I started this assignment wondering which questions should I ask with the given data. As a first impression, a relation between height, show size, reasons behind why taking this course, and which program each student takes, was not clear at all. So my process started organizing the data, and later process it through K-Means to understand its possibilities, find a relation, re frame it, and propose a question for Naive Bayes to answer.

For K-Means I tried to find a relation between 'Shoe size' and 'Height' of the participants, and for Naive Bayes I wanted to predict if the participant attends to a master at 'ITU' or 'Elsewhere' based on the 'Show size', 'Height' and 'Why are you taking this course?' columns.

Pre-processing. The given dataset had missing information, wrong unit inputs, and the 'Why are you taking…' column was not divided and for some rows there was only one reason chosen, and for other there were many. I dropped the 'TimeStamp' column as I didn't intend to work with it.

Given the values on the 'Height' column, I decided that all inputs lower than 10 and greater than 80 where written wrongly in feet or cm, so I converted them to international inches. Besides, if the vale was 0, I replaced it with the mean.

```
In [5]: #checking odd inputs and converting from cm or feet to inches. If input is nul, replace it with mean
for x in range(len(df['Your height (in International inches)'])):
    if df['Your height (in International inches)'][x] > 80:
        df['Your height (in International inches)'][x] = df['Your height (in International inches)'][x]*0.3937
    if df['Your height (in International inches)'][x] < 10:
        df['Your height (in International inches)'][x] = df['Your height (in International inches)'][x]*12
    if df['Your height (in International inches)'][x] == 0:
        df['Your height (in International inches)'][x] = df['Your height (in International inches)'].mean()
```

In addition, I gave it a nice float formating, and as seen on the image, I decided to change change inputs to integers on the 'Shoe' column, as some of the values in float didn't have much sense, for example: 43.13231 became 43.

```
In [5]: #formating float for clarity
df['Your height (in International inches)'] = df['Your height (in International inches)'].map('{:.2f}'.format)

In [7]: #changing data type for formating and manipulation
df['Shoe'] = df['Shoe'].astype(int)
```

The most difficult column to work with was the 'Why…'. I decided to separate the input of each row with ';' as separator. The first attempt was to create a single column with each separated-input, as a new row… but I missed the link with *to which program* was the input associated to. ->

Out[11]:

	Why Taking This Course
0	It'll give me business skills
1	This was mandatory for me
2	It'll give me business skills
3	This was mandatory for me
4	This was mandatory for me
...	...
82	Specialization track
83	The other courses were even less attractive

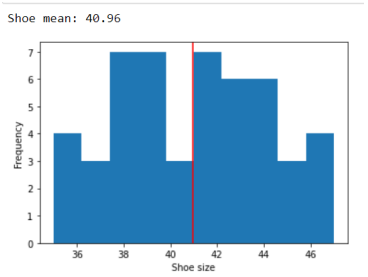
My solution was to use use the separator to create new columns on the dataset, keeping the relation to the master program, and then transposing and copying the information to a new row, to keep the which-why relation. I also changed 'Why' inputs like "specialization" and "Specialization track", to be the same and improved my dataset for later calculations (see on in[16] of my code). And this is what I got ->

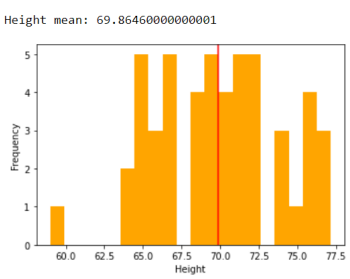
Out[16]:

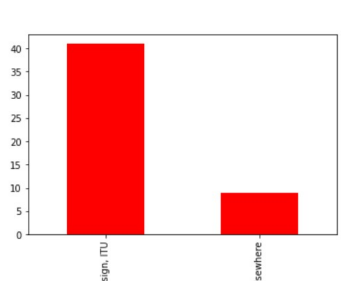
	Which programme are you studying?	Why are you taking this course?
0	MSc Software Design, ITU	It'll give me business skills
...	...	...
86	MSc Software Design, ITU	Specialization track

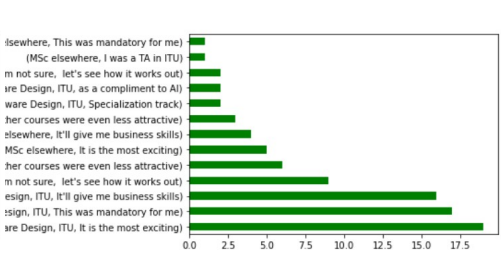
87 rows × 2 columns

I also plotted my data before applying K-means and Naive Bayes (more on code).



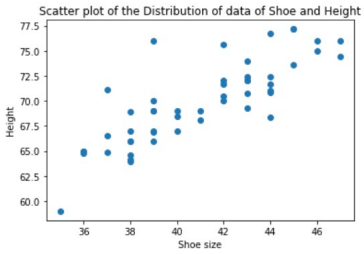


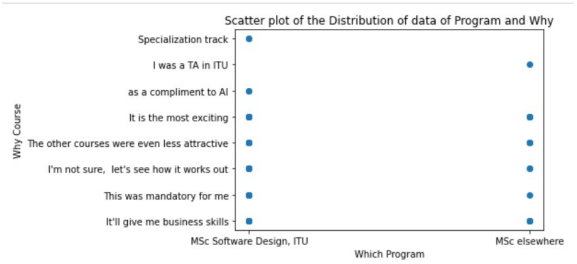




K-Means (clustering).

