

# **WATER QUALITY PREDICTION USING K-MEANS CLUSTERING**

**PROFESSOR: Ruijian Zhang**

**STUDENT NAME: WAJAHAT WAHEED**

## **Data Preprocessing:**

**Following is the information of the water data without any preprocessing:**

Date	317
Site ID	317
Stream Name	317
Drainage Area	286
Event	307
Flow	264
Flow2	265
Temp	281
DO	281
% Sat	264
pH	281
Conductivity	251
Ammonia nitrogen	149
Nitrate	41
Nitrate + Nitrite	242
Orthophosphate	120
% Dissolved P	90
Total phosphorus	149
Total suspended solids	149
E. coli	253
Time	30
Unit Flag	38
Ammonia Load	104
Nitrate Load	255
Orthophosphorus load	127
Total phosphorus load	144
Total suspended solids load	155
Turbidity	95
mIBI	24
dtype: int64	

Since we can see that there are a lot of missing values in a lot of features, I chose only 5 of these features which have precise data points and less missing values namely:

1. PH
2. DO
3. CONDUCTIVITY
4. SATURATION (IN %)
5. TEMPERATURE

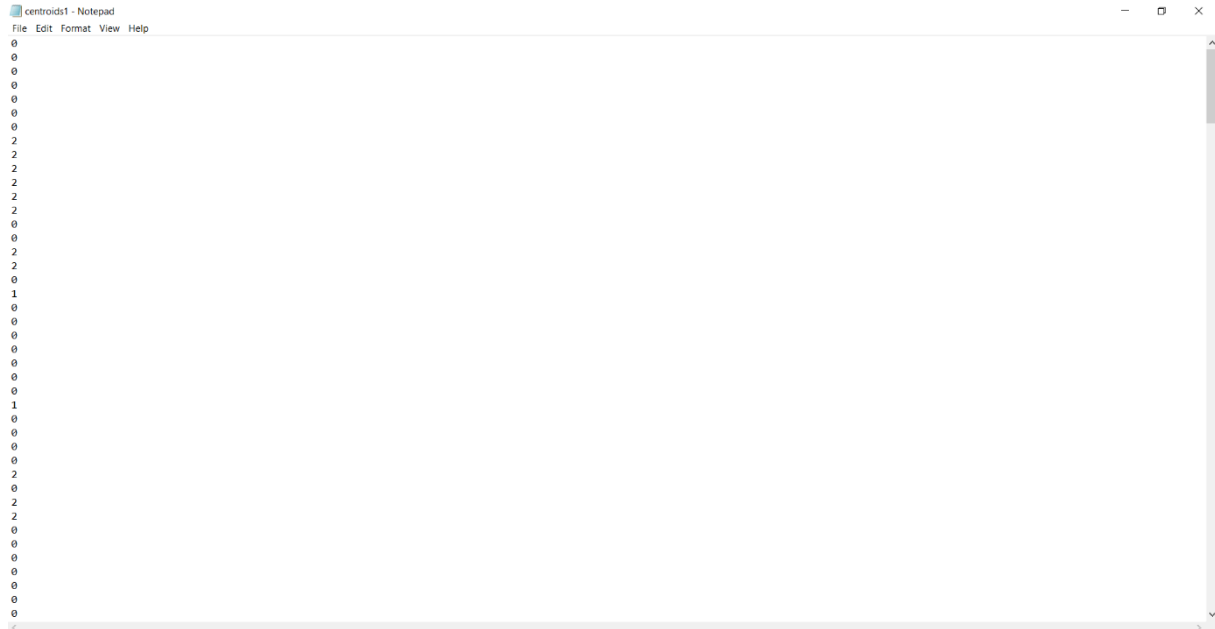
Code for the Data Processing written in Python is attached in the Project Folder. Following is the data summary and sample data that was prepared after Data Preprocessing:

	type	amount	null_values (%)	unique
<b>pH</b>	int32	0	0.0	80
<b>DO</b>	float64	0	0.0	220
<b>% Sat</b>	float64	0	0.0	198
<b>Temp</b>	float64	0	0.0	246
<b>Conductivity</b>	float64	0	0.0	188

	pH	DO	% Sat	Temp	Conductivity
0	7.800000	9.300000	77.350758	12.500000	843.000000
1	7.900000	9.500000	77.350758	12.200000	791.000000
2	8.100000	8.700000	77.350758	19.900000	1031.000000
3	8.100000	8.500000	77.350758	20.400000	980.000000
4	8.300000	7.300000	77.350758	24.900000	947.000000
5	8.300000	7.100000	77.350758	25.100000	872.000000
6	7.740000	7.350000	78.200000	18.120000	783.000000
7	7.700000	6.100000	77.200000	26.670000	597.000000
8	7.230000	2.510000	31.200000	27.550000	820.000000
9	7.180000	3.430000	37.700000	18.540000	804.000000
10	7.370000	5.080000	56.200000	22.050000	601.000000
11	7.630000	7.650000	75.100000	24.870000	621.000000
12	7.200000	6.300000	65.600000	16.560000	930.000000
13	8.050000	8.280000	90.700000	19.530000	774.000000
14	7.820000	6.750000	81.600000	24.910000	739.000000
15	7.650000	5.250000	60.300000	16.800000	642.000000
16	8.060000	5.200000	64.000000	20.200000	1378.000000
17	7.690000	7.320000	93.000000	25.930000	820.000000
18	7.880000	7.110000	81.900000	21.170000	831.000000

## K-MEANS CLUSTERING:

Once the data was preprocessed, I wrote a code for k-means clustering in C language which gives me a centroids.out file containing the classification of each row into either Good, Medium or Bad water quality. Following is the how the centroid.out looks like:

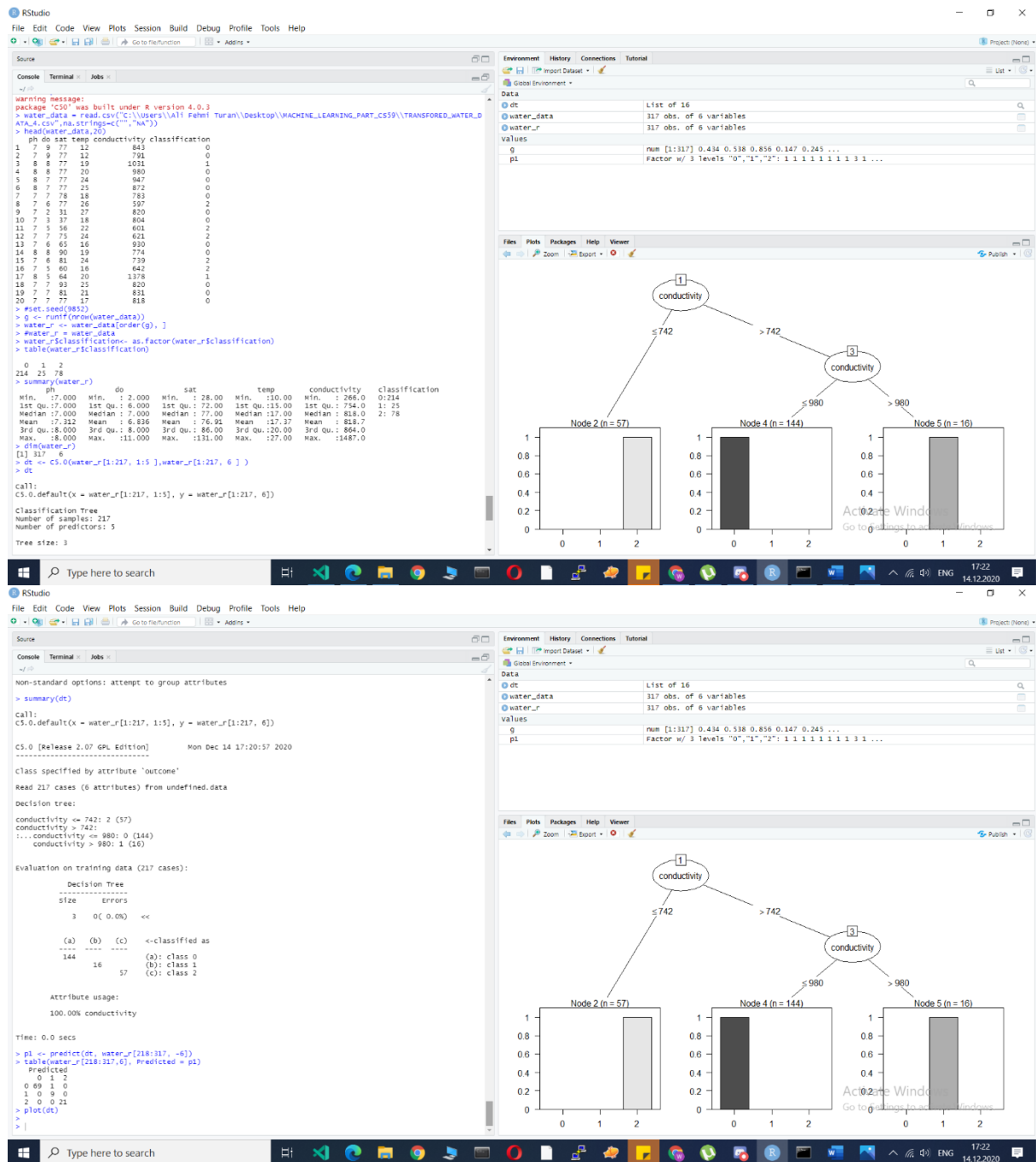


```
0
0
0
0
0
0
0
2
2
2
2
2
0
0
2
2
0
1
0
0
0
0
0
0
1
0
0
0
0
2
0
2
2
0
0
0
0
0
0
0
```

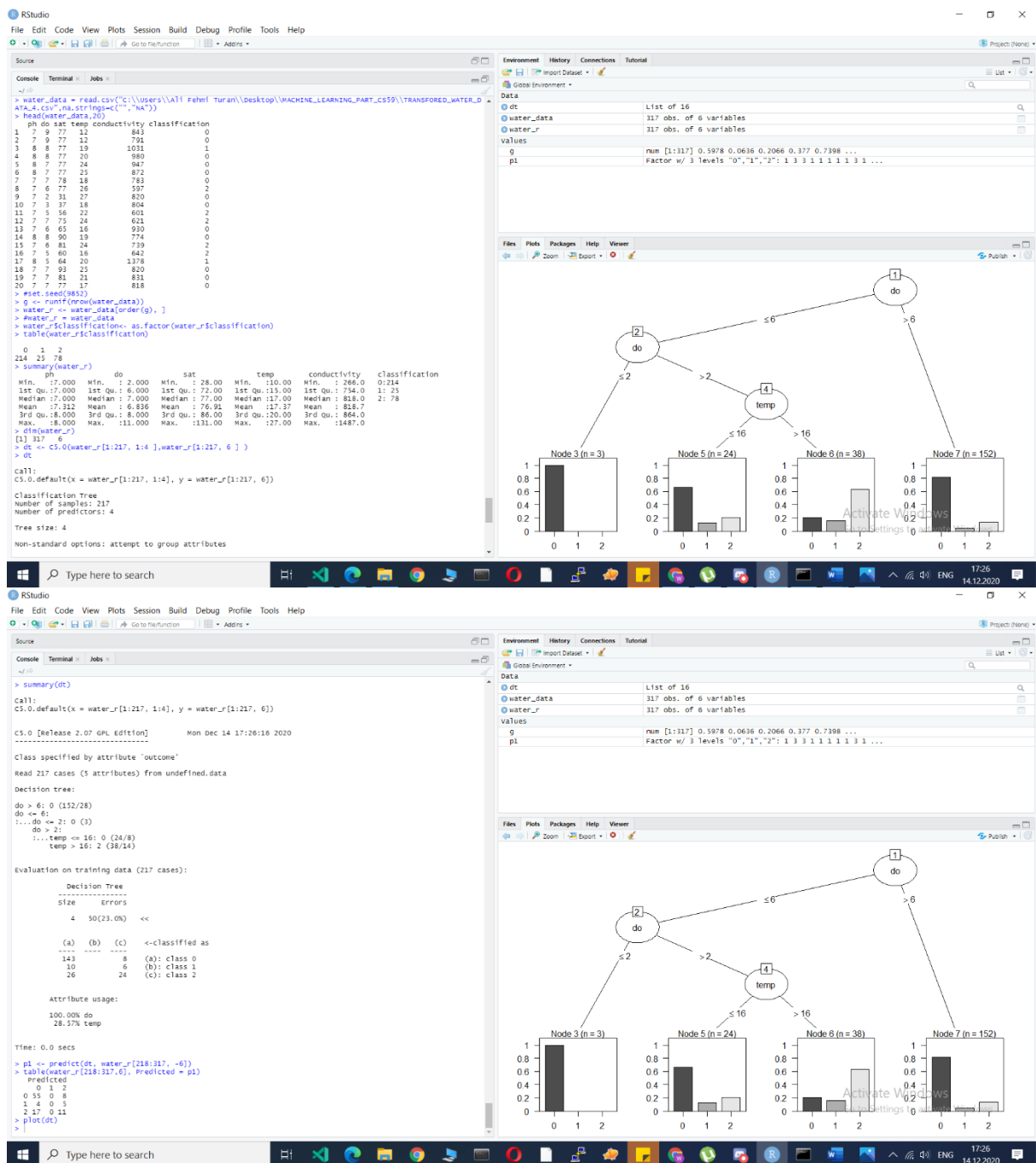
K-means code along with example centroids1.txt file is available in the folder. ReadMe file contains the instructions needed to run the C code with GCC compiler using Ubuntu.

