|  |  |  |
| --- | --- | --- |
|  | **FACULTY of COMPUTING, ENGINEERING & SCIENCE** | Final mark awarded:\_\_\_\_\_ |

**Assessment Cover Sheet and Feedback Form 2017/18**

|  |  |  |  |
| --- | --- | --- | --- |
| Module Code:  CS4S767 | Module Title:  Data Mining | | Module Lecturer:  Bertie Muller & Andrew Ware |
| Assessment Title and Tasks:  Health Analytics | | | Assessment No.  2 of 2 |
| No. of pages submitted in total including this page:  31 | | | Word Count of submission  (if applicable): 1.9k |
| Date Set:  Monday19th February 2018 | | Submission Date:  Friday 23rd March 2018 | Return Date:  Friday 20th April 2018 |

|  |  |
| --- | --- |
| ***Part A: Record of Submission (to be completed by Student)*** | |
| **Extenuating Circumstances**  If there are any exceptional circumstances that may have affected your ability to undertake or submit this assignment, make sure you contact the Advice Zone on your campus prior to your submission deadline. | |
| **Fit to sit policy**:  The University operates a fit to sit policy whereby you, in submitting or presenting yourself for an assessment, are declaring that you are fit to sit the assessment. You cannot subsequently claim that your performance in this assessment was affected by extenuating factors. | |
| **Plagiarism and Unfair Practice Declaration:**  By submitting this assessment, you declare that it is your own work and that the sources of information and material you have used (including the internet) have been fully identified and properly acknowledged as required[[1]](#footnote-1). Additionally, the work presented has not been submitted for any other assessment. You also understand that the Faculty reserves the right to investigate allegations of plagiarism or unfair practice which, if proven, could result in a fail in this assessment and may affect your progress. | |
| **Intellectual Property and Retention of Student Work:**  You understand that the University will retain a copy of any assessments submitted electronically for evidence and quality assurance purposes; requests for the removal of assessments will only be considered if the work contains information that is either politically and/or commercially sensitive (as determined by the University) and where requests are made by the relevant module leader or dissertation supervisor. | |
| **Details of Submission:**  Note that all work handed in after the submission date and within 5 working days will be capped at 40%[[2]](#footnote-2). No marks will be awarded if the assessment is submitted after the late submission date unless extenuating circumstances are applied for and accepted (Advice Zone to be consulted). | |
| You are required to acknowledge that you have read the above statements by writing your student number(s) in the box: | Student Number:  13054511 |

**IT IS YOUR RESPONSIBILITY TO KEEP RECORDS OF ALL WORK SUBMITTED**

|  |
| --- |
| **Part B: Marking and Assessment**  **(to be completed by Module Lecturer)** |
| This assignment will be marked out of 100%  This assignment contributes to 50% of the total module marks.  This assignment is non-bonded. Details: You must achieve 40% overall to pass the module. |
| **Assessment Task:**  You must complete a report, details of which are described in Assignment Task |
| **Learning Outcomes to be assessed** (as specified in the validated module descriptor <https://icis.southwales.ac.uk/> ):   1. Display knowledge of different data mining and Big Data tasks and appropriate models/algorithms evaluating these with respect to their accuracy. 2. Demonstrate the ability to apply data mining and Big Data concepts in appropriate contexts. |

**Grading Criteria:**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Criteria** | **Fail (<40)** | **Poor (40 – 49)** | **Reasonable (50 – 59)** | **Good (60-69)** | **Excellent (70+)** |
| **Background and introduction /10** | Missing or very superficial introduction. | Gives a basic insight into the aim and content of the report. | Provides a reasonable explanation of the topic and its relevance. | Clear explanation of the aim, content and conclusions of the report. | Exceptionally clear explanation of the aim, content and conclusions of the report. |
| **Analysis**  **/30** | Analytics techniques chosen are inappropriate.  Analysis performed and interpreted with many or major errors.  Technical reporting of analysis contains many or major errors.  Analysis performed is very basic and does not demonstrate sufficient level of skill. | Several errors in choice of analytics techniques or justifications are unclear.  Analysis performed and interpreted with several errors.  Technical reporting of analysis contains several errors.  Analysis performed is basic and demonstrates low level of skill. | Minor errors in choice of analytics techniques or justifications  Analysis performed and interpreted with some errors  Technical reporting of analysis contains some errors  Analysis performed is of moderate complexity and demonstrates some level of skill | Analytics techniques chosen appropriately although justifications could be slightly clearer.  Analysis performed and interpreted with only minor errors.  Technical reporting of analysis contains minor errors.  Analysis performed is fairly complex and demonstrates good level of skill. | Analytics techniques chosen appropriately with justifications  All analysis performed and interpreted correctly.  Technical reporting of analysis is complete and correct.  Analysis performed is complex and demonstrates high level of skill. |
| **Predictions**  **/40** | Predictive techniques chosen are inappropriate.  Only a single technique has been applied.  Predictions are not accurate. | Several errors in choice of predictive techniques or justifications are unclear.  Only two techniques have been applied.  Predictions are not always accurate. | Minor errors in choice of predictive techniques or justifications.  Two or fewer techniques have been applied.  Predictions are not always accurate. | Predictive techniques chosen appropriately although justifications could be slightly clearer.  Three or fewer techniques have been applied.  Analyses performed have led to accurate predictions. | Predictive techniques chosen appropriately with justifications.  Four or more techniques have been applied  Analyses performed have led to accurate predictions. |
| **Findings and Recommendations**  **/20** | Poor or superficial explanation of conclusions that contains many errors in answering questions. | Basic explanation of conclusions that answer some research questions accurately, but also contain several errors. | Reasonable explanation of conclusions that answer most research questions accurately, but also contain some errors. | Clear explanation of conclusions that answer research questions. | Thorough and concise explanation of conclusions that answer research questions effectively. |
| **%** |  |  |  |  |  |

