Beginning of the course, students were asked to fill out a questionnaire with information about themselves. The data set includes a total of 63 objects and 45 attributes. This paper seeks to answer the following questions with the associated data mining methods:

* How can students be grouped in terms of height and shoe size (K-Means)
* Which programming languages frequently appear together (Aprori Algorithm)
* What are the best questions to ask to identify a female student (Decision Tree)

Prior to the data mining analysis, the data had to cleaned. The focus of the data cleaning were on the following attributes:

* Age
* Gender
* Shoe Size
* Height
* Which programming languages do you know?
* Which phone OS do you prefer?

The cleaning of the Interval attributes (’Age’, ’Shoes Size’, and ’Height’) included removing the metric measurement scale (Height) and ensuring the same type of decimal mark (Shoe Size). The American decimal mark was chosen as it is compatible with the math operations in python. For the nominal attributes (’Gender’, ’Which programming languages do you know’, and ’Which phone OS do you prefer’), the string values were standardized.

First step before the clustering is to normalize the data to avoid any skewing of the values in the attributes. Next step is the actual implementation of K-Means. Here are the steps used to create the clustering function:

1. Determine the number of clusters
2. Assign each data point to its closest mean
3. Calculate the means based upon the assignments of data points
4. Continue this process until the assignments don’t change

The distance between observations is calculated with the Pythagoras Theorem: a2 + b2 = c2, as the dimensionality of the data is two dimensional. If more dimensionalities were included the Manhattan distance would be preferred.

To answer the initial question of how height and shoe size would be clustered, the matplot library is used to visualize the data. From the visualization a clear trend emerges: The two clusters are separated into high and low values of the two attributes. Further, it also appears as if the two are linearly correlated.

First step for the aprori algorithm is to define the threshold for frequent items/transactions to be considered frequent. The threshold is a subjective parameter and depends on the context and the size of the data set. To determine the threshold the counter function was applied to the cleaned formatted data set. The top five most frequent languages were determined to be the threshold for frequent transactions, capping frequent transactions at 19:

* c sharp: 28
* c++: 22
* java: 61
* python: 21
* javascript: 19

Combinations of set of two are then generated from the languages above. As there were five languages, the total number unique combinations are 10. Frequencies of each set is then calculated similar to before:

* c sharp, c++: 19
* c sharp, java: 27
* c sharp, javascript: 12
* c sharp, python: 13
* c++, java: 21
* c++, ’javascript: 9
* c++, python: 9
* java, javascript: 18
* java, python: 19
* javascript, python: 10

In the frequents sets of size two, it is only the sets: (c sharp, c++), (c sharp, java), (c++ java), and (java, python) that meets the criteria as frequent (minimum threshold of 19). Javascript is the only language from the previous frequent set of size one that is not frequent in the list of size two. Therefore, the same process is repeated with combination sets of size three without javascript. These are the results:

* c sharp, c++, java: 18
* c sharp, c++, python 8
* c sharp, java, python 12
* c++, java, python 8

As seen from the results above there are no frequent sets of size three. That said, the set (c sharp, c++, and java) is only short by one.

In the last part of the assignment, a decision tree is applied to determine the sequence of questions to ask in order to determine a female student in the class. The attributes used for the decision tree are: "Age", "Shoe Size", "Height", and "Favorite Mobile OS".

To begin with all the attributes are saved into a list of tuples each containing a dictionary of the attribute labels and values and a boolean determining whether or not the student is female or male. Since the goal is to determine a female student the boolean value True is associated with female. In the implementation of the decision tree, the paths are determined based on the attributes that generate the lowest entropy. Similar to clustering with KMeans, this approach can lead to local optimums.

In the first split, Shoe Size has the lowest entropy. More accurately the best separation of genders is made by asking the question: Does the student have a shoe size over 40? If the answer is over 40, the student is a male. However, the answer isn’t clear with students of shoe sizes of 40 and under. Therefore we need to generate another split for this grouping. The same process is repeated until the entropy level is 0 or there are no more attributes. In this decision tree, all attributes are applied and even for the last attribute, ’Favorite OS’, it is not possible to split with an entropy of 0. From this it can be concluded that in specific cases, it is not possible to determine whether the student is a male or female, based upon questions related to the shoe size, height and age. Nonetheless, here is the list of questions generated by the decision tree to determine the gender:

1. Is the shoe size over 40?
2. Is the height over 172?
3. Is the age over 25