Classification of urban areas

Question 3.1

The decision tree for the classification of urban areas is located in a source code file called decision_tree.py.

```
|--- buildings_age <= 35.53
|   |--- ThirdPlaces:oa_count <= 0.01
|   |   |--- buildings_age <= 25.80
|   |   |   |--- TrafficPoints:crossing <= 0.03
|   |   |   |   |--- TrafficPoints:crossing <= 0.00
|   |   |   |   |   |--- ThirdPlaces:out_count <= 0.00
|   |   |   |   |   |   |--- ThirdPlaces:out_count <= 0.00
|   |   |   |   |   |   |   |--- buildings_age:diversity <= 1.58
|   |   |   |   |   |   |   |   |--- class: age_over_60
|   |   |   |   |   |   |   |--- buildings_age:diversity >  1.58
|   |   |   |   |   |   |   |   |--- class: age_under_18
|   |   |   |   |   |   |--- ThirdPlaces:out_count >  0.00
|   |   |   |   |   |   |   |--- class: age_over_60
|   |   |   |   |   |--- ThirdPlaces:out_count >  0.00
|   |   |   |   |   |   |--- class: age_under_18
|   |   |   |   |--- TrafficPoints:crossing >  0.00
|   |   |   |   |   |--- ThirdPlaces:oa_count <= 0.00
|   |   |   |   |   |   |--- Buildings:diversity <= 0.32
|   |   |   |   |   |   |   |--- class: age_30_40
|   |   |   |   |   |   |--- Buildings:diversity >  0.32
|   |   |   |   |   |   |   |--- class: age_under_18
|   |   |   |   |   |--- ThirdPlaces:oa_count >  0.00
|   |   |   |   |   |   |--- class: age_over_60
```

```
|   |   |   |   |   |   |   |   |   |--- class: age_40_50
|   |   |   |   |   |   |   |   |--- poisAreas:area_park >  0.00
|   |   |   |   |   |   |   |   |   |--- class: age_over_60
|   |   |   |   |   |   |--- pois:diversity >  4.27
|   |   |   |   |   |   |   |--- pois:diversity <= 4.38
|   |   |   |   |   |   |   |   |--- class: age_50_60
|   |   |   |   |   |   |   |--- pois:diversity >  4.38
|   |   |   |   |   |   |   |   |--- class: age_40_50
|   |   |   |--- Buildings:diversity >  1.58
|   |   |   |   |--- ThirdPlaces:oa_count <= 0.24
|   |   |   |   |   |--- class: age_over_60
|   |   |   |   |--- ThirdPlaces:oa_count >  0.24
|   |   |   |   |   |--- class: age_40_50
|   |   |--- buildings_age:diversity >  1.30
|   |   |   |--- class: age_30_40
```

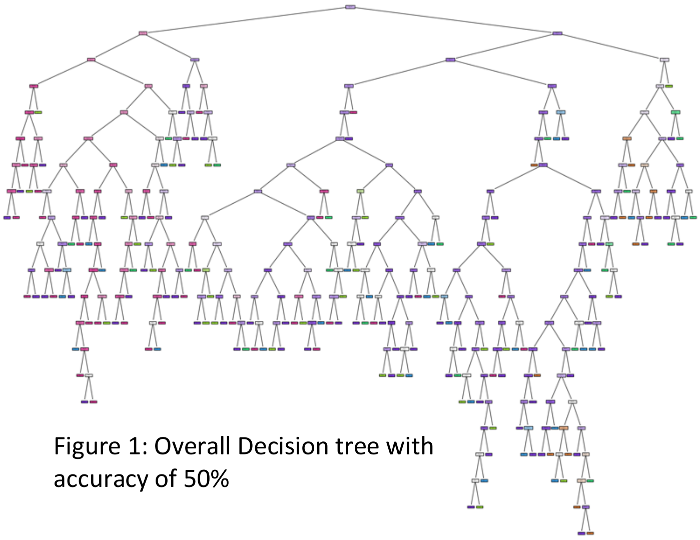

Figure 1: Overall Decision tree with accuracy of 50%

When performing the decision tree 'cell_id' wouldn't be a decision node, since it is just simply a label which refers to an identifier for each of the cells in which the city of Rome is divided. Adding 'cell_id' will reduce the overall accuracy of the tree. The featured columns will need every column except the 'cell_id', 'Roads:total', 'Buildings:total', 'pois:total', 'ThirdPlaces:total' and 'most_present_age'. This is because all the remaining columns help the program train, test and predict for the decision tree. The featured columns don't contain any columns with total density. This because the sub-categories are being processed so there isn't any need for the total columns. Finally, the target column for the tree is 'most_present_age'. Furthermore, an overview of the decision tree is presented in Figure 1.