|  |  |  |
| --- | --- | --- |
| **Feedback/feed-forward** (linked to assessment criteria):   * Areas where you have done well: * Feedback from this assessment to help you to improve future assessments: * Other comments | | |
| **Mark:** | **Marker’s Signature:** | **Date:** |
| **Work on this module has been marked, double marked/moderated in**  **line with USW procedures.** | | |
| *Provisional mark only: subject to change and/or confirmation by the Assessment Board* | | |

|  |  |
| --- | --- |
| **Part C: Reflections on Assessment**  **(to be completed by student – optional)** | |
| **Use of previous feedback:**  In this assessment, I have taken/took note of the following points in feedback on previous work: | |
| **Please indicate which of the following you feel/felt applies/applied to your submitted work**   * A reasonable attempt. I could have developed some of the   X  sections further.   * A good attempt, displaying my understanding and learning, with   analysis in some parts.   * A very good attempt. The work demonstrates my clear   understanding of the learning supported by relevant literature and scholarly work with good analysis and evaluation.   * An excellent attempt, with clear application of literature and   scholarly work, demonstrating significant analysis and evaluation. | |
| **What I found most difficult about this assessment:** | Python, I’m just awful with it. |
| **The areas where I would value/would have valued feedback:** | I prefer none, I’ve not done well on this assignment |

# Background and Introduction

### Aim

The aim of this report is to first conduct an analysis of health score data to ascertain the relative health scores of a population. Then to use the results of the analysis to drive predictive models that will help identify the health scores of individuals. The data used for analytics covers:

* Age
* Sex
* Weight
* Height
* IQ
* Units of Alcohol Consumed per Day
* Cigarettes Consumed per Day
* Level of Activity
* Final Health Score

### Content

The analytical requirements have been set in the form of three questions:

* Are there any significant differences between different segments of the population in terms of their lifestyle choices (for example, male and female, different age groups)?
* Which individual characteristics and lifestyle choices impact a person’s health score (and to what extent)?
* What would be the impact on the overall population (in terms of health score) if nobody consumed alcohol and did not smoke?

The predictive requirements have been set as predicting the health scores of 20 individuals based on the health data supplied.

### Conclusions

In terms of analytics, this report deals with the basic analysis of the data, showing the mean and median results for each of the health categories. A good deal of statistical detail is made on each group.

In summary, I did not manage to successfully implement predictive algorithms, though came very close to completing linear regression. The algorithm worked for the training data but did not for the test data.

# Analysis

### Analytical Techniques Chosen

Techniques will be aimed at answering the three questions posed in turn:

* Comparison of health score with each data value available (alcohol consumption, male/female)
* In generalised terms for age groups and other highly variable factors with a look into the depth of impact each factor has
* A look at the impact on the population if there was no smoking and drinking in a population

Specifically, I will be examining the mean and median values within each data set, then comparing the results. From a general set of data these are very effective methods of examining trends.

KMeans will be used to observe features within the data as it visualises clusters of similar data.

### Results of Analytical Techniques

#### Age

The ages within the data vary from between 18 and 80, obtained using min and max functions. I have separated the ages into the age groups of 18-30, 31-40, 41-50, 51-60, 61-70, 71-80.