## RESULTS AND ANALYSIS:

Once I got the classification data, I combined it with the original data that was preprocessed. For assessing how good the k-means code classified the data, I used R and the results are as follows:

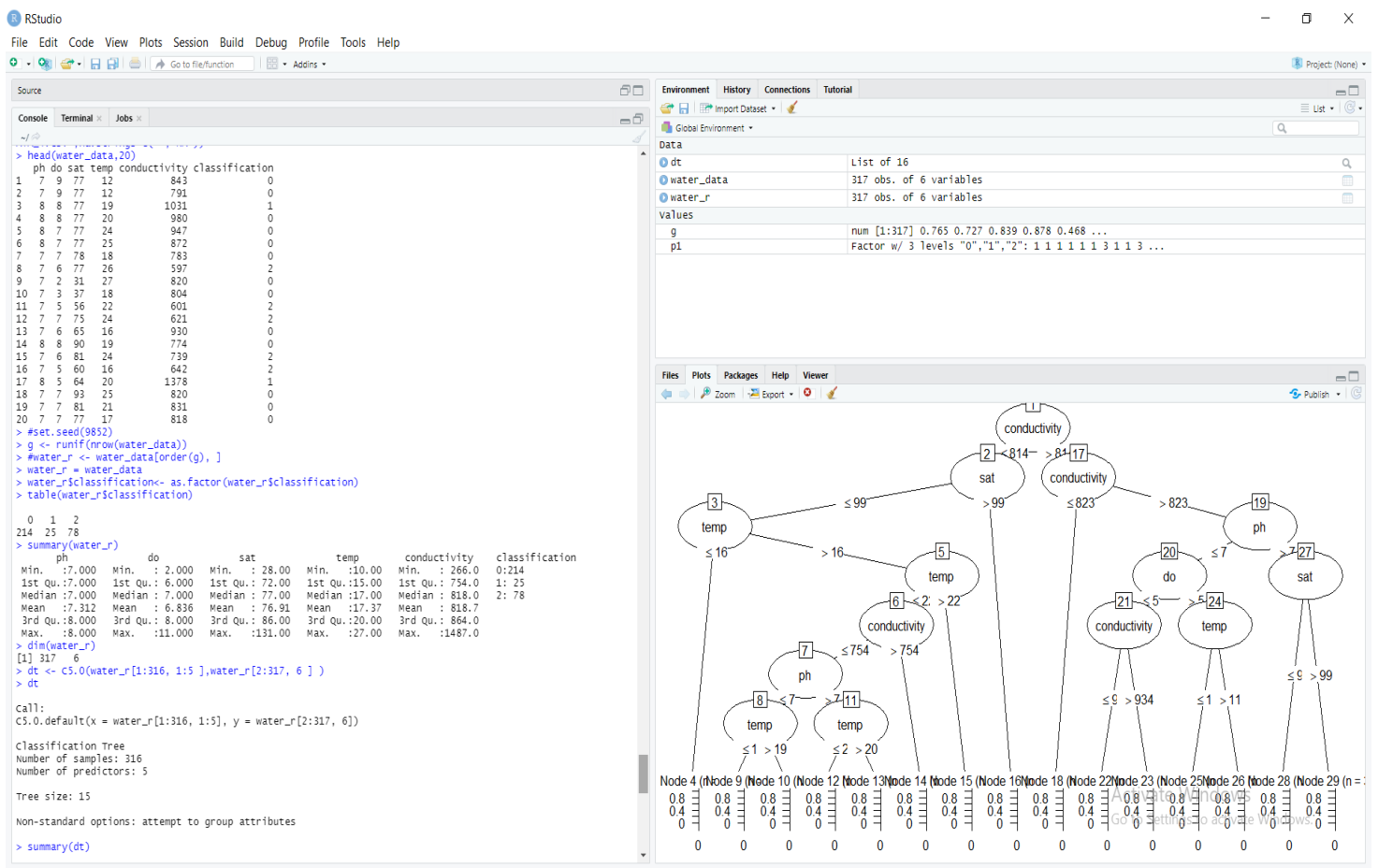


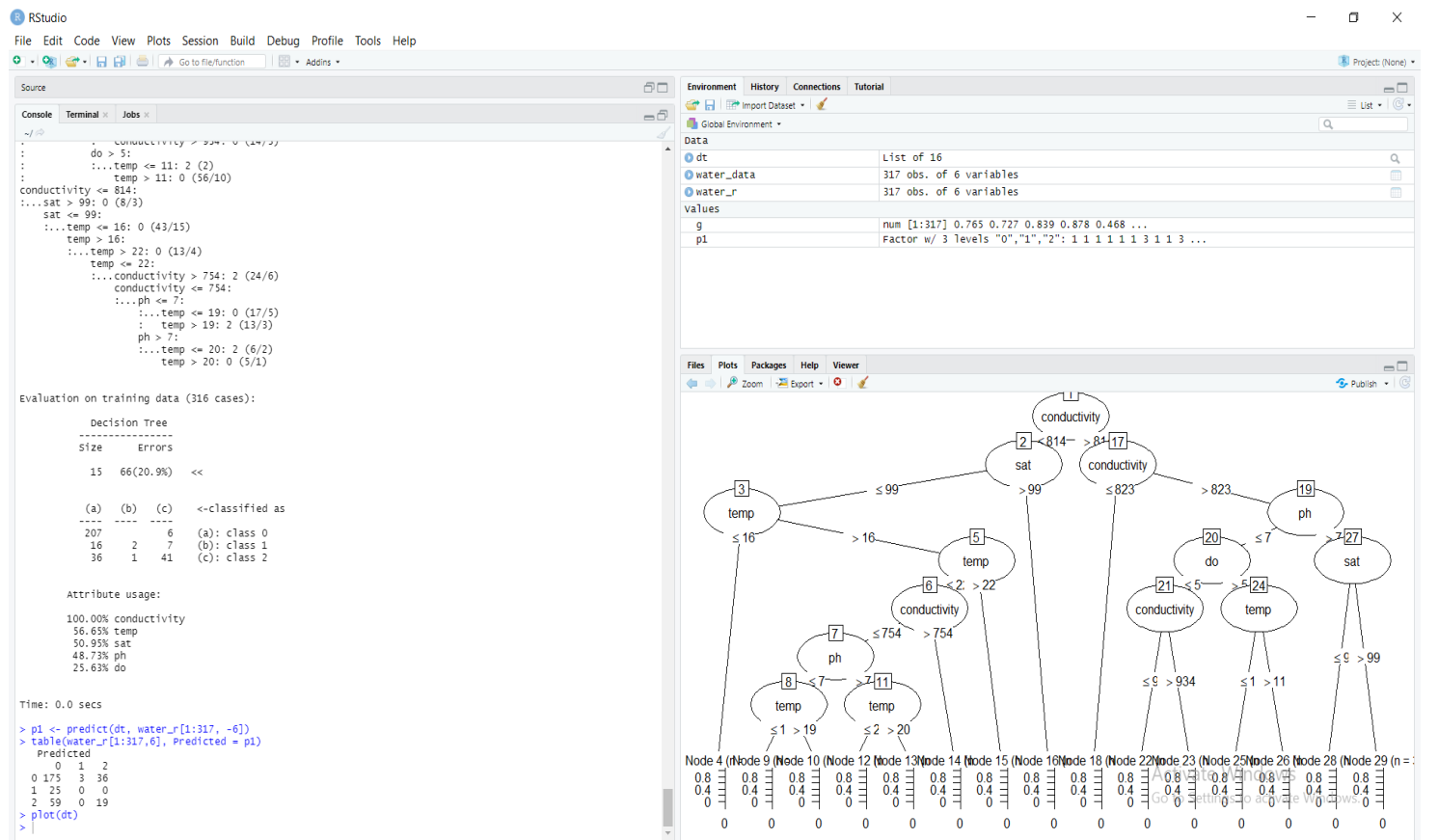
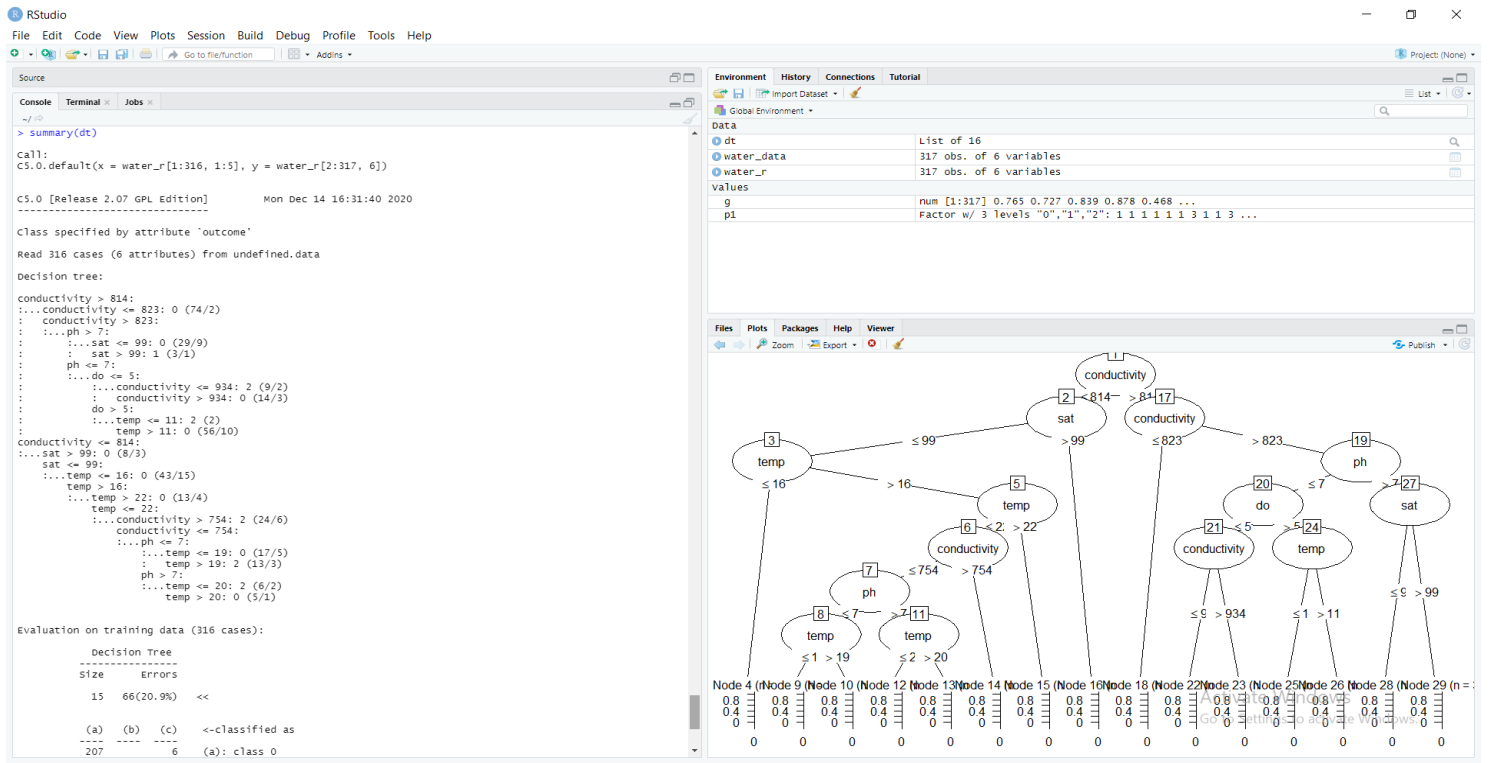
Please note from above that the classification results have a 100% classification accuracy since in the decision tree only of the predictor i.e. Conductivity is utilized for the C5 model. I split the data for training and prediction in this to keep training data different from test data for prediction. 0 represent Bad, 1 represent Medium and 2 represents Good water quality. Since the results felt too good to be true, I then removed Conductivity from the C5 Model and then did the whole process again. Following are the results without Conductivity as a predictor:



As you can see from the above screenshots, that this time the prediction accuracy for the test data decreased from 100% to 66 % and the C5 model uses temperature (28.57%) and DO(100%) as a predictor. In conclusion, Conductivity is a very strong predictor for the classification of water quality. However, when conductivity data is not present other predictors can also be used for classifying the water quality.

Relating to the discussion in class about prediction and classification as separate entities, I then used the “SHIFTED DATA” technique taught by the Professor himself in lecture. The whole idea is to use different data for classification and prediction. I used Day 1’s attributes for Day 2’s classification and in this way, we are predicting the next day’s water quality using today’s attributes. Following are the screenshots showing the results:





The results above show a 62% prediction accuracy for the water quality classification. The results do not seem very good. For this reason, I planned to add 2 more features and instead of using raw values for each feature, I used min-max normalization for all the features:

Following is the sample data after Data processing with each feature's summary:

Out[2]:

	type	amount	null_values (%)	unique
<b>pH</b>	int32	0	0.0	80
<b>DO</b>	int32	0	0.0	72
<b>Saturation</b>	int32	0	0.0	69
<b>Temp</b>	int32	0	0.0	79
<b>Conductivity</b>	int32	0	0.0	57
<b>Drainage_Area</b>	int32	0	0.0	12
<b>Nitrate_Plus_Nitrite</b>	int32	0	0.0	26

In [3]:

```
1 df_for_test
2 #df_for_test.to_csv("TRANSFORED_WATER_DATA_7_NORMALIZED.csv")
```

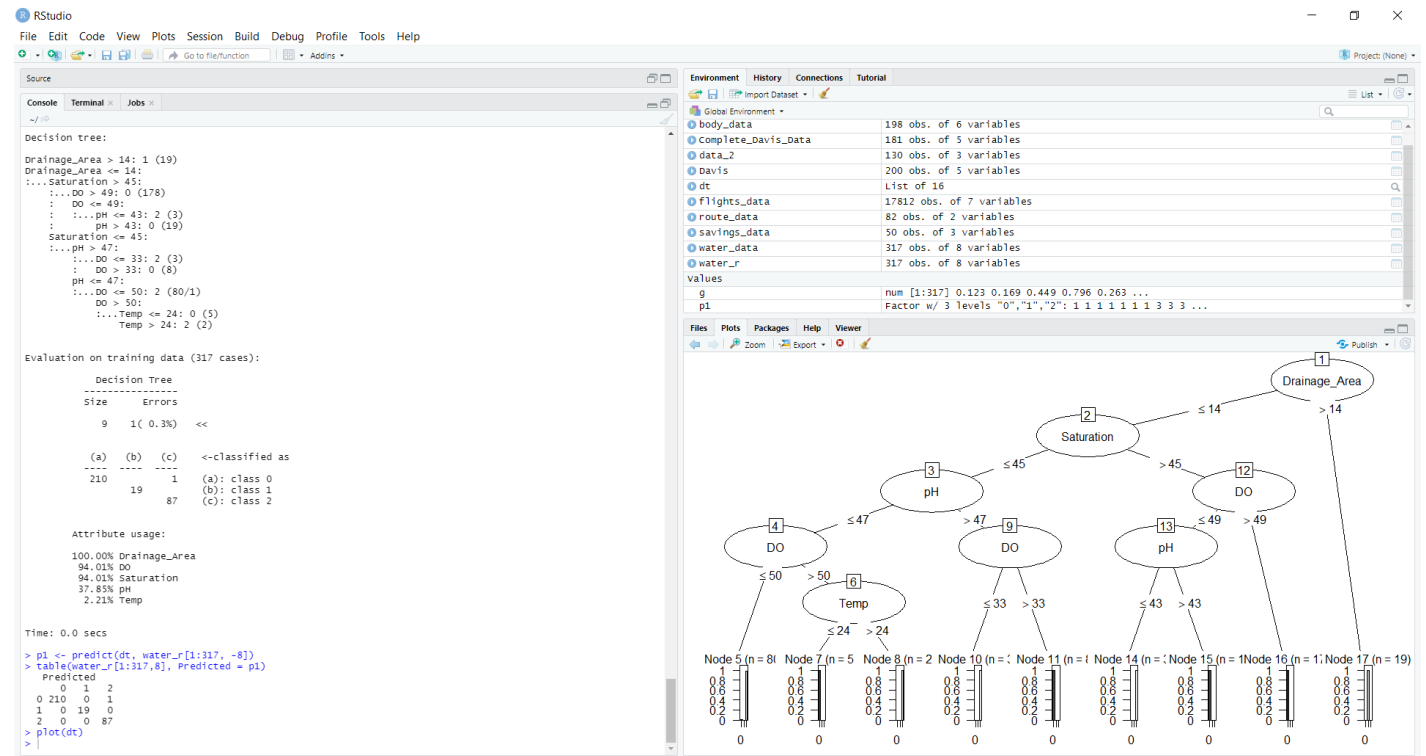
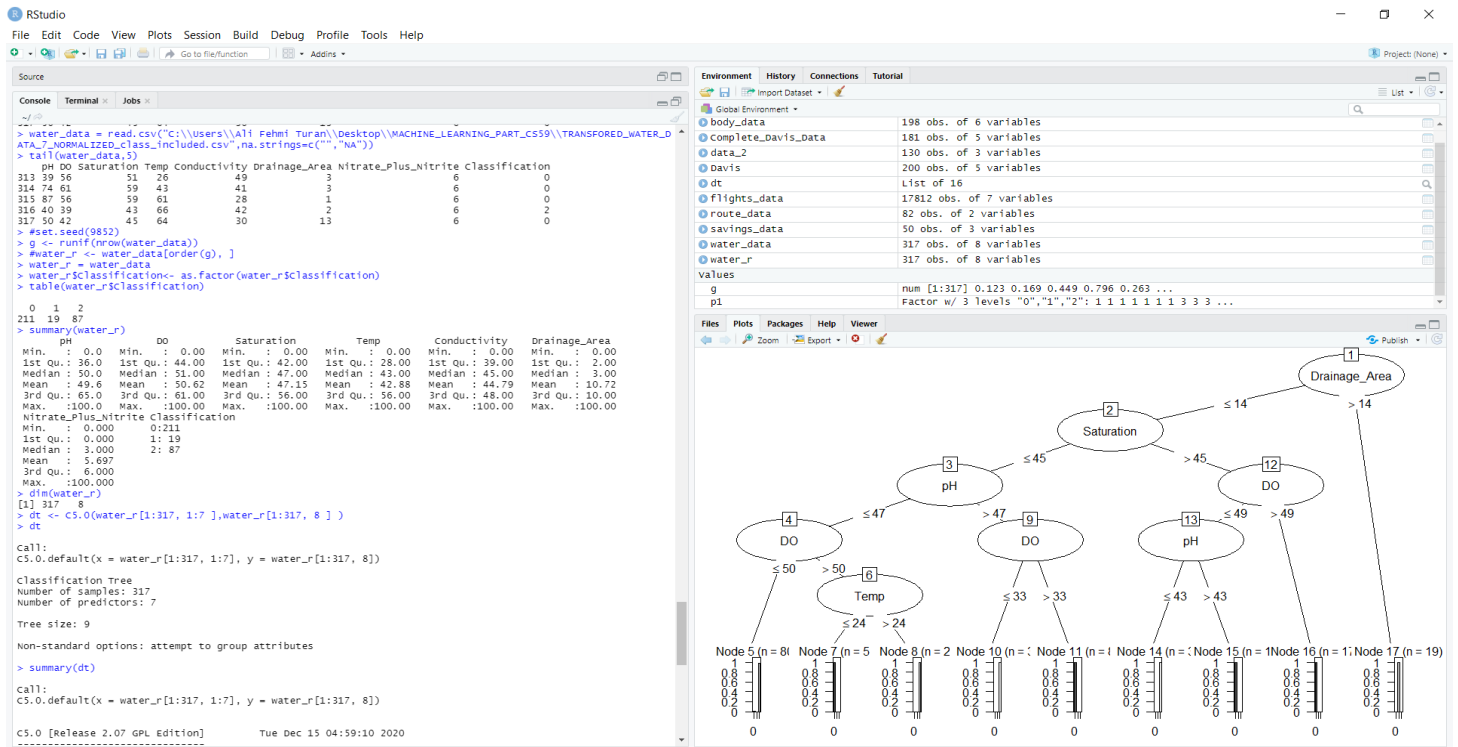
Out[3]:

	pH	DO	Saturation	Temp	Conductivity	Drainage_Area	Nitrate_Plus_Nitrite
<b>0</b>	46	72	47	11	47	11	8
<b>1</b>	54	74	47	9	42	11	5
<b>2</b>	69	66	47	54	62	11	14
<b>3</b>	69	63	47	57	58	11	11
<b>4</b>	84	51	47	84	55	11	10
<b>5</b>	84	49	47	85	49	11	7
<b>6</b>	42	51	48	44	42	3	16
<b>7</b>	39	38	47	94	27	5	0
<b>8</b>	3	0	3	100	45	1	0
<b>9</b>	0	10	9	47	44	0	0
<b>10</b>	14	27	27	67	27	10	1

After performing k-means clustering on this data using kmeans.c file (attached), I obtained TRANSFORED\_WATER\_DATA\_7\_NORMALIZED\_class\_include d.csv and then used C5 using R language to perform classification and prediction.

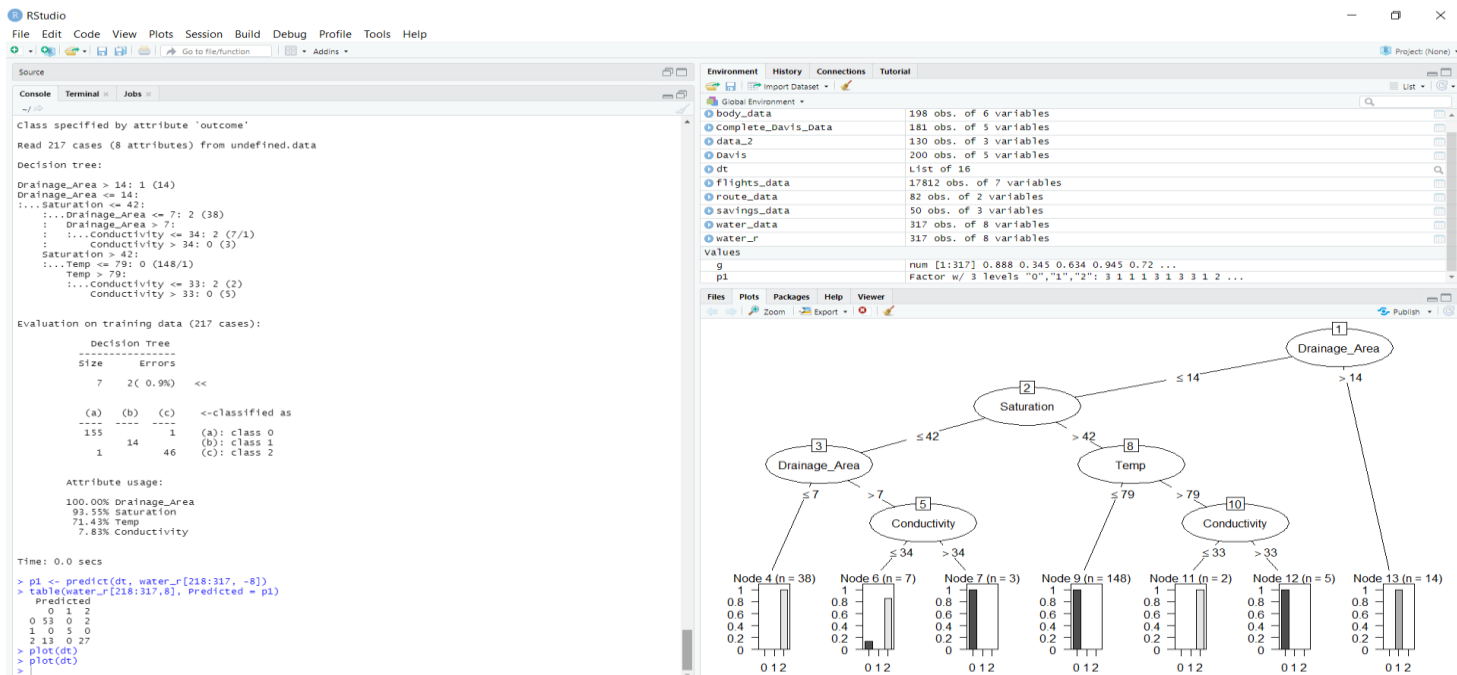
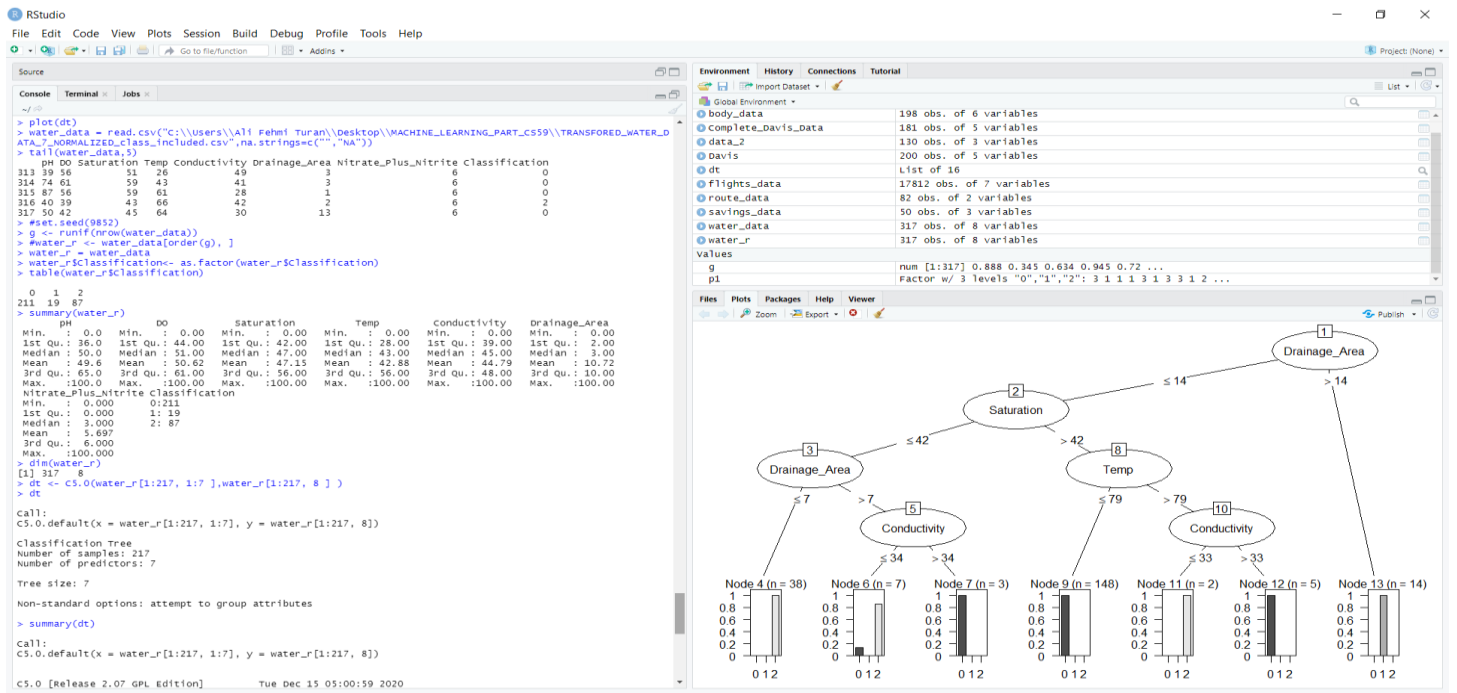


Following are the results obtained using C5  
(C5\_AND\_PREDICTIONS.R file attached) for same training and test data with no shifting of data:



This shows a prediction accuracy of 99.7%. Quite remarkable but it is basically equivalent to knowing the results beforehand.

Following are the results for training(1:217) and test data(218:317):



