurance are more likely to be identified as potentially "COVID positive." Additionally, the positive influences of features such as "working\_never," "height," "smoking\_quit," and "race\_hispanic" suggest that individuals who are not currently working, have greater height, have quit smoking, or belong to the Hispanic race may be more likely to be identified as potentially "COVID positive." However, it's important to note that these feature importances provide only a general understanding and further analysis and evaluation would be needed for accurate identification.

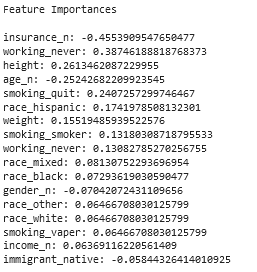


Figure 3.2.13 Importances of 20 features in RFECV model

**Question 3 - part 3 - Neural Networks**

1. **What pre-processing was required on the dataset before neural network modelling? What distribution split between training and test datasets was used?**

The ‘contacts\_count’ and ‘house\_count’ variables were converted to int type to match the variable description that was provided. All variables were checked for missing values or NaN’s, but they were clean. The dataset was divided into X (input variables) and Y (the target variable), in our case the Y was ‘covid19\_positive’ and everything else was X. Then the X and Y sets were divided into train data and test data using the data\_prep() function. The data\_prep() function performs one hot encoding on the data, divides the data into X and Y and then uses the train\_test\_split() function from SKLeanr to split the data into train and test datasets. The split used was 30% test data and 70% train data.

The data was then normalised using the StandardScaler() class. This was done because the attributes in the dataset were of varying ranges.

1. **Build a Neural Network model using the default setting. Answer the following:**
2. **Explain the parameters used in building this model, e.g., network architecture, iterations, activation function, etc.**

Activation - relu

Number of iterations - 200

Number of layer - 3

Number of features seen during ‘fit’ - 56

1. **What is the classification accuracy on training and test datasets?**

Training accuracy - 0.94

Test accuracy - 0.69

1. **Did the training process converge and result in the best model?**

No, it did not. The algorithm ended with a convergence warning where it warned that the maximum number of iterations for the default model (200 iterations) was reached before the model could reach convergence. ‘Max\_iter’ parameter in the algorithm should be raised to avoid this issue.

**3.Refine this network by tuning it with GridSearchCV. Report the trained model.**

1. **Explain the parameters used in building this model, e.g., network architecture, iterations, activation function, etc.**

Activation - relu

Number of iterations - 700

Number of layer - 4

Alpha - 0.0001

Number of features seen during ‘fit’ - 56

1. **What is the classification accuracy on training and test datasets?**

Training accuracy - 0.73

Test accuracy - 0.71

1. **Did the training process converge and result in the best model?**

On the default max\_iter (200 iterations) it did not converge (alpha = 0.1, hidden\_layer\_size = 3). So the max\_iter had to be manually increased, then the model converged and the best parameters and network architecture for this fully converged model changed (alpha = 0.0001, hidden\_layer\_size = 4).

1. **Do you see any sign of over-fitting?**

No significant overfitting was observed. While the performance of the model is around the range of 70% accuracy, this level is common in both the training accuracy as well as test accuracy (resulting in similar accuracies for both the sets)

**4. Let us see if feature selection helps in improving the model. Build another Neural Network model with a reduced feature set. Perform dimensionality reduction by selecting variables with a decision tree (use the best decision tree model that you have built in the previous modelling task). Tune the model with GridSearchCV to find the best parameter setting. Answer the followings:**

1. **Did feature selection favour the outcome? Any change in network architecture? What inputs are being used as the network input?**

Feature selection did improve the results, but on a very minor scale; it increased the test accuracy by around 1%, reducing whatever the little over-fittingness that was seen before. Yes, the network architecture did change, now with 9 hidden layers.

Inputs used - income\_high, worried, weight, contacts\_count, height, house\_count, insurance\_no, income\_med, race\_white, age\_60\_70, immigrant\_immigrant, alcohol, age\_70\_80

1. **What is the classification accuracy of training and test datasets?**

Training accuracy - 0.73

Test accuracy - 0.72

1. **How many iterations are now needed to train this network?**

400 iterations were needed to be able to complete executing and finding the final model successfully.

1. **Do you see any sign of over-fitting? Did the training process converge and result in the best model?**

No major signs of overfitting, because the training accuracy is very similar to the value of the test accuracy. Yes, the max\_iter hyperparameter of the MLP Classifier needed to be fine tuned to find the correct number of iterations needed to reach convergence and result in the best model.

