COMP8410 Individual Project

Data Mining Project on

Life Satisfaction Analysis

DONGRUI LEI
U7176516
2022/5/8

Introduction

This report is aimed to discover a general method that predicts life satisfaction among people who are currently living in Australia and having a paid job. And dig out the major factors that affects life satisfaction the most.

From data collected by "Life in Australia", released by ANU Human Research Ethics Committee, questions in section "A" contains how people evaluate their life satisfaction, section "D" illustrates mental health and section "E" shows the income and employment situation. I would like to organize data from these three sections to show how income and employment situation and mental health level affects people's life satisfaction, among people who have got a paid job.

Exclusive Summary

Life satisfaction is how you evaluate the quality of your life and how happy/satisfied you are with it. High satisfaction suggests that the quality of life, in the population concerned, is good. Low satisfaction marks serious shortcomings of some kind. And mental health plays a great role in increasing life satisfaction. Mental disorders are major cause of non-fatal burden of disease worldwide. (Gigantesco Antonella; Fagnani Corrado; Toccaceli Virgilia; Stazi Maria Antonietta; Lucidi Fabio; Violani Cristiano; Picardi Angelo, 2019). In this word, if people could understand more about how mental health can affect life satisfaction, it would be a great help for the well-being of the entire society.

Life-satisfaction also tends to be higher in more socially equal societies, income wise for example. Some study shows that income of an individual also affects life satisfaction, in countries that are relatively less developed, the correlation becomes even more obvious. (Veenhoven, 1996)At the same time, in Australia, 73% of the working-age population aged 15 to 64 has a paid job. And the fact is that societies with high levels of employment are also richer, more politically stable and healthier. (OECD Jobs, n.d.) In this case, in order to improve life satisfaction, people with a paid job play a great deal of many social surveys.

In this report, I tried 3——technically 4——different methods to build a model in order to predict "Life satisfaction" among the observations in the sample given. At the end, non of them provide me a good enough accuracy. This may due to the size of the training sample. The entire dataset provides me around 3,000 rows, and along with few unavailable observations.NA values for instance. After I pre-processed the data, there are only 1,602 entries totally to use for both training and testing. This is quite limited. And the target factor I chose here is question "A1", which is how people feel about their overall life in Australia, from 1 - 5, "Very satisfied", "Satisfied", "Neither satisfied nor dissatisfied”, “Dissatisfied” and “Very dissatisfied”. The results are distributed with great bias, which means some result has dominantly high percentage and other might be quite lower. This will also affect the accuracy of all these 5 classes while doing classification to predict unseen data.

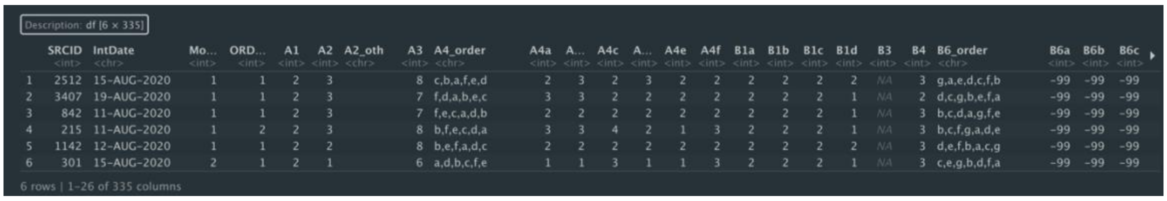
However, I did accomplish some goals described above. I found the most influential attributes that affects the result of life satisfaction, and the other correlation among attributes. This might be helpful for many social problems that are conducting researches on “What should we improve in our society to improve life satisfaction?”. To make a simple conclusion here: Mental health, total hours of overtime works and the level of losing job anxiety are the top three attributes affecting life satisfaction. If some social researchers are focusing on this, my conclusion might be helpful.

If there is some recommendations of this project, I might suggest to have more data collected from people living in Australia currently. This data size might be enough to conduct some conclusion in social science, but it is very limited for machine learning experiments. And try to get people more generally. In this dataset, I’ve found that lots of observations tend to have the same answer for a lot of questions. This will lead the result of machine learning tend to conclude that everyone is the same, classified in one category. Maybe more people having other answers will be helpful.

Data Description

The dataset come from “Life in Australia” survey, released by ANU Human Research Ethics Committee. Here I’m using the original data, “2.ADA.ANUPoll35.CSV.01474”, to complete this project. And this project is conducted by R in R studio. The population this sample is representing is “People living in Australia and having a paid job currently”.

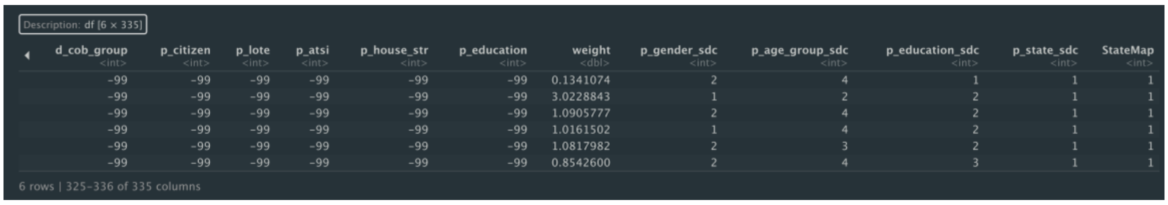
First 6 rows of first 26 attributes and last 11 attributes:



Description: df [6 × 335]

	SRCID	IntDate	Mo...	ORD...	A1	A2	A2_oth	A3	A4_order	A4a	A...	A4c	A...	A4e	A4f	B1a	B1b	B1c	B1d	B3	B4	B6_order	B6a	B6b	B6c
	<int>	<chr>	<int>	<int>	<int>	<int>	<chr>	<int>	<chr>	<int>	<int>	<int>	<int>	<int>	<int>	<int>	<int>	<int>	<int>	<int>	<int>	<chr>	<int>	<int>	<int>
1	2512	15-AUG-2020	1	1	2	3		8	c,b,a,f,e,d	2	3	2	3	2	2	2	2	2	2	NA	3	g,a,e,d,c,f,b	-99	-99	-99
2	3407	19-AUG-2020	1	1	2	3		7	f,d,a,b,e,c	3	3	2	2	2	2	2	2	2	1	NA	2	d,c,g,b,e,f,a	-99	-99	-99
3	842	11-AUG-2020	1	1	2	3		7	f,e,c,a,d,b	2	2	2	2	2	2	2	2	2	1	NA	3	b,c,d,a,g,f,e	-99	-99	-99
4	215	11-AUG-2020	1	2	2	3		8	b,f,e,c,d,a	3	3	4	2	1	3	2	2	2	1	NA	3	b,c,f,g,a,d,e	-99	-99	-99
5	1142	12-AUG-2020	1	1	2	2		8	b,e,f,a,d,c	2	2	2	2	2	2	2	2	2	2	NA	3	d,e,f,b,a,c,g	-99	-99	-99
6	301	15-AUG-2020	2	1	2	1		6	a,d,b,c,f,e	1	1	3	1	1	3	2	2	2	1	NA	3	c,e,g,b,d,f,a	-99	-99	-99

6 rows | 1-26 of 335 columns

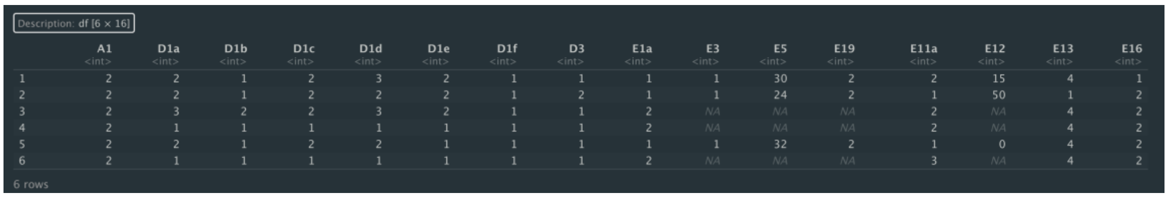


Description: df [6 × 335]

d_cob_group	p_citizen	p_lote	p_atsi	p_house_str	p_education	weight	p_gender_sdc	p_age_group_sdc	p_education_sdc	p_state_sdc	StateMap
<int>	<int>	<int>	<int>	<int>	<int>	<dbl>	<int>	<int>	<int>	<int>	<int>
-99	-99	-99	-99	-99	-99	0.1341074	2	4	1	1	1
-99	-99	-99	-99	-99	-99	3.0228843	1	2	2	1	1
-99	-99	-99	-99	-99	-99	1.0905777	2	4	2	1	1
-99	-99	-99	-99	-99	-99	1.0161502	1	4	2	1	1
-99	-99	-99	-99	-99	-99	1.0817982	2	3	2	1	1
-99	-99	-99	-99	-99	-99	0.8542600	2	4	3	1	1

6 rows | 325-336 of 335 columns

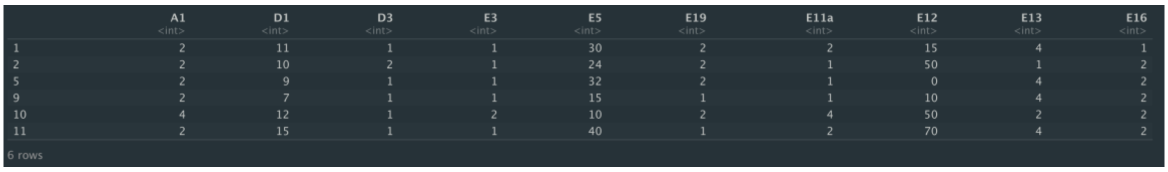
Totally, there are 336 columns representing answer of each question, and 3,061 rows representing each person answering the questions. After examine the original data and the questionnaire of this survey, columns I need are restricted. In order to complete my goal, I request answers from question “A1” in section A as the target factors for classification and prediction. And answers from “D1, D3” in section D, to measure mental health level as well. Finally I need all the answers from “E1, E3, E5, E19, E11a, E12, E13 and E16” to represent attributes of “Income and Employment”.