The mean health score of each age group was as follows:

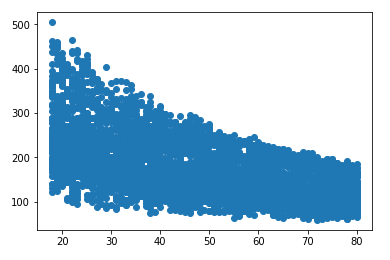
* 18-30: 238.669912366
* 31-40: 196.799758745
* 41-50: 167.826923077
* 51-60: 148.592405063
* 61-70: 129.027918782
* 71-80: 116.848407643

The 18-30 age group had a 204.3% higher mean value than the 71-80 age group.

The median health score of each age group was as follows:

* 18-30: 225.0
* 31-40: 188.0
* 41-50: 159.0
* 51-60: 143.5
* 61-70: 121.0
* 71-80: 109.0

The 18-30 age group had a 206.4% higher median value than the 71-80 age group.



#### Male/Female

The mean health score of males and females were:

* Males: 141.353159851
* Females: 187.351091142

Females are therefore 32.5% more healthy on average than males.

The median results were:

* Males: 131.0
* Females: 174.0

Females having a 32.8% higher median value than males.

#### Weight

The weight within the data varies from between 77 pounds and 217 pounds. Division into 5 equal groups gives a 28-pound range for each group.

The mean health score of each weight group was as follows:

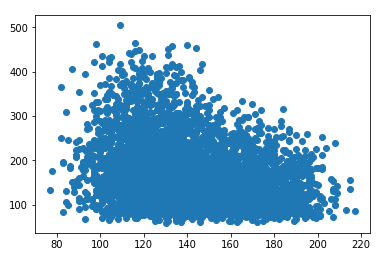
* 77 - 104: 201.438016529
* 105 - 132: 190.261529809
* 133 - 160: 167.562848297
* 161 - 188: 141.699077955
* 189 - 217: 136.023391813

The 77 – 104 weight group had a 148.1% higher mean value than the 189 – 217 weight group.

The median health score of each weight group was as follows:

* 77 - 104: 188.0
* 105 - 132: 177.0
* 133 - 160: 155.0
* 161 - 188: 131.0
* 189 - 217: 130.0

The 77 – 104 weight group had a 144.6% higher median value than the 189 – 217 weight group.



#### Height

The height within the data varies from between 56 inches and 83 inches. I’ve divided this into 4 groups of 7 inches with the 1st group containing 6.

The mean health score of each height group was as follows:

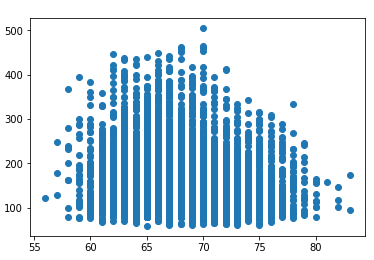
* 56 - 62: 173.597938144
* 63 - 69: 181.708484409
* 70 - 76: 152.313771063
* 76 - 83: 146.318181818

The 63 – 69 height group had a 124.2% higher mean value than the 76 – 83 height group. This is the first result that does not have an upper and lower bound on the extremes.

The median health score of each height group was as follows:

* 56 - 62: 161.0
* 63 - 69: 167.0
* 70 - 76: 138.0
* 76 - 83: 135.5

The 56 – 62 height group had a 123.2% higher median value than the 76 – 83 height group.



#### IQ

The IQ within the data varies from between 79 and 122. Division into 4 groups gives an 11-point range for each group except for the 1st group which has 10 points.

The mean health score of each IQ group was as follows:

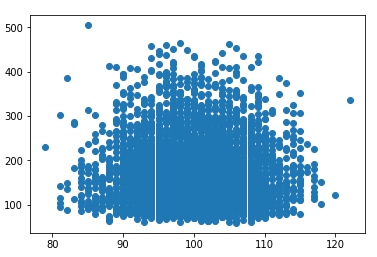
* 79 - 89: 168.286486486
* 90 - 100: 172.073260073
* 101 - 111: 168.031151242
* 112 - 122: 167.943661972

The 90 – 100 IQ group had a 102.4% higher mean value than the 112 – 122 IQ group. Another case where it is not the extremes that have the highest contrast.

The median health score of each IQ group was as follows:

* 79 - 89: 155.0
* 90 - 100: 159.0
* 101 - 111: 154.0
* 112 - 122: 153.0

The 90 – 100 IQ group had a 103.9% higher median value than the 112 – 122 IQ group.



#### Units of Alcohol Consumed Per Day

The units of alcohol consumed within the data varies from between 0 and 7. I’ve divided the dataset into groups according to zero, light, moderate and heavy consumption.