**5. Produce the ROC curve for all different NNs. Now, using the best neural network model, can you provide characteristics of the individuals identified as COVID positive by the model? If it is difficult (or even infeasible) to comprehend, discuss why.**

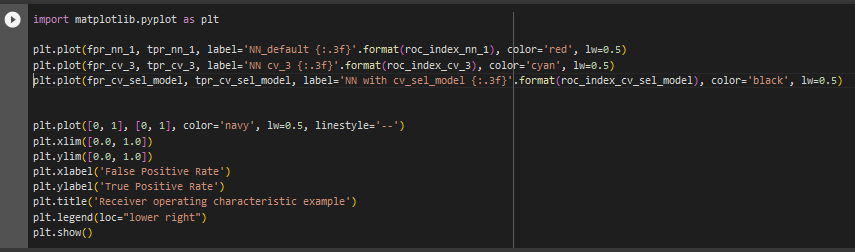


Figure 3.3.5.1 ROC graph

The above code snippet resulted in the below ROC graphs for the three neural networks trained for this task. ‘NN\_default’ is the neural network trained under default hyperparameter values of the MLP Classifier, ‘NN cv\_3’ is the final GridSearchCV fine-tuned model, and ‘NN with cv\_sel\_model’ is the model trained after feature selection using the optimal decision tree trained in a previous task.

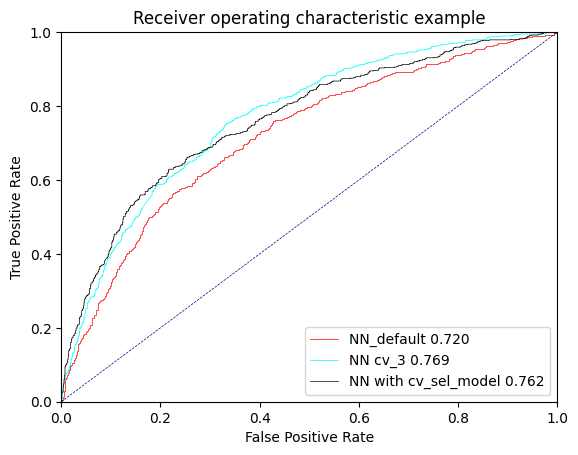


Figure 3.3.5.2 ROC graph

Looking at the three ROC curves for the three neural networks, it can be seen that the neural network that had its hyperparameters fine-tuned using GridSearchCV performed slightly better (0.7%) than the feature selected (through the decision tree) model and 4% better than the default neural network.

While it is understood that identifying the characteristics of Covid19 positive patients is a key feature for this task, it does come with certain limitations. However, the below code can be used to get the respective output that gives us an idea on the weights of the most important features within this best performing model.



Figure 3.3.5.3 GridSearchCV

**Final remarks - decision making**

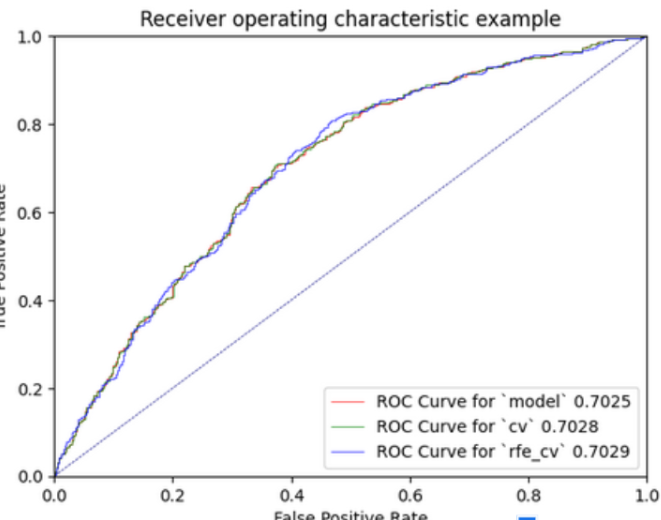
1. **Finally, based on all models and analysis, is there a model you will use in decision making? Justify your choice. Draw a ROC chart and accuracy table to support your findings.**

Choosing the best model comes after considering various different factors - these factors can be training speed, performance accuracy, and more and this can also be subjected to limitations due to limited machine capabilities that we might have to face. However due to the sensitivity of correctly identifying a Covid19 patient we need to be as sure as possible under our circumstances. Because a wrong diagnosis can lead to either a patient receiving unnecessary treatment leading to unavoidable repercussions or not getting the treatment that is needed which can lead to long term Covid19 or even death. So, we need to focus a lot on the accuracy of our models and the ROC curves.