Description: df [6 × 16]

	A1	D1a	D1b	D1c	D1d	D1e	D1f	D3	E1a	E3	E5	E19	E11a	E12	E13	E16
	<int>	<int>	<int>	<int>	<int>	<int>	<int>	<int>	<int>	<int>	<int>	<int>	<int>	<int>	<int>	<int>
1	2	2	1	2	3	2	1	1	1	1	30	2	2	15	4	1
2	2	2	1	2	2	2	1	2	1	1	24	2	1	50	1	2
3	2	3	2	2	3	2	1	1	2	NA	NA	NA	2	NA	4	2
4	2	1	1	1	1	1	1	1	2	NA	NA	NA	2	NA	4	2
5	2	2	1	2	2	1	1	1	1	1	32	2	1	0	4	2
6	2	1	1	1	1	1	1	1	2	NA	NA	NA	3	NA	4	2

6 rows

Like this, all attributes I need are shown above (first 6 rows). Now I need to fix some problems. Since this report mainly focus on people with a paid job, there is going to be a data filter on the dataset shown above. In column “E1a”, value “1” represents observations who has a paid job. So only “1” is the eligible answer for question E1a, and I pick them out. Another thing to do is gathering all answers from D1 (D1a-f) into one column. I basically add all answer together, to represent the answer of D1(further explanation later). The result is shown below.



	A1	D1	D3	E3	E5	E19	E11a	E12	E13	E16
	<int>	<int>	<int>	<int>	<int>	<int>	<int>	<int>	<int>	<int>
1	2	11	1	1	30	2	2	15	4	1
2	2	10	2	1	24	2	1	50	1	2
5	2	9	1	1	32	2	1	0	4	2
9	2	7	1	1	15	1	1	10	4	2
10	4	12	1	2	10	2	4	50	2	2
11	2	15	1	1	40	1	2	70	4	2

6 rows

After those processes, there are totally 1,602 rows left and 10 columns containing 1 target factor and 9 attributes.

Here need a brief description and statistical summary of this data that contains attributes and target factors of people who has a paid job.

A1: Categorical variable of life satisfaction,

Very satisfied	1
Satisfied	2
Neither satisfied nor dissatisfied	3
Dissatisfied	4
Very dissatisfied	5

D1: Since in D1, there are 6 total question asking about how an individual feels mentally, including "How often do you feel (Nervous? Hopeless? Restless or fidgety? That everything was an effort? So sad that nothing could cheer you up? Worthless?) in the past 7 days, each from

1(Non of the time) to 5 (All of the time). So I would like to generalize them and combine points from D1a - D1f, to represent how much the person answering the question feels unwell mentally. 5(None of the time) - 30 (All of the time). Discrete numeric data, restricted from 5 - 30.

D3: Measurement of how often the person feels lonely in the past week, discrete numeric data, scale through 1(Rarely or none of the time less then one day), 2(Some or a little of the time 1-2days), 3(Occasionally or a moderate amount of time, 3-4 days), 4(Most or all of the time (5-7 days).

E3: Indicates what kind of paid job does the person have as a second job, discrete numeric data:

An employee	1
Self-employed / business owner	2
Working for your own family's business	3

E5: How many hours worked last week in main job, including any paid or unpaid overtime. Continuous numeric data scale 0-168.

E19: Have this person received $1500 JobKeeper payment from the employer. Discrete numeric data, 1(Yes) or 2(No).

E11a: How does this person feel about his or her current income situation. Discrete numeric data,

Living comfortably on present income	1
Coping on present income	2
Finding it difficult on present income	3
Finding it very difficult on present income	4

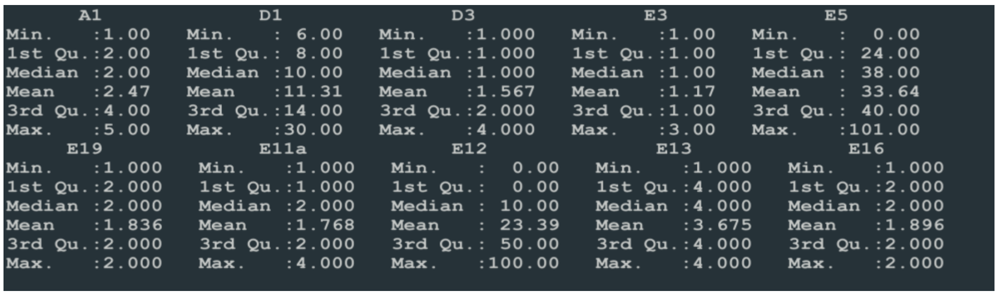
E12: What does this person think is the chance in percentage that him or her will lose the job during the next 12 months? Continuous numeric data, 0 - 100.

E13: In the last 3 months have this person not been able to pay the mortgage or rent on time because of a shortage of money? Discrete numeric data

Yes, mortgage	1
Yes, rent	2
Yes, both	3
No, neither rent or mortgage	4

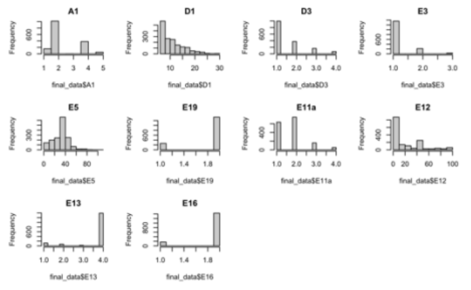
E16: Does this person help or supervise another member in the household who has a long-term health condition or disability, or is elderly? Discrete numeric data, 1(Yes) and 2(No).

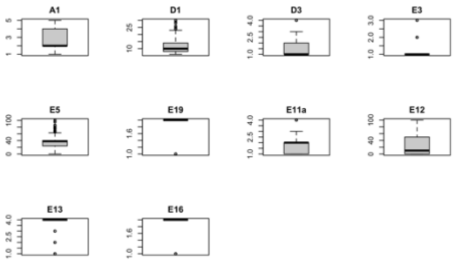
And all of these questions have possible answers like -99 and -98 represent "Refuse to answer" and "Don't know". I'd like to get rid of them since they don't provide any helpful information that



```
      A1             D1              D3              E3              E5
Min.   :1.00   Min.   : 6.00   Min.   :1.000   Min.   :1.00   Min.   :  0.00
1st Qu.:2.00   1st Qu.: 8.00   1st Qu.:1.000   1st Qu.:1.00   1st Qu.: 24.00
Median :2.00   Median :10.00   Median :1.000   Median :1.00   Median : 38.00
Mean   :2.47   Mean   :11.31   Mean   :1.567   Mean   :1.17   Mean   : 33.64
3rd Qu.:4.00   3rd Qu.:14.00   3rd Qu.:2.000   3rd Qu.:1.00   3rd Qu.: 40.00
Max.   :5.00   Max.   :30.00   Max.   :4.000   Max.   :3.00   Max.   :101.00
     E19             E11a             E12              E13             E16
Min.   :1.000   Min.   :1.000   Min.   :  0.00   Min.   :1.000   Min.   :1.000
1st Qu.:2.000   1st Qu.:1.000   1st Qu.:  0.00   1st Qu.:4.000   1st Qu.:2.000
Median :2.000   Median :2.000   Median : 10.00   Median :4.000   Median :2.000
Mean   :1.836   Mean   :1.768   Mean   : 23.39   Mean   :3.675   Mean   :1.896
3rd Qu.:2.000   3rd Qu.:2.000   3rd Qu.: 50.00   3rd Qu.:4.000   3rd Qu.:2.000
Max.   :2.000   Max.   :4.000   Max.   :100.00   Max.   :4.000   Max.   :2.000
```

I need. After wrangling the redundent data, here is the simple summary of the dataset we have here.