The mean health score of each group was as follows:

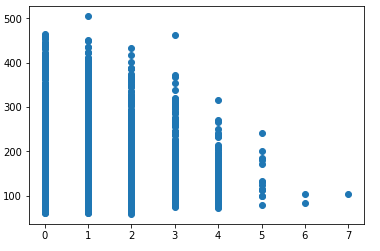
* Non-Drinker: 175.188859416
* 1 - 2: 168.22483468
* 3 - 4: 158.765333333
* 5+: 134.0

The Non-Drinker group had a 130.7% higher mean value than the 5+ a day group.

The median health score of each group was as follows:

* Non-Drinker: 161.0
* 1 - 2: 154.0
* 3 - 4: 149.0
* 5+: 115.0

The Non-Drinker group had a 140% higher median value than the 5+ a day group.



#### Cigarettes Smoked Per Day

The number of cigarettes smoked a day within the data varies from between 0 and 40. I’ve divided the dataset into groups according to zero, light, moderate and heavy consumption.

The mean health score of each group was as follows:

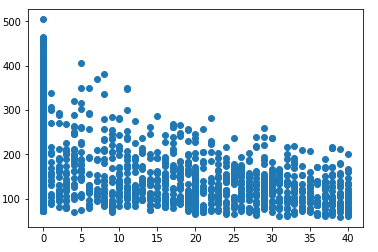
* Non-Smoker: 177.234580134
* 1 - 10: 168.502262443
* 11 - 20: 144.482889734
* 21+: 123.891213389

The Non-Smoker group had a 143.1% higher mean value than the 21+ a day group.

The median health score of each group was as follows:

* Non- Smoker: 163.0
* 1 - 10: 151.0
* 11 - 20: 131.0
* 21+: 118.5

The Non-Smoker group had a 137.6% higher median value than the 21+ a day group.



#### Level of Activity

The amount of activity within the data varies from between inactive to very active. Given the explicit declaration, I have counted it based on the 3 groups available in the data.

The mean health score of each group was as follows:

* Inactive: 134.628286853
* Active: 193.719018405
* Very Active: 228.492433062

The Very Active group had a 169.7% higher mean value than the Inactive group.

The median health score of each group was as follows:

* Inactive: 124.0
* Active: 181.0
* Very Active: 213.0

The Very Active group had a 171.8% higher median value than the Inactive group.

# Predictions

### Predictive Techniques Chosen

Linear Regression has been selected to predict the health values due to its ability to deal with n-dimensional data, using a training set to test its accuracy followed by the application of the algorithm then on a set of novel data. This makes it perfect for estimating the health scores that are not known.

### Results of Predictive Techniques

|  |  |  |
| --- | --- | --- |
| Person ID | Technique Name | Final Answer |
| 1 | Regression Analysis | 191 |
| 2 | Regression Analysis | 99 |
| 3 | Regression Analysis | 135 |
| 4 | Regression Analysis | 126 |
| 5 | Regression Analysis | 114 |
| 6 | Regression Analysis | 100 |
| 7 | Regression Analysis | 119 |
| 8 | Regression Analysis | 177 |
| 9 | Regression Analysis | 185 |
| 10 | Regression Analysis | 105 |
| 11 | Regression Analysis | 157 |
| 12 | Regression Analysis | 172 |
| 13 | Regression Analysis | 132 |
| 14 | Regression Analysis | 126 |
| 15 | Regression Analysis | 134 |
| 16 | Regression Analysis | 143 |
| 17 | Regression Analysis | 98 |
| 18 | Regression Analysis | 162 |
| 19 | Regression Analysis | 188 |
| 20 | Regression Analysis | 134 |

# Findings and Recommendations

### Analytic Methods

The basic work I conducted on getting the mean and median to represent the populations was easy enough and there were some interesting observations:

* The single most important factor in being healthy is age with a more than 200% difference in health between the youngest and oldest on average
* The second most important factor was activity at about 170% better health for the most active compared to the most inactive
* Women are healthier than men, though all other lifestyle factors have bigger impacts than sex

I attempted KMeans, which worked to an extent except for being able to render different clusters of colour. Clusters were identified but I got RGB errors for too long and gave up on it in favour of linear regression.

Regarding removing smoking and drinking from the samples entirely, we see a decrease of the mean differences between various groups and as expected an increase in the general health of the population.