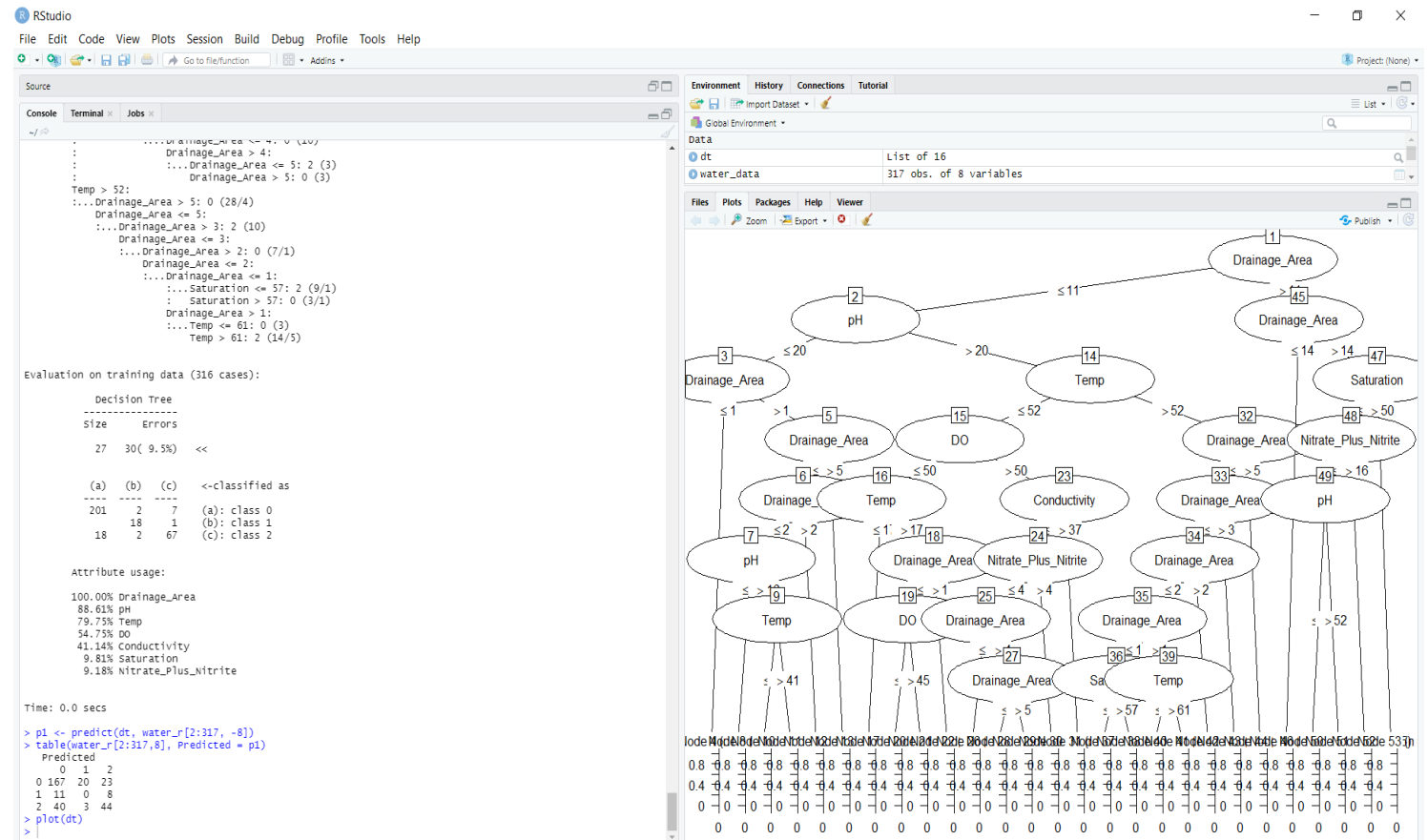
This shows a prediction accuracy of 85.86 % which is a good number.

The screenshot shows the RStudio interface with the following components:

- Console:** Contains R code for loading data, summarizing it, and training a decision tree. The code includes:
 

```

      ATA_7_NORMALIZED_class_included.csv", na_strings=c("", "NA"))
      > water_data = read.csv("C:\\Users\\A11 Fehmi Turan\\Desktop\\\\MACHINE_LEARNING_PART_C559\\\\TRANSFORMED_WATER_D
      ATA_7_NORMALIZED_class_included.csv", na_strings=c("", "NA"))
      > tail(water_data, 5)
      pH DO Saturation Temp Conductivity Drainage_Area Nitrate_Plus_Nitrite Classification
      313 39 56 51 26 49 3 6 0
      314 74 61 59 43 41 3 6 0
      315 87 56 59 61 28 1 6 0
      316 40 39 43 66 42 2 6 2
      317 50 42 45 64 30 13 6 0
      > #set.seed(9852)
      > g <- runif(nrow(water_data))
      > water_r <- water_data[order(g, )]
      > water_r = water_data
      > water_r$Classification<- as.factor(water_r$Classification)
      > table(water_r$Classification)
      0 1 2
      211 19 87
      > summary(water_r)
      pH Min.: 0.0 Min.: 0.00 Saturation Min.: 0.00 Temp Min.: 0.00 Conductivity Min.: 0.00 Drainage_Area Min.: 0.00
      1st Qu.: 36.0 1st Qu.: 44.0 1st Qu.: 42.0 1st Qu.: 28.0 1st Qu.: 39.0 1st Qu.: 2.00
      Median: 50.0 Median: 51.0 Median: 47.0 Median: 45.0 Median: 45.0 Median: 3.00
      Mean: 49.6 Mean: 50.62 Mean: 47.15 Mean: 42.88 Mean: 44.79 Mean: 10.72
      3rd Qu.: 65.0 3rd Qu.: 61.0 3rd Qu.: 56.0 3rd Qu.: 56.0 3rd Qu.: 48.0 3rd Qu.: 10.00
      Max.: 100.0 Max.: 100.00 Max.: 100.00 Max.: 100.00 Max.: 100.00 Max.: 10.00
      Nitrate_Plus_Nitrite Classification
      Min.: 0.000 0.211
      1st Qu.: 0.000 1: 19
      Median: 3.000 2: 87
      Mean: 5.697
      3rd Qu.: 6.000
      Max.: 100.000
      > dt(water_r)
      [1] 317 8
      > dt <- c5.0(water_r[1:316, 1:7 ], water_r[2:317, 8 ])
      > dt
      Call:
      c5.0.default(x = water_r[1:316, 1:7 ], y = water_r[2:317, 8 ])
      Classification Tree
      Number of samples: 316
      Number of predictors: 7
      Tree size: 27
      Non-standard options: attempt to group attributes
      > summary(dt)
      Call:
      c5.0.default(x = water_r[1:316, 1:7 ], y = water_r[2:317, 8 ])
      C5.0 [Release 2.07 GPL Edition] Tue Dec 15 05:04:19 2020
      
```
- Environment:** Shows the loaded data objects: 'dt' (List of 16) and 'water\_data' (317 obs. of 8 variables).
- Plots:** Displays a complex decision tree structure with nodes and splits. The tree starts with a root node (1) splitting on 'Drainage\_Area'. Subsequent nodes (2, 3, 14, 15, 16, 23, 24, 25, 27, 32, 33, 34, 35, 36, 39, 41, 45, 47, 48, 49) split on various predictors like pH, Temp, DO, Conductivity, Nitrate\_Plus\_Nitrite, and Saturation. The final nodes (53) show the predicted class probabilities for each leaf.



The shifted data which predicts the next day's water quality using today's attributes has a prediction accuracy of 66.77 %. Although we added two more features and normalized the data, our prediction accuracy still did not improve much. However, this time the reliance on just perimeter from the C 5.0 model was removed and instead almost all the features are just in building the decision tree. From the decision trees, the feature at the root node is the most important one since it is the one with Largest Information Gain. From the results, I can conclude that Drainage\_Area along with Conductivity are the two most important features.

Possible changes which can improve the model performance:

1. Use a larger dataset
2. Use more features
3. Try normalization for each feature one by one and see the affect it has on prediction accuracies
4. Use a dataset with better quality data (fewer missing values, more precise data points)