**Note: all percentage shown below are all calculated by R studio. The processes are shown in appendices.

Now let’s see how these columns are distributed:
As we could tell from these diagrams, **most people are quite satisfied with their current life**. Over half of the participants are choosing 1 or 2 in A1, and over 87% chose 1 or 2 in E11a. At the same time, according to E12 and E13, we can tell that people are not so worrying about losing their jobs

Another thing worth notice is that the mental unwellness of these people is quite small, mean value is 11.31 and 1.57 in D1 and D3. According to both histograms of these two attributes, they are right skewed.

Though there are some values may need further investigation. T**here are still around 30% of people think they lose their job in the next coming year,** which is not a figure should be overlooked. And there are still some people dealing with these problems since nearly 14.5% are having high score (larger than 2) in D3, according to the box-plot.

The overtime hours may be the cause of problems above, because based on what E3, E5 and E19, over 85% people are having a paid job as an employee and 84% of them are **not reviving** $1500 JobKeeper payment from the employer, and **33.7 hours overtime** at work is the mean.

These data are collected from ANU Human Research Ethics Committee (2014/241), “Life in Australia Wave 41” conducted in August 2020. **After the pre-processing the data, the**

population of this report should be: people living in Australia, and currently having a paid job.

After this, I would like to initiate some machine learning method to construct a suitable model to predict new observations with similar attributes.

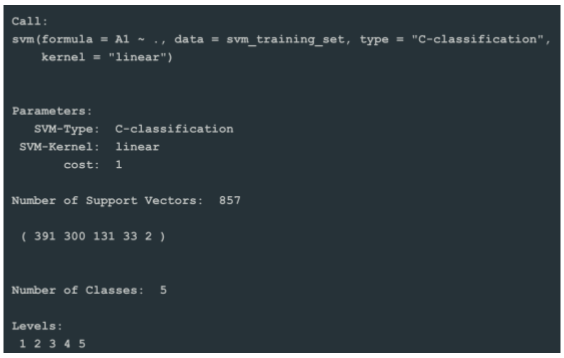
Machine Learning Method Description

Method 1:

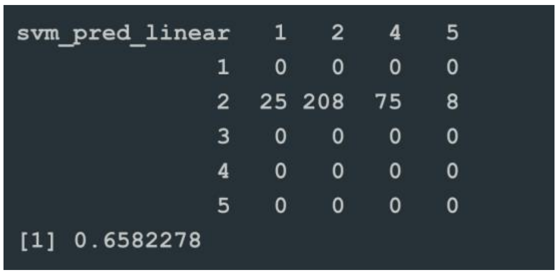
SVM (calculated by R) is used to classify what kind of person would be more possible to choose 1, 2, 3 or 4 in question A1. And there will be some explanation of the parameter tuning and the limitation of this method. With this method, I would like to construct a model that could correctly classify the unseen data into those 5 categories.

SVM mainly transform a nonlinear mapping into a higher dimension to classify different classes with hyperplanes. In this case we have here, the attributes are representing people with lots of overlaps. So, classification problem cannot be solved by simply drawing some soft margins, otherwise it'll be a relatively higher bias.

I split data into training and testing sets, with the ratio of (80% training-20% testing). And then fit the training set into a basic SVM model with **linear kennel**. Then fit the trained model with unseen data, to show the accuracy of the prediction. Here is the summary of this model and result of confusion matrix:

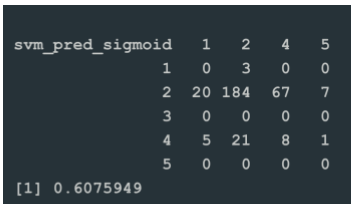


```
Call:
svm(formula = A1 ~ ., data = svm_training_set, type = "C-classification",
    kernel = "linear")


Parameters:
   SVM-Type:  C-classification
 SVM-Kernel:  linear
       cost:  1

Number of Support Vectors:  857

 ( 391 300 131 33 2 )


Number of Classes:  5

Levels:
 1 2 3 4 5
```

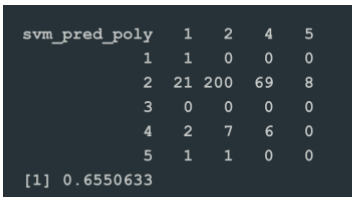


```
svm_pred_linear   1   2   4   5
              1   0   0   0   0
              2  25 208  75   8
              3   0   0   0   0
              4   0   0   0   0
              5   0   0   0   0
[1] 0.6582278
```

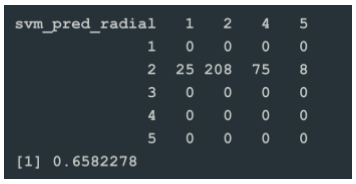
From this model I trained here, the result is quite unsatisfied. Only around 65% data has been correctly predicted, and all of them are in answer "2". Means that other answers are not trained well. As a matter of fact, since in the training set there are too many observations scoring "2" as the answer of A1(over 63%), the training sample of other results is too limited. In this case, all new data would be look very similar to those results in “2”. Let’s try other activation function to see if the performance is going to be better.



```
svm_pred_sigmoid   1   2   4   5
               1   0   3   0   0
               2  20 184  67   7
               3   0   0   0   0
               4   5  21   8   1
               5   0   0   0   0
[1] 0.6075949
```



```
svm_pred_poly   1   2   4   5
            1   1   0   0   0
            2  21 200  69   8
            3   0   0   0   0
            4   2   7   6   0
            5   1   1   0   0
[1] 0.6550633
```



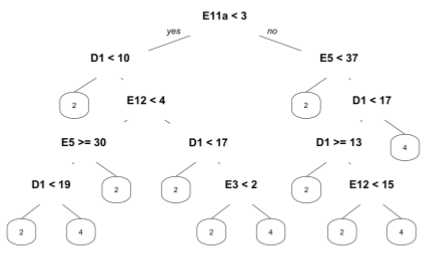
```
svm_pred_radial   1   2   4   5
              1   0   0   0   0
              2  25 208  75   8
              3   0   0   0   0
              4   0   0   0   0
              5   0   0   0   0
[1] 0.6582278
```

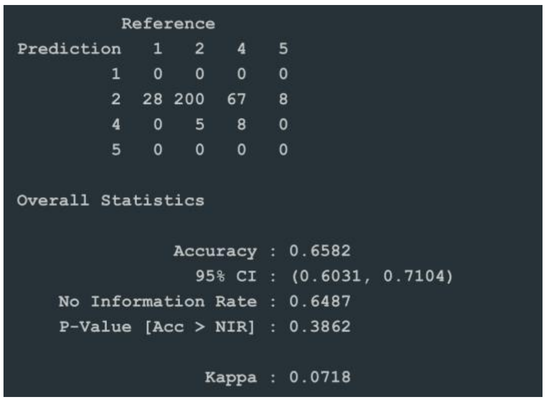
“Radial” and “Linear” kernel seem to have a slightly better performance than other ones. But still not good enough. Since we have more than 2 attribute here, so it could be quite difficult to plot a 2D or 3D diagram to illustrate the result of this SVM classifier. And due to the limitation of this training data, this is the best SVM could do in this situation.

Method 2:

Next, I’ll try decision tree to do this classification. A decision tree is simply a series of sequential decisions made to reach a specific result. Here’s an illustration of a decision tree in action. With this method, I would like to construct a model that could correctly classify the unseen data into those 5 categories. And try to analyze the most influential attributes in this experiment that affects life satisfaction, use Random Forest.

Same, split the data into training and testing sets with the same ratio (80% training-20% testing). And construct a decision tree use R studio. With the training set we have here, this is the diagram showing the tree, and summary of the prediction result.

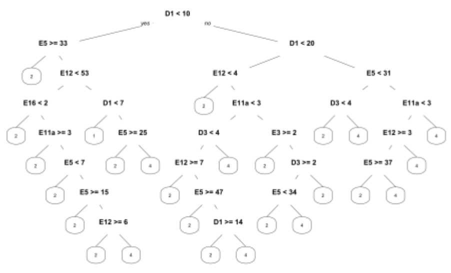


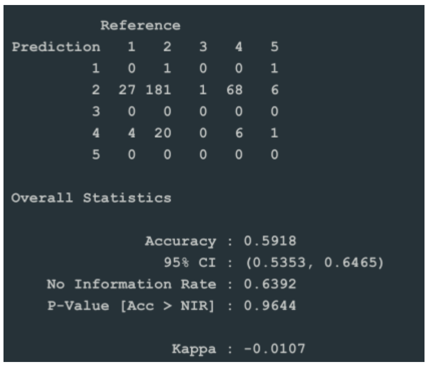


```
          Reference
Prediction   1   2   4   5
         1   0   0   0   0
         2  28 200  67   8
         4   0   5   8   0
         5   0   0   0   0

Overall Statistics

               Accuracy : 0.6582
                 95% CI : (0.6031, 0.7104)
    No Information Rate : 0.6487
    P-Value [Acc > NIR] : 0.3862

                  Kappa : 0.0718
```

After this, we could tell from this diagram, the decision process is 5 level deep. All these result in a figure around 2 and 4, which means lots of training observations tend to vote for “Satisfied” and “Dissatisfied.”. Unfortunately, the result is not better. 65.8% accuracy is definitely not a good figure to predict unseen data. As we can see from the confusion matrix here, there are some more options like “4” are correctly predicted, which is better than what SVM did earlier. But there are still many incorrect predictions made.