Question 3.2

The decision tree for the classification of urban areas is located in a source code file called decision_tree.py.

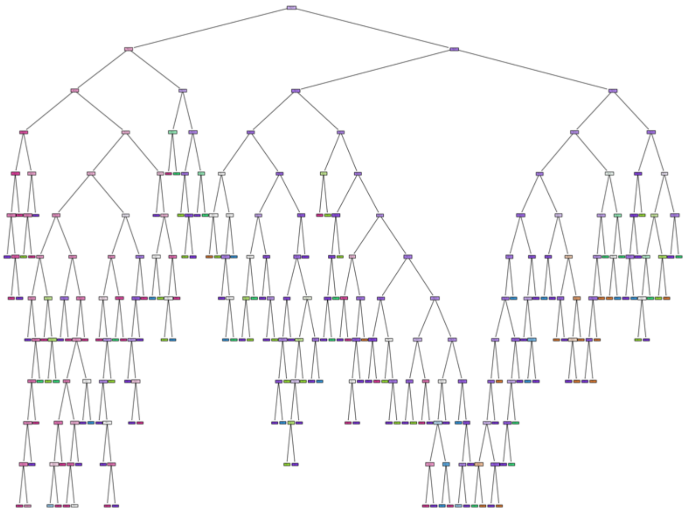

Figure 2: Decision tree with a maximum depth = 12. This has an accuracy of 52%

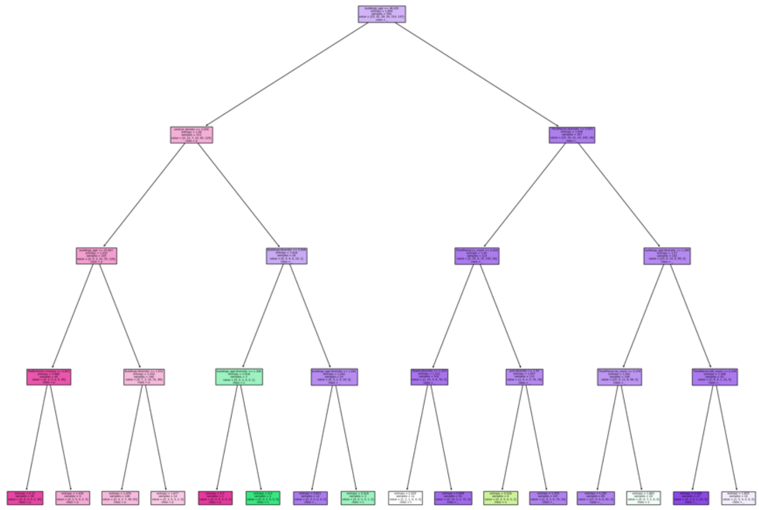

Figure 3: Decision tree with a maximum depth = 4. This has an accuracy of 60%

From figure 1 we know that the accuracy for the overall decision tree is 50%. Whereas, when the maximum depth of the tree is set to 12 the accuracy increased to 52%, seen in figure 2. Similarly, in figure 3, when the maximum depth is reduced to 4 the accuracy increased to 60%.

The maximum depth that you allow the tree to grow to. The deeper you allow, the more complex your model will become. For training error, it is easy to see what will happen. If you increase the maximum depth, training error will always go down. For testing error, it gets less obvious. If you set maximum depth is too high, then the decision tree might simply overfit the training data without capturing useful patterns as we would like; this will cause testing error to increase. But if you set it too low, that is not good as well; then you might be giving the decision tree too little flexibility to capture the patterns and interactions in the training data. This will also cause the testing error to increase.

This is all due to instability and complexity. This is because decision trees are heavily dependent on the data they are given and whereas a tree can become extremely accurate if you provide it with additional data the whole tree can be thrown off. Furthermore, decision trees are easy to use compared to other decision making models, however a very complex decision tree with multiple branches can be very time consuming both in preparation and computation.