The table below contains the accuracy levels of each different type of algorithm and the different models for each that we have trained for this task.

| Algorithm | Model | Model name | Train accuracy | Test accuracy | ROC |
| --- | --- | --- | --- | --- | --- |
| Decision tree | Default | DT\_default | 1.0 | 0.66 | 0.65 |
| Fine-tuned | DT\_optimal | 0.74 | 0.70 | 0.75 |
| Logistic regression | Default | model | 0.66 | 0.64 | 0.7025 |
| Fine-tuned | cv | 0.65 | 0.64 | 0.7028 |
| Featured reduction (RFE) | rfe\_cv | 0.66 | 0.65 | 0.7029 |
| Neural network | Default | NN\_default | 0.94 | 0.69 | 0.72 |
| Fine-tuned | NN cv\_3 | 0.73 | 0.71 | 0.77 |
| Feature reduction (using decision tree) | NN with cv\_sel\_model | 0.73 | 0.72 | 0.76 |

The best model of each of the algorithms have been highlighted in blue colour in the above graph. According to the ROC values of each of these models, the fine-tuned neural network is the best performing model. The fine-tuning for this model was done using SKLearn’s GridSearchCV() function. According to the accuracy values, the fine-tuned (GridSearchCV) decision tree and the fine-tuned neural network have very similar accuracies. While the fine-tuned decision tree does have better accuracy on the training set, it also has a 4% difference between train accuracy and test accuracy, while this difference in the fine-tuned neural network is only 2%. While it is observed that this difference is very small, due to even the slightest chance of overfitting, the fine-tuned neural network is the best model even in terms of accuracy. So, we can come to the sound conclusion that the neural network that was fine-tuned using GridSearchCV is the best model out of all for this task of properly classifying patients as Covid19 positive or negative.

The below ROC graphs can be referenced when considering the above tables. These ROC graphs were used to get the ROC values that are being considered in the model selection that is done above.

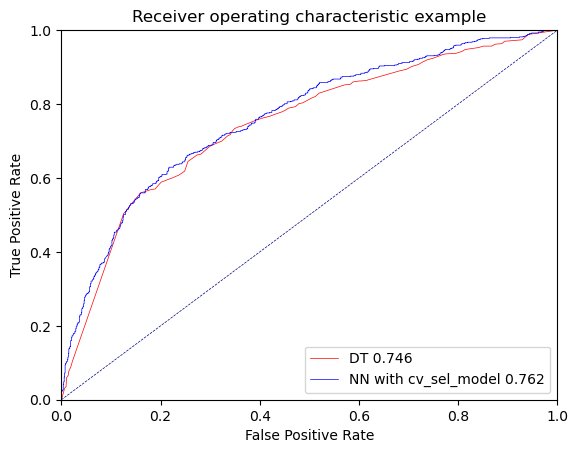
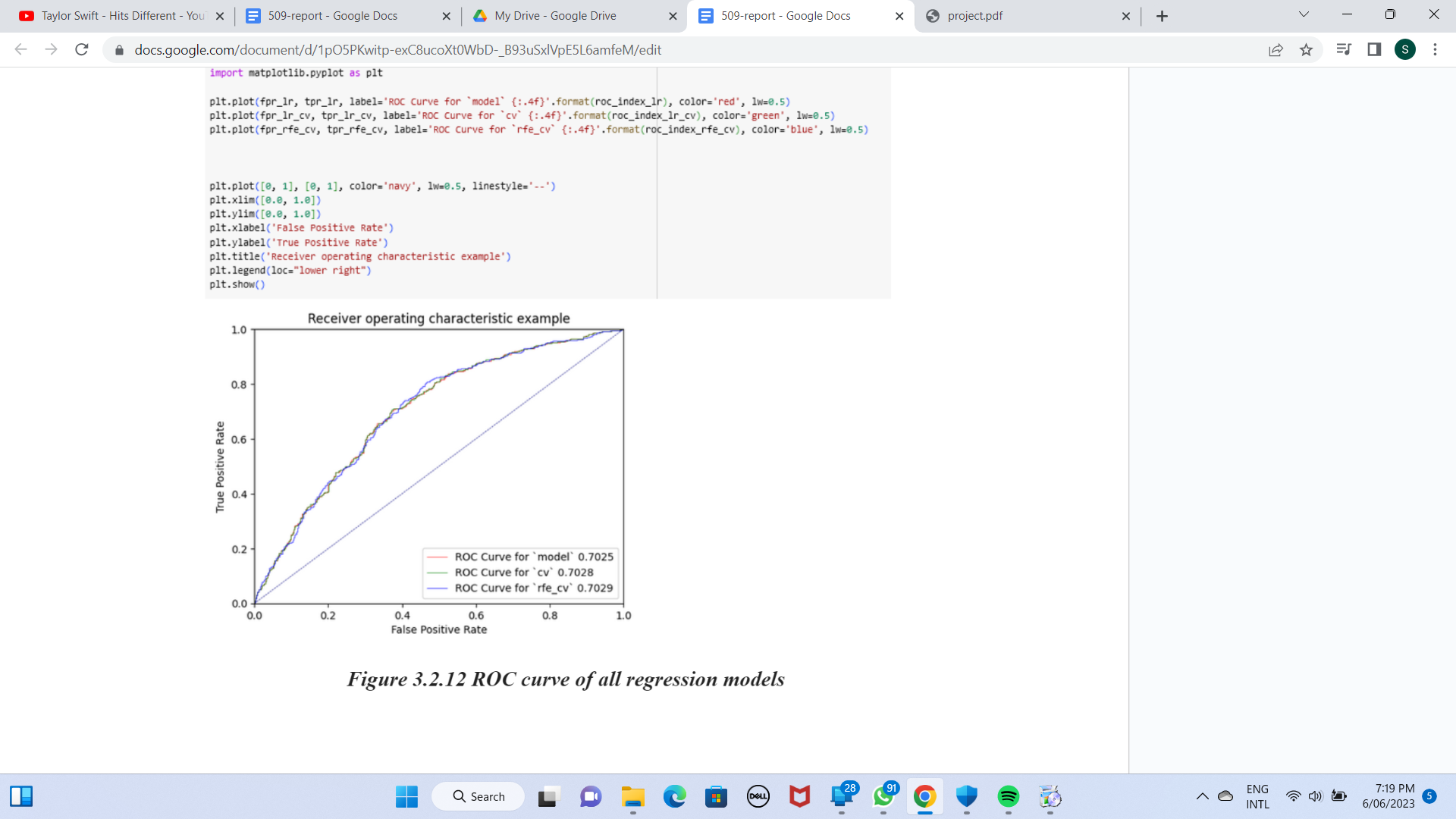
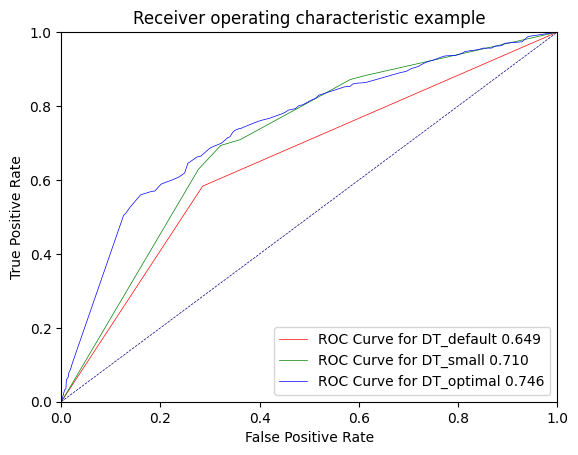