What if the tree is deeper? Let’s tune the parameter “maxdepth” into a larger value and see the result.

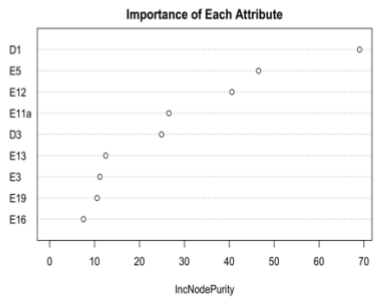




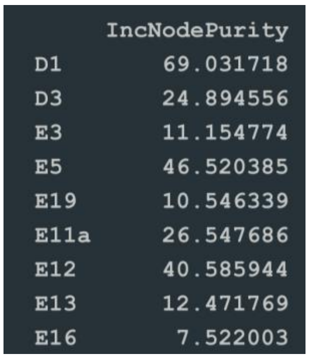
```
          Reference
Prediction   1   2   3   4   5
         1   0   1   0   0   1
         2  27 181   1  68   6
         3   0   0   0   0   0
         4   4  20   0   6   1
         5   0   0   0   0   0

Overall Statistics

               Accuracy : 0.5918
                 95% CI : (0.5353, 0.6465)
    No Information Rate : 0.6392
    P-Value [Acc > NIR] : 0.9644

                  Kappa : -0.0107
```

Here is the diagram of “maxdepth” is 8 and the tree gets deeper. But the prediction accuracy is getting lower. Since “5” is a relatively low level of a decision tree, if we go shallower than 5 then the tree would have too small number of branches and nodes. I might just say this decision tree with 5 levels is the best decision tree I can do.

However, there is more a decision tree could do. This is where the **Random Forest** algorithm comes into the picture. Random Forest is a tree-based machine learning algorithm that leverages the power of multiple decision trees for making decisions. As the name suggests, it is a “Forest” of trees. It is a forest of randomly created decision trees. Each node in the decision tree works on a random subset of features to calculate the output. The random forest then combines the output of individual decision trees to generate the final output. Random forest can do one thing that decision tree cannot do, that is show the “importance”, or call it “influence”, each attribute holds. Like this shown below:



:



```
     IncNodePurity
D1       69.031718
D3       24.894556
E3       11.154774
E5       46.520385
E19      10.546339
E11a     26.547686
E12      40.585944
E13      12.471769
E16       7.522003
```

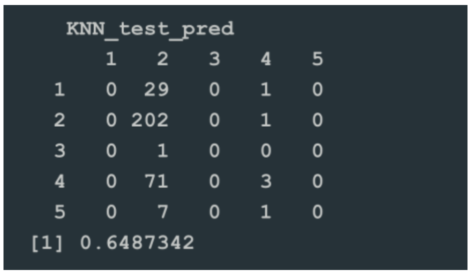
The importance chart shows the most influential attributes are **“E5”, “D1” and “E12”.** Tells us that overtime working hours, mental health and how much a person is afraid to lose the job really affect people on life satisfaction. And “D1” is leading the scoring board way further than other attributes. “E11a” and “D3”, representing how people feel about their current income situation and how often they feel lonely, are also quite important.

Method 3:

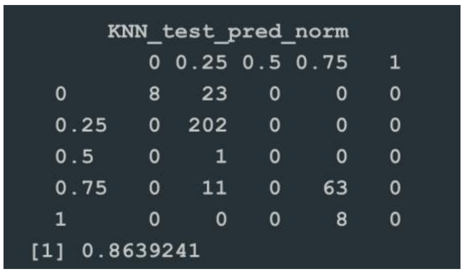
K-NN(K-Nearest Neighbors) is a clustering algorithm used to find features in data that are related in natural or hard to understand ways. K-NN is great for finding ‘groups ’in data and classifying them. Use K-NN could possibly increase the accuracy of the prediction. With this method, I would like to construct a model that could correctly classify the unseen data into those 5 categories. And try to do more on data pre-processing in order to increase the predicting accuracy.

Same, split the data into training and testing sets with the same ratio (80% training-20% testing). And then fit the training set into KNN model in R studio.

Here I will try something different, normalization. Due to the limitation of this dataset, there are some attributes affecting the prediction result more than others, I’ll try to normalize all attributes into range of [0-1], in order to make sure every attribute is having the same weight. After doing this on both training and testing set, the prediction result is show here:



```
   KNN_test_pred
      1   2   3   4   5
  1   0  29   0   1   0
  2   0 202   0   1   0
  3   0   1   0   0   0
  4   0  71   0   3   0
  5   0   7   0   1   0
[1] 0.6487342
```



```
          KNN_test_pred_norm
           0 0.25 0.5 0.75   1
  0        8   23   0    0   0
  0.25     0  202   0    0   0
  0.5      0    1   0    0   0
  0.75     0   11   0   63   0
  1        0    0   0    8   0
[1] 0.8639241
```

The comparison is obvious. Having the normalization dramatically increased the accuracy of my model. But at the same time, I’m losing some important information on this attributes that are affecting the result with heavier weight.

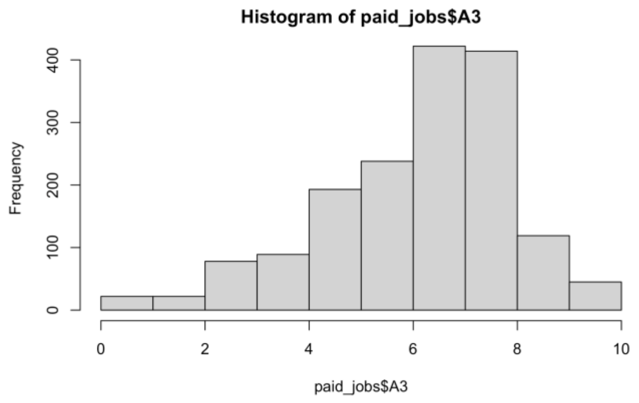
Results and Conclusion

From these methods applied above, I have tried 3 methods to predict results of question “A1”, and one more method trying to understand the most influential attributes that affect the result. I got the best classification result with KNN model, which gives me **86% accuracy** on the test set. In this model, answers like “Very satisfied “and “Dissatisfied” are correctly predicted more than other methods. Normalization plays a great role in it, maybe other methods are having better results with normalization as well. The other methods like SVM and Decision Tree, can provide only

65% accuracy. Changing the kernel in SVM and the tree depth of Decision Tree is not very helpful. Maybe further parameter tuning is needed to fix those problems.

However, through Random Forest, **I found the most influential attributes**: Overtime working hours, mental health and how much a person is afraid to lose the job. This conclusion is quite important for many social researchers trying to figure out method to improve life satisfaction among their society. New laws focusing on overtime work and unemployment, might come up to stage. More organizations consist of psychologists and mental health experts may show up to fix problems for people who feel lonely all the time or having large pressure dealing with their emotions. This conclusion is only providing a general direction, more specific research under these aspects could help the society getting better.

To get better performance on this project, I need more data. Maybe 5 to 10 times larger is required. This will provide me more general training cases with more than one kind of result of question "A1". Another way is to use another question as the target factor, "A3". Since "A3" is scaling from 0-10, representing life satisfaction as well, and the distribution is more uniform, relatively. Distribution histogram of "A3" is down here.



Use this as the target factor might give us a better performance while training a machine learning

model.

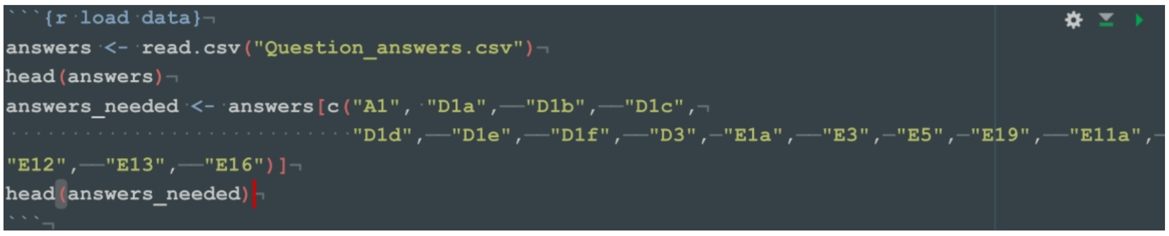
Bibliography

Gigantesco Antonella; Fagnani Corrado; Toccaceli Virgilia; Stazi Maria Antonietta; Lucidi Fabio; Violani Cristiano; Picardi Angelo. (2019). The Relationship Between Satisfaction With Life and Depression Symptoms by Gender. *rontiers in Psychiatry*.