### Predictive Methods

The values I’ve entered in the table are categorically not correct. I know this because I studied the data associated with the persons by eye and through comparison to the other dataset. A simple glance at the mean and medians also points towards higher values across the board. While my predictive method against the original dataset worked with accuracy, I couldn’t get it working on the population dataset. I spent considerable time on trying to get it working including a whole round of normalisation on the advice of a friend of all data used for the regression and after even all that, I got the same values coming back. I figure I have some error in the code for it somewhere, but I haven’t been able to identify it and neither have colleagues when I’ve posted the regression code I have publicly.

# Appendix – Code Listing

### File Read In

# -\*- coding: utf-8 -\*-

import numpy as np

from sklearn.cluster import KMeans

#Read file in

dataArray = np.genfromtxt('HealthScores.csv', delimiter=',', dtype=None).astype(str)

#Extract variables from raw input

age = [dataArray[i][0] for i in range(len(dataArray))]

sex = [dataArray[i][1] for i in range(len(dataArray))]

weight = [dataArray[i][2] for i in range(len(dataArray))]

height = [dataArray[i][3] for i in range(len(dataArray))]

iq = [dataArray[i][4] for i in range(len(dataArray))]

unitsAlcohol = [dataArray[i][5] for i in range(len(dataArray))]

unitsCigs = [dataArray[i][6] for i in range(len(dataArray))]

activity = [dataArray[i][7] for i in range(len(dataArray))]

healthScore = [dataArray[i][8] for i in range(len(dataArray))]

#strip off column labels

del age[0]

del sex[0]

del weight[0]

del height[0]

del iq[0]

del unitsAlcohol[0]

del unitsCigs[0]

del activity[0]

del healthScore[0]

#convert to ints where appropriate

age = list(map(int, age))

weight = list(map(int, weight))

height = list(map(int, height))

iq = list(map(int, iq))

unitsAlcohol = list(map(int, unitsAlcohol))

unitsCigs = list(map(int, unitsCigs))

healthScore = list(map(int, healthScore))

### Pre-Processing of Data

#setup arrays for loop use

femaleScores = []

maleScores = []

genderNumberScores = []

scores1830 = []

scores3140 = []

scores4150 = []

scores5160 = []

scores6170 = []

scores7180 = []

scoresWeight77 = []

scoresWeight105 = []

scoresWeight133 = []

scoresWeight161 = []

scoresWeight189 = []

scoresHeight56 = []

scoresHeight63 = []

scoresHeight70 = []

scoresHeight76 = []

scoresIQ79 = []

scoresIQ90 = []

scoresIQ101 = []

scoresIQ112 = []

scoresAlcohol0 = []

scoresAlcohol1 = []

scoresAlcohol3 = []

scoresAlcohol5 = []

scoresCigs0 = []

scoresCigs1 = []

scoresCigs11 = []

scoresCigs21 = []

scoresInactive = []

scoresActive = []

scoresVeryActive = []

scoresNumberActivity = []

### Extraction of Data

#extract female and male scores

for index, item in enumerate(sex):

if item == "Female":

femaleScores.append(healthScore[index])

genderNumberScores.append(0)

elif item == "Male":

maleScores.append(healthScore[index])

genderNumberScores.append(1)

#extract age group scores

for index, item in enumerate(age):

if int(item) >= 18 and int(item) <= 30:

scores1830.append(healthScore[index])

elif int(item) >= 31 and int(item) <= 40:

scores3140.append(healthScore[index])

elif int(item) >= 41 and int(item) <= 50:

scores4150.append(healthScore[index])

elif int(item) >= 51 and int(item) <= 60:

scores5160.append(healthScore[index])

elif int(item) >= 61 and int(item) <= 70:

scores6170.append(healthScore[index])

elif int(item) >= 71 and int(item) <= 80:

scores7180.append(healthScore[index])

#extract weight group scores

for index, item in enumerate(weight):

if int(item) >= 77 and int(item) <= 104:

scoresWeight77.append(healthScore[index])

elif int(item) >= 105 and int(item) <= 132:

scoresWeight105.append(healthScore[index])

elif int(item) >= 133 and int(item) <= 160:

scoresWeight133.append(healthScore[index])

elif int(item) >= 161 and int(item) <= 188:

scoresWeight161.append(healthScore[index])

elif int(item) >= 189 and int(item) <= 217:

scoresWeight189.append(healthScore[index])

#extract height group scores

for index, item in enumerate(height):

if int(item) >= 56 and int(item) <= 62:

scoresHeight56.append(healthScore[index])

elif int(item) >= 63 and int(item) <= 69:

scoresHeight63.append(healthScore[index])

elif int(item) >= 70 and int(item) <= 76:

scoresHeight70.append(healthScore[index])

elif int(item) >= 76 and int(item) <= 83:

scoresHeight76.append(healthScore[index])