Figure 3.3.6 ROC graphs

1. **Can you summarise the positives and negatives of each predictive modelling method based on this analysis?**

Decision tree - due to the scale of the task, training the decision trees was relatively easier. Even when performing an extensive search of the best hyper-parameters it was fast. The decision tree algorithm was also the easiest to interpret because it would output a visual of what the actual result looks like - so the decision tree had more interpretability and understandability. The decision trees can handle both categorical and numerical variables without requiring any extra pre processing which makes it easier to use too. When considering the disadvantages of this algorithm, it can be very prone to overfitting if not taken precautions. Because unless the relevant restrains are held upon the decision tree, the algorithm tends to produce very deep trees with a lot of nodes and rules which means there’s a high probability that the tree is overfitting on the training data. Parameters like max\_depth should be fine tuned accordingly to avoid this from happening.

Logistic regression - similar to decision trees, logistic regression is also a very easy-to-use model, where training and testing the model is relatively fast and does not require too much computational power either. Feature importances of the model also easily lets us interpret which attributes used in the model has a higher impact on the results - which gives us high interpretability from the logistic regression model. One of the disadvantages of logistic regression comes up when using it on datasets with outliers - because logistic regression is sensitive to outliers. So, if there are any outliers in the dataset, they need to be handled accordingly.

Neural networks - one of the main advantages of neural networks is that they are very capable of learning complex relationships that exist between input variables of a dataset and they can also deal well with unstructured data by utilising hidden layers. Neural networks also work well with noisy data due to its high pattern recognition capabilities. As for the disadvantages of neural networks - they are prone to overfitting as well, so it needs careful hyper parameter tuning to make sure that it does not overfit on the training data. When hyper parameter tuning a lot of hyper-parameters need to be considered with attention in order to get a best performing model. Training neural networks can also be very computationally expensive, with it taking a long time to reach convergence after a high number of iterations or epochs, and can most of the time require high device specifications to successfully train a neural network. Another disadvantage that we had to face when training the neural networks was the low interpretability of the algorithm (also sometimes referred to as the “black box nature”of neural networks) – because of this it was touch to interpret what the results of the neural network actually meant or how they ended up with the relevant results.