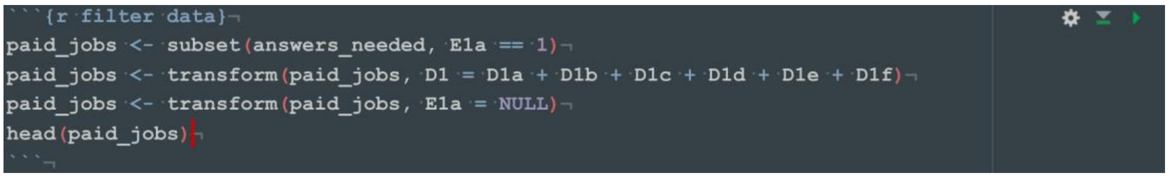
Veenhoven, R. (1996). *The study of life-satisfaction.* Eötvös University Press.

OECD Jobs. (n.d.). Retrieved from OECD Better Llife Index: https://www.oecdbetterlifeindex.org/topics/jobs/

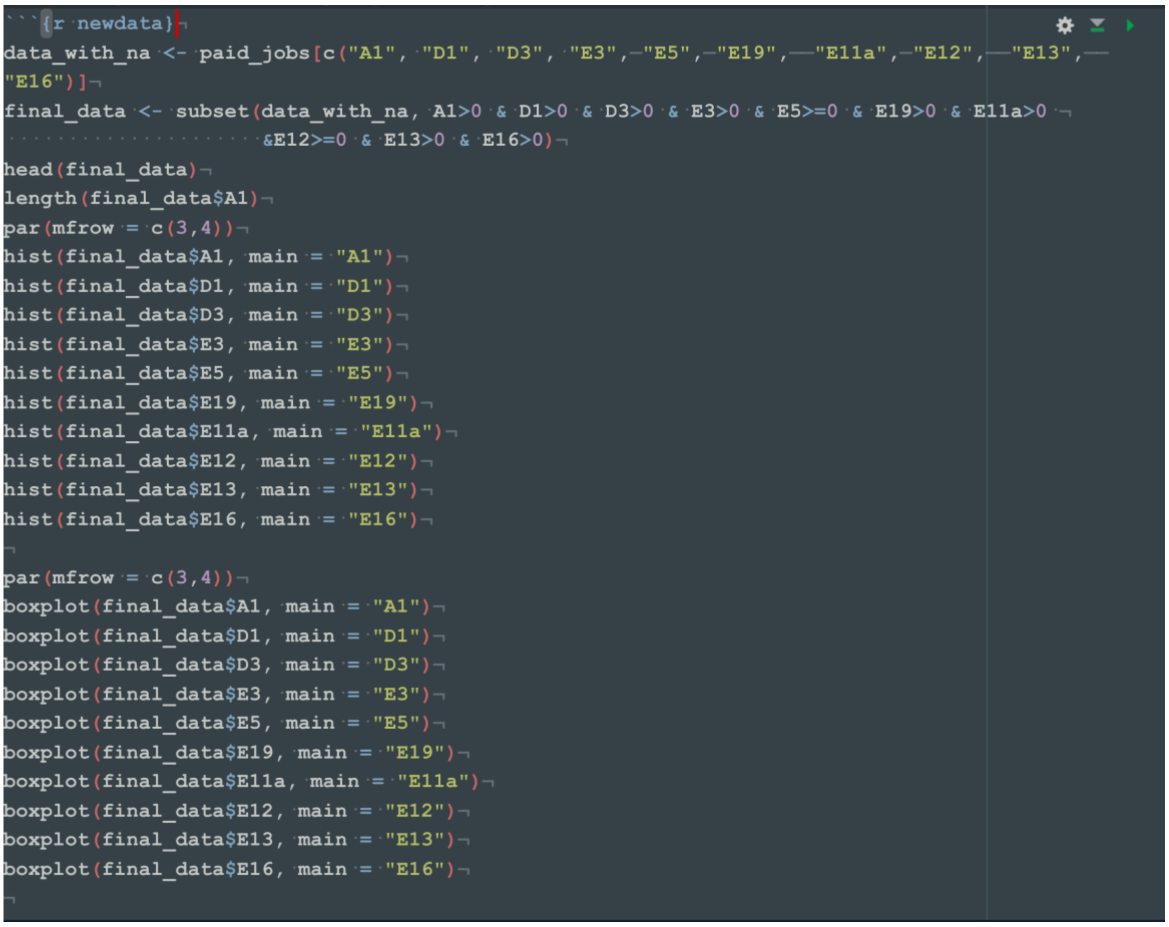
Appendices:



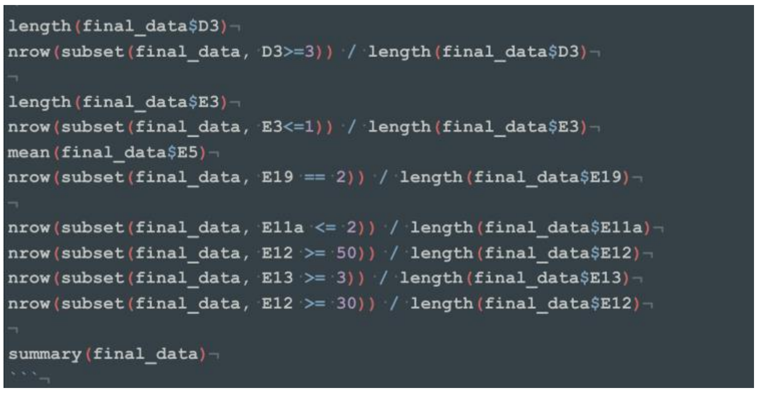
~~~
```{r load data}
answers <- read.csv("Question_answers.csv")
head(answers)
answers_needed <- answers[c("A1", "D1a", "D1b", "D1c",
                            "D1d", "D1e", "D1f", "D3", "E1a", "E3", "E5", "E19", "E11a",
"E12", "E13", "E16")]
head(answers_needed)
```
~~~



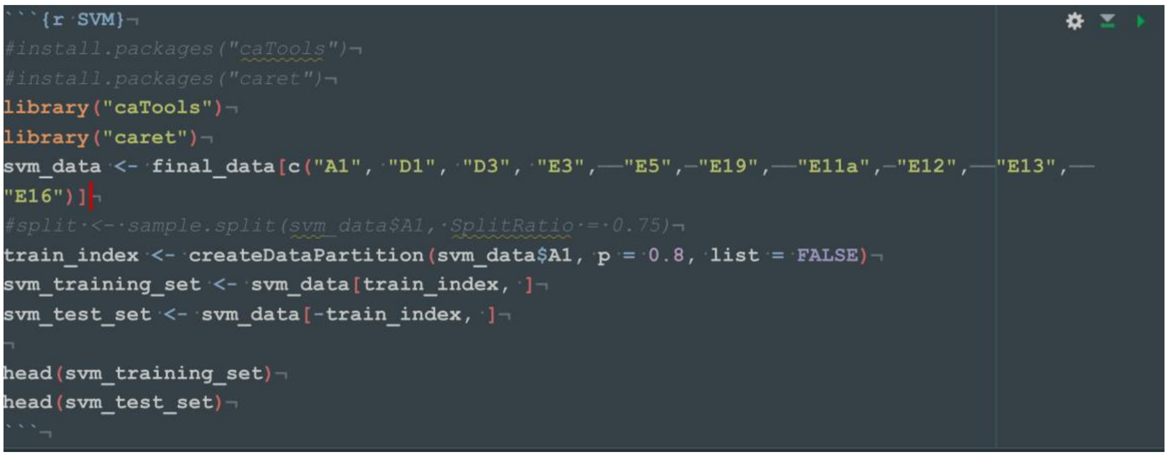
~~~
```{r filter data}
paid_jobs <- subset(answers_needed, E1a == 1)
paid_jobs <- transform(paid_jobs, D1 = D1a + D1b + D1c + D1d + D1e + D1f)
paid_jobs <- transform(paid_jobs, E1a = NULL)
head(paid_jobs)
```
~~~