#extract iq group scores

for index, item in enumerate(iq):

if int(item) >= 79 and int(item) <= 89:

scoresIQ79.append(healthScore[index])

elif int(item) >= 90 and int(item) <= 100:

scoresIQ90.append(healthScore[index])

elif int(item) >= 101 and int(item) <= 111:

scoresIQ101.append(healthScore[index])

elif int(item) >= 112 and int(item) <= 122:

scoresIQ112.append(healthScore[index])

#extract drinking group scores

for index, item in enumerate(unitsAlcohol):

if int(item) >= 0 and int(item) <= 0:

scoresAlcohol0.append(healthScore[index])

elif int(item) >= 1 and int(item) <= 2:

scoresAlcohol1.append(healthScore[index])

elif int(item) >= 3 and int(item) <= 4:

scoresAlcohol3.append(healthScore[index])

elif int(item) >= 5 and int(item) <= 7:

scoresAlcohol5.append(healthScore[index])

#extract smoking group scores

for index, item in enumerate(unitsCigs):

if int(item) >= 0 and int(item) <= 0:

scoresCigs0.append(healthScore[index])

elif int(item) >= 1 and int(item) <= 10:

scoresCigs1.append(healthScore[index])

elif int(item) >= 11 and int(item) <= 20:

scoresCigs11.append(healthScore[index])

elif int(item) >= 21 and int(item) <= 40:

scoresCigs21.append(healthScore[index])

#extract activity scores

for index, item in enumerate(activity):

if item == "Inactive":

scoresInactive.append(healthScore[index])

scoresNumberActivity.append(0)

elif item == "Active":

scoresActive.append(healthScore[index])

scoresNumberActivity.append(1)

elif item == "Very Active":

scoresVeryActive.append(healthScore[index])

scoresNumberActivity.append(2)

### Print Results

#print mean and median of results

print ("Mean of male scores")

print (np.mean(maleScores))

print ("Mean of female scores")

print (np.mean(femaleScores))

print ("Median of male scores")

print (np.median(maleScores))

print ("Median of female scores")

print (np.median(femaleScores))

print ("Mean of 18-30 scores")

print (np.mean(scores1830))

print ("Mean of 31-40 scores")

print (np.mean(scores3140))

print ("Mean of 41-50 scores")

print (np.mean(scores4150))

print ("Mean of 51-60 scores")

print (np.mean(scores5160))

print ("Mean of 61-70 scores")

print (np.mean(scores6170))

print ("Mean of 71-80 scores")

print (np.mean(scores7180))

print ("Median of 18-30 scores")

print (np.median(scores1830))

print ("Median of 31-40 scores")

print (np.median(scores3140))

print ("Median of 41-50 scores")

print (np.median(scores4150))

print ("Median of 51-60 scores")

print (np.median(scores5160))

print ("Median of 61-70 scores")

print (np.median(scores6170))

print ("Median of 71-80 scores")

print (np.median(scores7180))

print ("Mean of 77-104 scores")

print (np.mean(scoresWeight77))

print ("Mean of 105-132 scores")

print (np.mean(scoresWeight105))

print ("Mean of 133-160 scores")

print (np.mean(scoresWeight133))

print ("Mean of 161-188 scores")

print (np.mean(scoresWeight161))

print ("Mean of 189-217 scores")

print (np.mean(scoresWeight189))

print ("Median of 77-104 scores")

print (np.median(scoresWeight77))

print ("Median of 105-132 scores")

print (np.median(scoresWeight105))

print ("Median of 133-160 scores")

print (np.median(scoresWeight133))

print ("Median of 161-188 scores")

print (np.median(scoresWeight161))

print ("Median of 189-217 scores")

print (np.median(scoresWeight189))

print ("Mean of 56-62 scores")

print (np.mean(scoresHeight56))

print ("Mean of 63-69 scores")

print (np.mean(scoresHeight63))

print ("Mean of 70-76 scores")

print (np.mean(scoresHeight70))

print ("Mean of 76-83 scores")

print (np.mean(scoresHeight76))

print ("Median of 56-62 scores")

print (np.median(scoresHeight56))

print ("Median of 63-69 scores")

print (np.median(scoresHeight63))

print ("Median of 70-76 scores")

print (np.median(scoresHeight70))

print ("Median of 76-83 scores")

print (np.median(scoresHeight76))

print ("Mean of 79-89 scores")

print (np.mean(scoresIQ79))

print ("Mean of 90-100 scores")

print (np.mean(scoresIQ90))

print ("Mean of 101-111 scores")

print (np.mean(scoresIQ101))

print ("Mean of 112-122 scores")