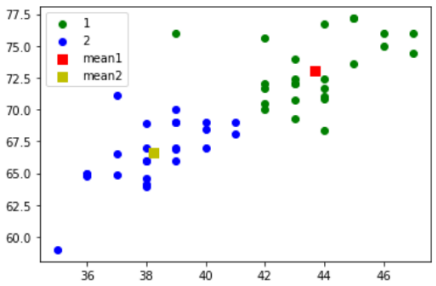
After plotting the column 'Shoe' in relation to 'Height', and 'Why' in relation to 'Program'. I decided to apply K-Means to the first one and find its clusters. I tried first normalizing the data, but I found it confusing to understand the relation.





I applied the code we worked in the lab that involved choosing random means, calculating the euclidean distance of every point to those means, assigned a class in relation to the closest and update the mean. All these steps where called under the function *"def Kmeans(dataf,iterations):" and* after 4 iterations, I found the right clustering for the data.

```
In [30]: #a function that uses the previous def a number of times (iterations), claculating distance and updating the mean
def Kmeans(dataf,iterations):
    mean1,mean2 = initializeMeans(dataf)

    for iteration in range(iterations):

        print("Iteration {}/{}".format(iteration,iterations))

        for i in range(len(dataf)):
            dataf = euclideanDist(dataf,i,mean1,mean2)

        mean1,mean2 = updateMean(dataf)

    return dataf, mean1, mean2
```



And it means that we have two groups that fall under the following characteristics:

GROUP 1		GROUP 2	
max shoe size: 47	max height size: 77.17	max shoe size: 41	max height size: 71.17
min shoe size: 39	min height size: 68.4	min shoe size: 35	min height size: 59.0
shoe size mean: 43.68	height size mean: 73.06	shoe size mean: 38.24	height size mean: 66.65

Naive Bayes (supervised).

I used the data set as training for my code to predict 'which program the student attend' given shoe size, height in inches, and reason to study data mining. In this sense, I used all 50 entries, as inputs for 'Msc Elsewhere' might have been not enough on a segmented set. Then, to try out the code, I wrote a new (controlled) dataset called tryDataSet.

I classified my existing data on class = 1, for Msc ITU, and class= 2, for Msc Elsewhere. Then, I separated my data into categorical data and numerical data, as for the categorical data I needed to calculate likelihood, and for numerical data, I calculated mean and standard deviation for later normal distribution calculations. I wrote functions to calculate likelihood, mean, standard deviation, and a function to build a DataFrame with this information (def sumarize_by_class).

Out[81]:

class	Probability	Shoe_mean	Your height (in International inches)_mean	Shoe_stdev	Your height (in International inches)_stdev	It is the most exciting	This was mandatory for me	It'll give me business skills	I'm not sure, let's see how it works out	The other courses were even less attractive	Specialization track	as a compliment to AI	I wa a T in IT
1	0.82	40.585366	69.323659	3.106249	4.121991	1.267606	1.239437	1.225352	1.126761	1.084507	1.028169	1.028169	1.00
2	0.18	42.666667	72.328889	3.316625	3.210130	1.312500	1.062500	1.250000	1.125000	1.187500	1.000000	1.000000	1.06

I also wrote a function to calculate normal distribution.

Finally, I wrote the function 'sumarize_by_class' that calculates the probability of each row of my tryDataSet to belong to class=1 (Msc ITU) or class=2 (Msc Elsewhere). At this point I faced the issue that if there was NaN elements in my likelihoods, my function was not going to work, so I added one count to each 'Why' input on my likelihood function.

Finally I run my function with my tryDataSet and got the following results:

```
In [44]: df_probability_calculated
```

Out[44]:

	Shoe	Your height (in International inches)	Why?	ITU	Elsewhere	class
0	41	61	This was mandatory for me	16.304765	0.182896	1
1	32	70	It is the most exciting	2.735117	0.458333	1
2	47	80	It is the most exciting	0.523434	2.575265	2
3	50	67	Specialization track	1.091154	2.298914	2
4	39	74	The other courses were even less attractive	58.290498	49.804684	1
5	40	69	Specialization track	123.763560	44.425219	1
6	43	73	Specialization track	62.761904	102.290345	2
7	39	66	This was mandatory for me	80.149170	8.167229	1
8	43	73	I was a TA in ITU	62.761904	108.683491	2

```
#Where 1 is 'MSc ITU' and 2 is 'MSc Elsewhere'
```

What it makes me think that the code works as it calculates the probability of new inputs per row to belong to class 1 or 2.

Miguel Angel Crozzoli (Oct 2021 _ Data Mining _ fall semester)