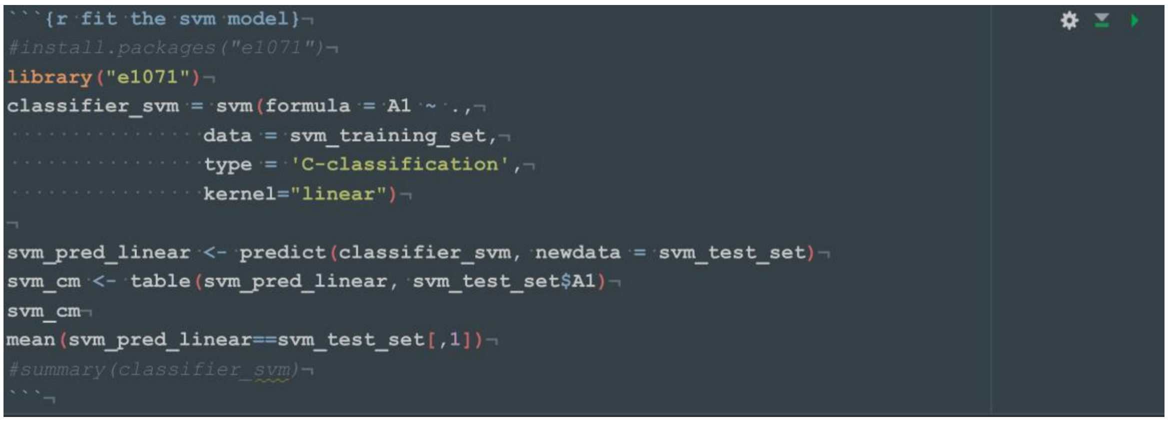
~~~
```{r newdata}
data_with_na <- paid_jobs[c("A1", "D1", "D3", "E3", "E5", "E19", "E11a", "E12", "E13",
"E16")]
final_data <- subset(data_with_na, A1>0 & D1>0 & D3>0 & E3>0 & E5>=0 & E19>0 & E11a>0
                     &E12>=0 & E13>0 & E16>0)
head(final_data)
length(final_data$A1)
par(mfrow = c(3,4))
hist(final_data$A1, main = "A1")
hist(final_data$D1, main = "D1")
hist(final_data$D3, main = "D3")
hist(final_data$E3, main = "E3")
hist(final_data$E5, main = "E5")
hist(final_data$E19, main = "E19")
hist(final_data$E11a, main = "E11a")
hist(final_data$E12, main = "E12")
hist(final_data$E13, main = "E13")
hist(final_data$E16, main = "E16")

par(mfrow = c(3,4))
boxplot(final_data$A1, main = "A1")
boxplot(final_data$D1, main = "D1")
boxplot(final_data$D3, main = "D3")
boxplot(final_data$E3, main = "E3")
boxplot(final_data$E5, main = "E5")
boxplot(final_data$E19, main = "E19")
boxplot(final_data$E11a, main = "E11a")
boxplot(final_data$E12, main = "E12")
boxplot(final_data$E13, main = "E13")
boxplot(final_data$E16, main = "E16")

~~~



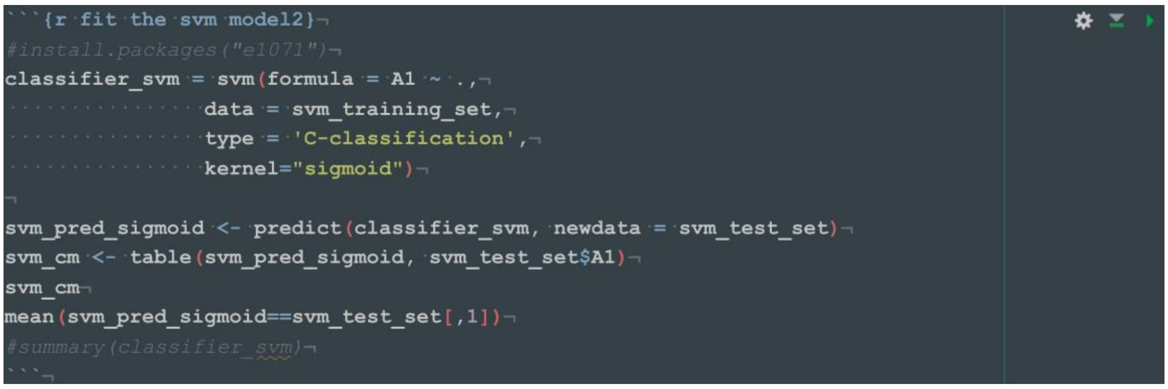
```
length(final_data$D3)
nrow(subset(final_data, D3>=3)) / length(final_data$D3)

length(final_data$E3)
nrow(subset(final_data, E3<=1)) / length(final_data$E3)
mean(final_data$E5)
nrow(subset(final_data, E19 == 2)) / length(final_data$E19)

nrow(subset(final_data, E11a <= 2)) / length(final_data$E11a)
nrow(subset(final_data, E12 >= 50)) / length(final_data$E12)
nrow(subset(final_data, E13 >= 3)) / length(final_data$E13)
nrow(subset(final_data, E12 >= 30)) / length(final_data$E12)