print (np.mean(scoresIQ112))

print ("Median of 79-89 scores")

print (np.median(scoresIQ79))

print ("Median of 90-100 scores")

print (np.median(scoresIQ90))

print ("Median of 101-111 scores")

print (np.median(scoresIQ101))

print ("Median of 112-122 scores")

print (np.median(scoresIQ112))

print ("Mean of 0-0 scores")

print (np.mean(scoresAlcohol0))

print ("Mean of 1-2 scores")

print (np.mean(scoresAlcohol1))

print ("Mean of 3-4 scores")

print (np.mean(scoresAlcohol3))

print ("Mean of 5-7 scores")

print (np.mean(scoresAlcohol5))

print ("Median of 0-0 scores")

print (np.median(scoresAlcohol0))

print ("Median of 1-2 scores")

print (np.median(scoresAlcohol1))

print ("Median of 3-4 scores")

print (np.median(scoresAlcohol3))

print ("Median of 5-7 scores")

print (np.median(scoresAlcohol5))

print ("Mean of 0-0 scores")

print (np.mean(scoresCigs0))

print ("Mean of 1-10 scores")

print (np.mean(scoresCigs1))

print ("Mean of 11-20 scores")

print (np.mean(scoresCigs11))

print ("Mean of 21-40 scores")

print (np.mean(scoresCigs21))

print ("Median of 0-0 scores")

print (np.median(scoresCigs0))

print ("Median of 1-10 scores")

print (np.median(scoresCigs1))

print ("Median of 11-20 scores")

print (np.median(scoresCigs11))

print ("Median of 21-40 scores")

print (np.median(scoresCigs21))

print ("Mean of Inactive scores")

print (np.mean(scoresInactive))

print ("Mean of Active scores")

print (np.mean(scoresActive))

print ("Mean of Very Active scores")

print (np.mean(scoresVeryActive))

print ("Median of Inactive scores")

print (np.median(scoresInactive))

print ("Median of Active scores")

print (np.median(scoresActive))

print ("Median of Very Active scores")

print (np.median(scoresVeryActive))

print (min(unitsCigs))

print (max(unitsCigs))

### KMeans

#\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

from pylab import plt

#Kmeans

age = list(map(float, unitsCigs))

weight = list(map(float, healthScore))

#[age, weight, height, iq, unitsAlcohol, unitsCigs]

kMeansArray = [age, weight]

kmeans = KMeans(n\_clusters=2) # initialization

kmeans.fit(kMeansArray) # actual execution

y\_kmeans = kmeans.predict(kMeansArray)

centers = kmeans.cluster\_centers\_

print(centers)

#plt.scatter(kMeansArray[0], kMeansArray[1])

#++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++

print(len(age))

print(len(weight))

print(len(height))

print(len(iq))

print(len(unitsAlcohol))

print(len(unitsCigs))

print(len(scoresNumberActivity))

print(len(genderNumberScores))

print(len(healthScore))

### Regression Analysis

import matplotlib.pyplot as plt

import numpy as np

import pandas as pd

from sklearn import datasets, linear\_model

tempAge = age

tempWeight = weight

tempHeight = height

tempIq = iq

tempUnitsAlcohol = unitsAlcohol

tempUnitsCigs = unitsCigs

tempScoresNumberActivity = scoresNumberActivity

tempGenderNumberScores = genderNumberScores

for index, item in enumerate(age):

item = (item-min(tempAge))/(max(tempAge)-min(tempAge))

for index, item in enumerate(weight):

item = (item-min(tempWeight))/(max(tempWeight)-min(tempWeight))

for index, item in enumerate(height):

item = (item-min(tempHeight))/(max(tempHeight)-min(tempHeight))

for index, item in enumerate(iq):

item = (item-min(tempIq))/(max(tempIq)-min(tempIq))

for index, item in enumerate(unitsAlcohol):

item = (item-min(tempUnitsAlcohol))/(max(tempUnitsAlcohol)-min(tempUnitsAlcohol))

for index, item in enumerate(unitsCigs):

item = (item-min(tempUnitsCigs))/(max(tempUnitsCigs)-min(tempUnitsCigs))

for index, item in enumerate(tempScoresNumberActivity):

item = (item-min(tempScoresNumberActivity))/(max(tempScoresNumberActivity)-min(tempScoresNumberActivity))

for index, item in enumerate(tempGenderNumberScores):

item = (item-min(tempGenderNumberScores))/(max(tempGenderNumberScores)-min(tempGenderNumberScores))

x = np.array([age, genderNumberScores, weight, height, iq, unitsAlcohol, unitsCigs, scoresNumberActivity])

y = np.array([healthScore])

print(x.shape)

print(y.shape)

x = np.transpose(x)

y = np.transpose(y)

print(x)

print(y)

print(x.shape)

print(y.shape)

predict\_value = x

# Create linear regression object

regr = linear\_model.LinearRegression()

regr.fit(x, y)

predict\_outcome = regr.predict(predict\_value)

result = {}

result['intercept'] = regr.intercept\_

result['coefficient'] = regr.coef\_

result['predicted\_value'] = predict\_outcome

print ("Intercept value " , result['intercept'])

print ("coefficient" , result['coefficient'])

print ("Predicted value: ",result['predicted\_value'])