summary(final_data)
```
```
```



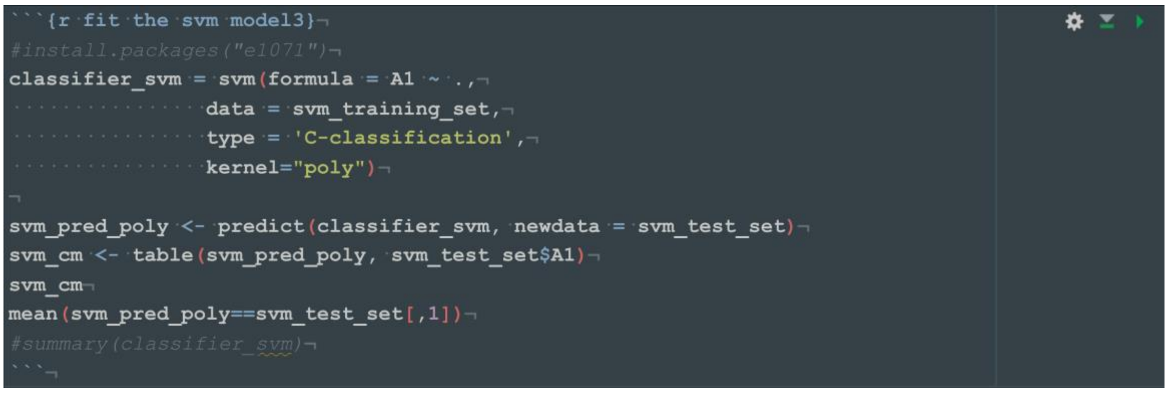
~~~
```{r SVM}
#install.packages("caTools")
#install.packages("caret")
library("caTools")
library("caret")
svm_data <- final_data[c("A1", "D1", "D3", "E3",	"E5",	"E19",	"E11a",	"E12",	"E13",	
"E16")]
#split <- sample.split(svm_data$A1, SplitRatio = 0.75)
train_index <- createDataPartition(svm_data$A1, p = 0.8, list = FALSE)
svm_training_set <- svm_data[train_index, ]
svm_test_set <- svm_data[-train_index, ]

head(svm_training_set)
head(svm_test_set)
```
~~~



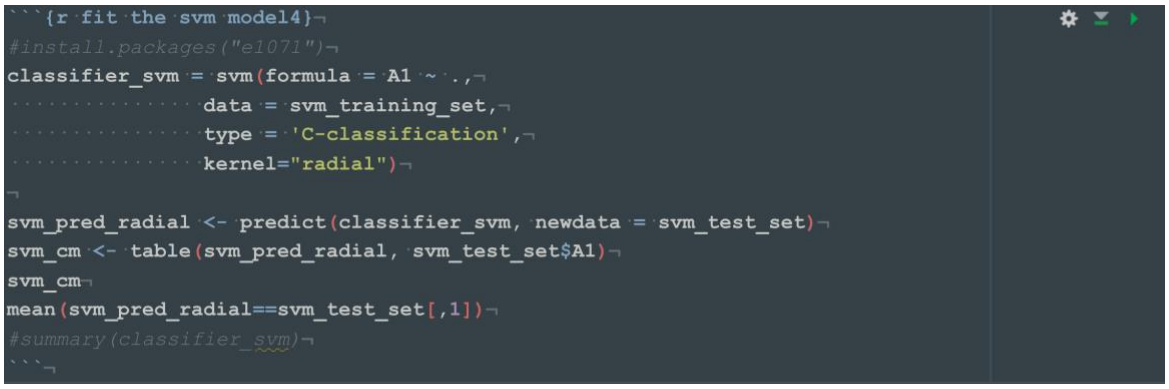
~~~
```{r fit the svm model}
#install.packages("e1071")
library("e1071")
classifier_svm = svm(formula = A1 ~ .,
                data = svm_training_set,
                type = 'C-classification',
                kernel="linear")

svm_pred_linear <- predict(classifier_svm, newdata = svm_test_set)
svm_cm <- table(svm_pred_linear, svm_test_set$A1)
svm_cm
mean(svm_pred_linear==svm_test_set[,1])
#summary(classifier_svm)
```
~~~

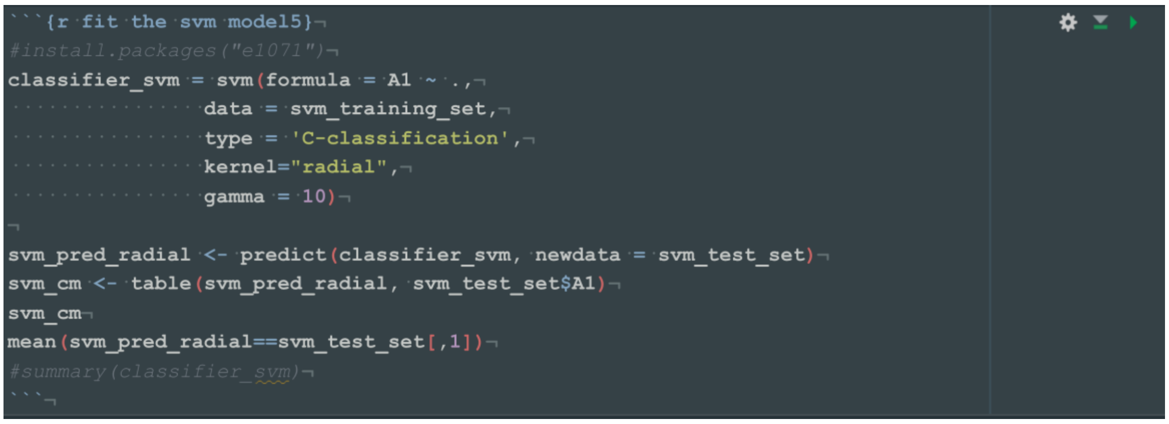
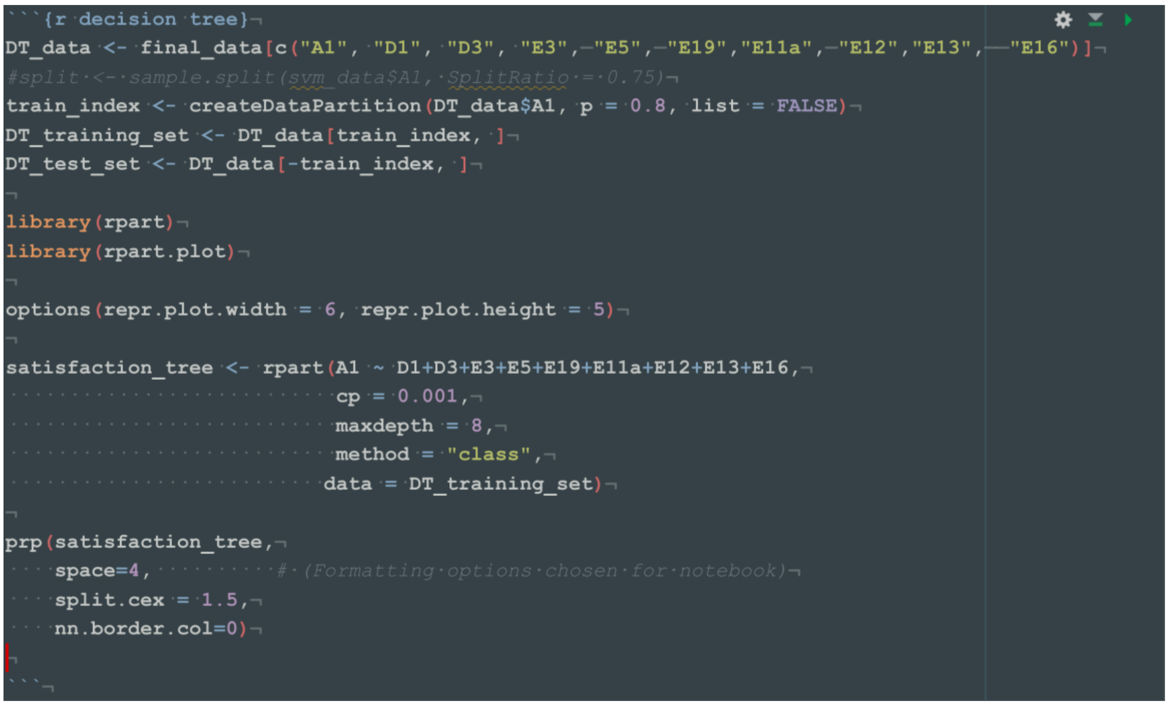


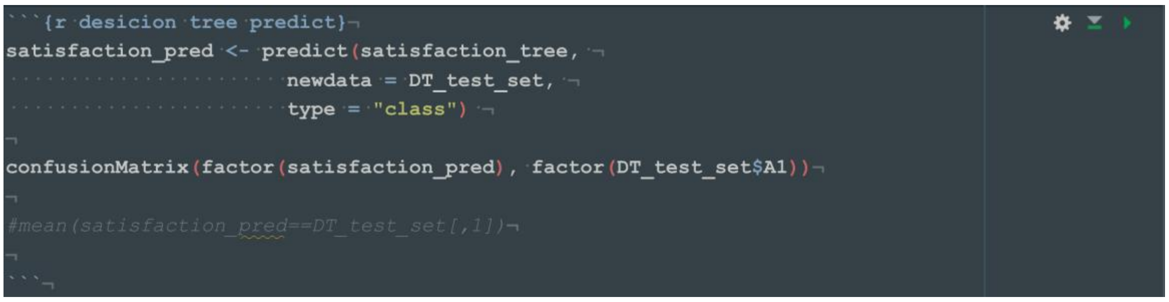
~~~
```{r fit the svm model2}
#install.packages("e1071")
classifier_svm = svm(formula = A1 ~ .,
                data = svm_training_set,
                type = 'C-classification',
                kernel="sigmoid")

svm_pred_sigmoid <- predict(classifier_svm, newdata = svm_test_set)
svm_cm <- table(svm_pred_sigmoid, svm_test_set$A1)
svm_cm
mean(svm_pred_sigmoid==svm_test_set[,1])
#summary(classifier_svm)
```
~~~



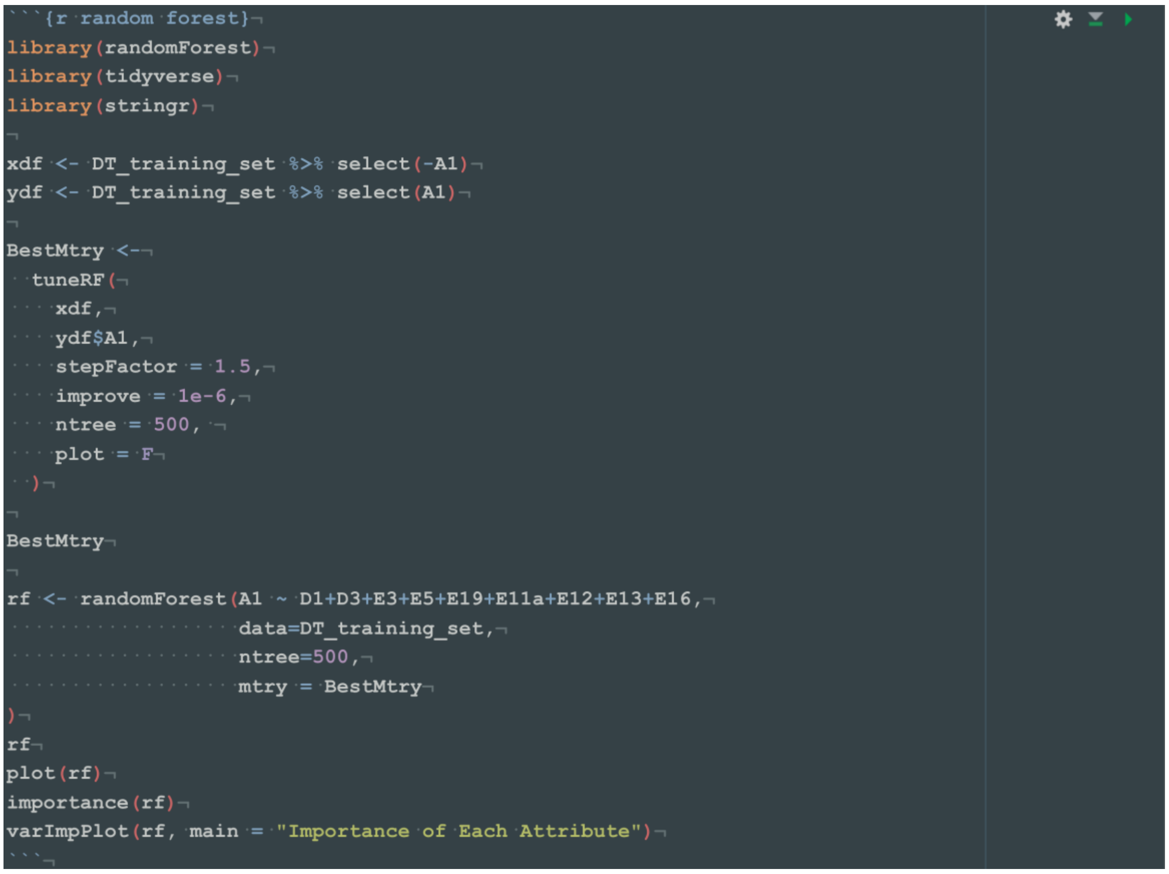
~~~
```{r fit the svm model3}
#install.packages("e1071")
classifier_svm = svm(formula = A1 ~ .,
                data = svm_training_set,
                type = 'C-classification',
                kernel="poly")

svm_pred_poly <- predict(classifier_svm, newdata = svm_test_set)
svm_cm <- table(svm_pred_poly, svm_test_set$A1)
svm_cm
mean(svm_pred_poly==svm_test_set[,1])
#summary(classifier_svm)
```
~~~



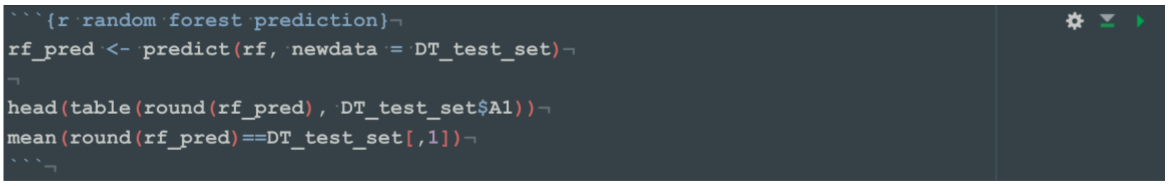
~~~r
```{r fit the svm model4}
#install.packages("e1071")
classifier_svm = svm(formula = A1 ~ .,
                data = svm_training_set,
                type = 'C-classification',
                kernel="radial")

svm_pred_radial <- predict(classifier_svm, newdata = svm_test_set)
svm_cm <- table(svm_pred_radial, svm_test_set$A1)
svm_cm
mean(svm_pred_radial==svm_test_set[,1])
#summary(classifier_svm)
~~~
~~~