#Read file in

predictArray = np.genfromtxt('Population.csv', delimiter=',', dtype=None).astype(str)

#Extract variables from raw input

ageP = [predictArray[i][1] for i in range(len(predictArray))]

sexP = [predictArray[i][2] for i in range(len(predictArray))]

weightP = [predictArray[i][3] for i in range(len(predictArray))]

heightP = [predictArray[i][4] for i in range(len(predictArray))]

iqP = [predictArray[i][5] for i in range(len(predictArray))]

unitsAlcoholP = [predictArray[i][6] for i in range(len(predictArray))]

unitsCigsP = [predictArray[i][7] for i in range(len(predictArray))]

activityP = [predictArray[i][8] for i in range(len(predictArray))]

#strip off column labels

del ageP[0]

del sexP[0]

del weightP[0]

del heightP[0]

del iqP[0]

del unitsAlcoholP[0]

del unitsCigsP[0]

del activityP[0]

#print(activityP)

#convert to ints where appropriate

ageP = list(map(int, ageP))

weightP = list(map(int, weightP))

heightP = list(map(int, heightP))

iqP = list(map(int, iqP))

unitsAlcoholP = list(map(int, unitsAlcoholP))

unitsCigsP = list(map(int, unitsCigsP))

genderNumberScoresP = []

scoresNumberActivityP = []

#extract female and male scores

for index, item in enumerate(sexP):

if item == "Female":

genderNumberScoresP.append(0)

elif item == "Male":

genderNumberScoresP.append(1)

#extract activity scores

for index, item in enumerate(activityP):

if item == "Inactive":

scoresNumberActivityP.append(0)

elif item == "Active":

scoresNumberActivityP.append(1)

elif item == "Very Active":

scoresNumberActivityP.append(2)

for index, item in enumerate(ageP):

item = (item-min(tempAge))/(max(tempAge)-min(tempAge))

for index, item in enumerate(weightP):

item = (item-min(tempWeight))/(max(tempWeight)-min(tempWeight))

for index, item in enumerate(heightP):

item = (item-min(tempHeight))/(max(tempHeight)-min(tempHeight))

for index, item in enumerate(iqP):

item = (item-min(tempIq))/(max(tempIq)-min(tempIq))

for index, item in enumerate(unitsAlcoholP):

item = (item-min(tempUnitsAlcohol))/(max(tempUnitsAlcohol)-min(tempUnitsAlcohol))

for index, item in enumerate(unitsCigsP):

item = (item-min(tempUnitsCigs))/(max(tempUnitsCigs)-min(tempUnitsCigs))

for index, item in enumerate(scoresNumberActivityP):

item = (item-min(tempScoresNumberActivity))/(max(tempScoresNumberActivity)-min(tempScoresNumberActivity))

for index, item in enumerate(genderNumberScoresP):

item = (item-min(tempGenderNumberScores))/(max(tempGenderNumberScores)-min(tempGenderNumberScores))

predictData = np.array([ageP, genderNumberScoresP, weightP, heightP, iqP, unitsAlcoholP, unitsCigsP, scoresNumberActivityP])

predictData = np.transpose(predictData)

# Create linear regression object

regrP = linear\_model.LinearRegression()

regrP.fit(x, y)

predict\_outcomeP = regrP.predict(predictData)

resultP = {}

resultP['predicted\_value'] = predict\_outcomeP

print ("Predicted value: ",resultP['predicted\_value'])

print(predictData)

#show\_linear\_line(x,y)

#plt.scatter(centers[:, 0], centers[:, 1], c='black', s=200, alpha=0.5);

#print (age)

#print (sex)

#print (weight)

#print (height)

#print (iq)

#print (unitsAlcohol)

#print (unitsCigs)

#print (activity)

#print (healthScore)

1. University Academic Misconduct Regulations [↑](#footnote-ref-1)
2. Information on exclusions to this rule is available from the Advice Centre at each Campus [↑](#footnote-ref-2)