```r
```{r fit the svm model5}
#install.packages("e1071")
classifier_svm = svm(formula = A1 ~ .,
                data = svm_training_set,
                type = 'C-classification',
                kernel="radial",
                gamma = 10)

svm_pred_radial <- predict(classifier_svm, newdata = svm_test_set)
svm_cm <- table(svm_pred_radial, svm_test_set$A1)
svm_cm
mean(svm_pred_radial==svm_test_set[,1])
#summary(classifier_svm)
~~~
~~~



```r
```{r decision tree}
DT_data <- final_data[c("A1", "D1", "D3", "E3",	"E5",	"E19","E11a",	"E12","E13",	"E16")]
#split <- sample.split(svm_data$A1, SplitRatio = 0.75)
train_index <- createDataPartition(DT_data$A1, p = 0.8, list = FALSE)
DT_training_set <- DT_data[train_index, ]
DT_test_set <- DT_data[-train_index, ]

library(rpart)
library(rpart.plot)

options(repr.plot.width = 6, repr.plot.height = 5)

satisfaction_tree <- rpart(A1 ~ D1+D3+E3+E5+E19+E11a+E12+E13+E16,
                           cp = 0.001,
                           maxdepth = 8,
                           method = "class",
                          data = DT_training_set)

prp(satisfaction_tree,
    space=4,          # (Formatting options chosen for notebook)
    split.cex = 1.5,
    nn.border.col=0)

~~~
```
```



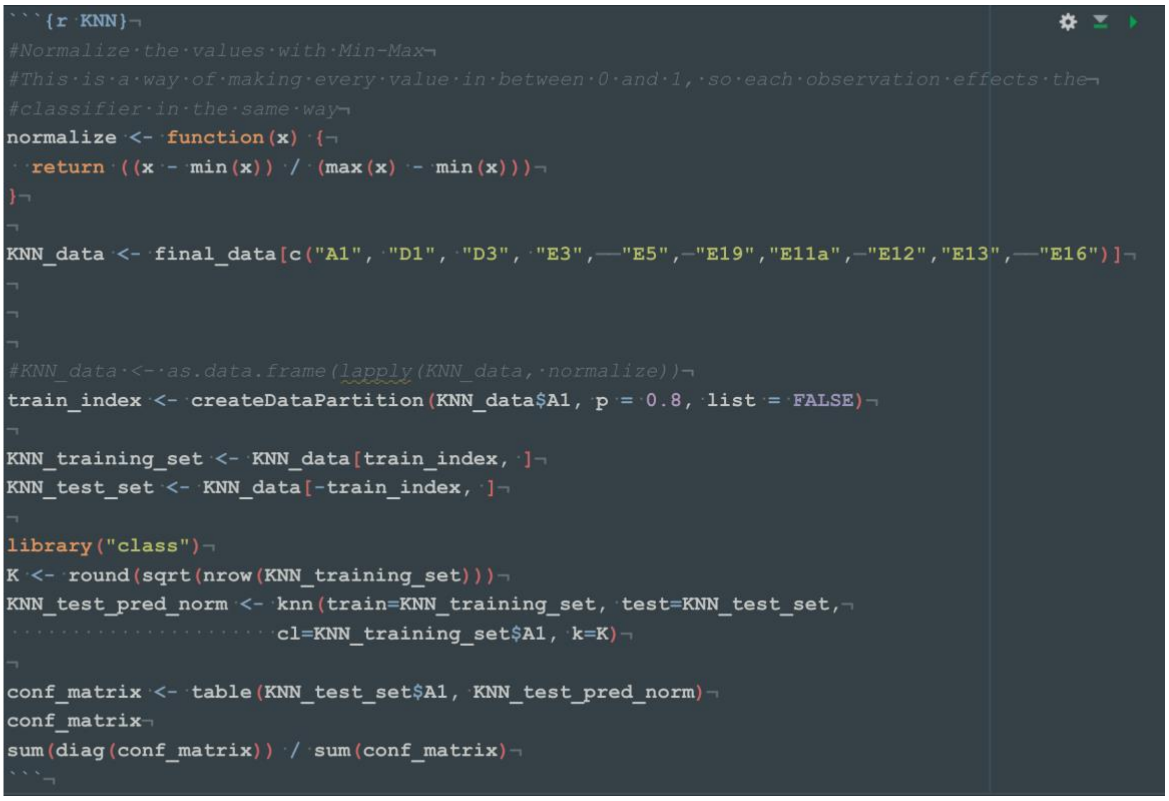
~~~
```{r desicion tree predict}
satisfaction_pred <- predict(satisfaction_tree, 
                       newdata = DT_test_set, 
                       type = "class") 

confusionMatrix(factor(satisfaction_pred) , factor(DT_test_set$A1))

#mean(satisfaction_pred==DT_test_set[,1])

```
~~~

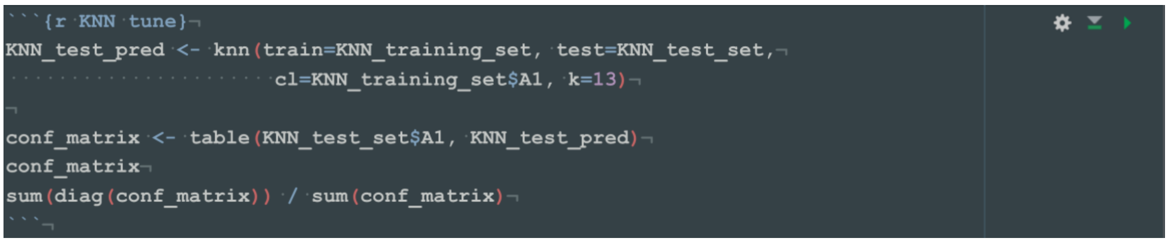


~~~
```{r random forest}
library(randomForest)
library(tidyverse)
library(stringr)

xdf <- DT_training_set %>% select(-A1)
ydf <- DT_training_set %>% select(A1)

BestMtry <-
  tuneRF(
    xdf,
    ydf$A1,
    stepFactor = 1.5,
    improve = 1e-6,
    ntree = 500, 
    plot = F
  )

BestMtry

rf <- randomForest(A1 ~ D1+D3+E3+E5+E19+E11a+E12+E13+E16,
                   data=DT_training_set,
                   ntree=500,
                   mtry = BestMtry
)
rf
plot(rf)
importance(rf)
varImpPlot(rf, main = "Importance of Each Attribute")
```
~~~



~~~
```{r random forest prediction}
rf_pred <- predict(rf, newdata = DT_test_set)

head(table(round(rf_pred), DT_test_set$A1))
mean(round(rf_pred)==DT_test_set[,1])
```
~~~



~~~r
```{r KNN}
#Normalize the values with Min-Max
#This is a way of making every value in between 0 and 1, so each observation effects the
#classifier in the same way
normalize <- function(x) {
  return ((x - min(x)) / (max(x) - min(x)))
}

KNN_data <- final_data[c("A1", "D1", "D3", "E3",	"E5",	"E19","E11a",	"E12","E13",	"E16")]



#KNN_data <- as.data.frame(lapply(KNN_data, normalize))
train_index <- createDataPartition(KNN_data$A1, p = 0.8, list = FALSE)

KNN_training_set <- KNN_data[train_index, ]
KNN_test_set <- KNN_data[-train_index, ]

library("class")
K <- round(sqrt(nrow(KNN_training_set)))
KNN_test_pred_norm <- knn(train=KNN_training_set, test=KNN_test_set,
                      cl=KNN_training_set$A1, k=K)

conf_matrix <- table(KNN_test_set$A1, KNN_test_pred_norm)
conf_matrix
sum(diag(conf_matrix)) / sum(conf_matrix)
~~~
~~~



```r
```{r KNN tune}
KNN_test_pred <- knn(train=KNN_training_set, test=KNN_test_set,
                      cl=KNN_training_set$A1, k=13)

conf_matrix <- table(KNN_test_set$A1, KNN_test_pred)
conf_matrix
sum(diag(conf_matrix)) / sum(conf_matrix)
~~~
```
```