**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**[Classification of Key Risk Factors for Alzheimer's From Questionnaire Data]**

**Team Members:**

**Leah Zhang and Victoria Zhang**

Period 4

**9/17/2024**

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Table of Contents**

[**Part 1 – Project Statement 3**](#_heading=h.30j0zll)

[**Part 2 – Description of Dataset 3**](#_heading=h.e04jbfmyj0pi)

[**Part 3 – Preprocessing 6**](#_heading=h.vl0agaql18qs)

[3.1 Cleaning Up Redundant Columns 6](#_heading=h.5u1rauj9r5kc)

[3.2 Processing Strings; Fixing Specific Values 6](#_heading=h.1icgkj2yiu4v)

[3.3 Replacing Nans/Null values 7](#_heading=h.b6pjjkg32vs)

[3.4 Normalizing Features 7](#_heading=h.7gpv925fqpul)

[3.5 Transform Non-Numerical data into Numerical Data 7](#_heading=h.krtattlti7v3)

[3.6 Splitting the Dataset to Train/Validation/Test 7](#_heading=h.lhdnbu52txz9)

[**Part 4 – Attribute Selection 7**](#_heading=h.3qtewplonyqo)

[Following preprocessing, we conducted attribute selection to filter out attributes that do not contribute significantly to the class, as to reduce dimensionality. All files post-attribute selection can be found in a folder labeled accordingly in the drive. 7](#_heading=h.fwahna40wor4)

[4.1 Kendall Rank Correlation Coefficient Feature Selection (Non-Weka) 7](#_heading=h.7pllxulyiyd0)

[4.2 Selection by Pearson’s Correlation 8](#_heading=h.12cbeigxzwli)

[4.3 Selection by Subset Analysis 10](#_heading=h.hfvw10y9o9iu)

[4.4 Selection by the OneR Algorithm 11](#_heading=h.uisgjb8i6tnp)

[4.5 Selection by Information Gain 13](#_heading=h.27uymm2oexha)

[**Part 4 - Classifier Models 14**](#_heading=h.uonotc9johr)

[**Part 5 – Results and Analysis 14**](#_heading=h.tyjcwt)

[5.1 - Results 14](#_heading=h.a85d1hcxi1hb)

[CLASSIFICATION WITH KENDALL’S CORRELATION-SELECTED ATTRIBUTES: 14](#_heading=h.nq8j4q7mffyt)

[CLASSIFICATION WITH PEARSON’S CORRELATION-SELECTED ATTRIBUTES: 16](#_heading=h.abspc8t3pqdg)

[CLASSIFICATION WITH SUBSET ANALYSIS-SELECTED ATTRIBUTES: 18](#_heading=h.fbcd0k3cfraa)

[CLASSIFICATION WITH ONER-SELECTED ATTRIBUTES: 21](#_heading=h.ac68in2ej3mo)

[CLASSIFICATION WITH ENTROPY GAIN-SELECTED ATTRIBUTES: 23](#_heading=h.ttysp7hzr8rt)

[5.2 - Analysis 26](#_heading=h.7wd8fj5ll7yc)

[5.3 - Results Without Topic & Question 31](#_heading=h.ekt97y8s03sb)

[**Part 6 – Conclusion 33**](#_heading=h.3dy6vkm)

[**Part 7 - Team Members and Tasks Performed 34**](#_heading=h.1t3h5sf)

[**Part 8 – Appendix and Sources: 35**](#_heading=h.4d34og8)

[**References 35**](#_heading=h.tghlrfrexpml)

# **Part 1 – Project Statement**

The Behavioral Risk Factor Surveillance System (BRFSS) compiled a list for risk factors regarding Alzheimer’s disease and Healthy Aging Data. Alzheimer’s disease is a common brain disorder that slowly destroys memory capabilities and cognitive processes. The six conditions we are attributing to a patient's health as indicators are:

* Screenings and Vaccines
* Nutrition/Physical Activity/Obesity
* Caregiving
* Mental Health
* Smoking and Alcohol Use
* Cognitive Decline

From the questionnaire, we use this project to identify and classify the six categories used to differentiate risk factors for Alzheimer’s based on specific questions asked. These classification models can allow for improved annotation in medical notes in future surveys.

# **Part 2 – Description of Dataset**

The link to our data can be found here: [Alzheimer's Disease and Healthy Aging Data | Data | Centers for Disease Control and Prevention (cdc.gov)](https://chronicdata.cdc.gov/Healthy-Aging/Alzheimer-s-Disease-and-Healthy-Aging-Data/hfr9-rurv/about_data)**.** The data comes from a questionnaire, conducted via telephone surveys by the BRFSS to collect state data about U.S. residents regarding health. In this particular dataset, such questions, identified under the ‘Question’ attribute, asked about health-related risk behaviors, chronic health conditions, health-care access, and use of preventive services.

We are classifying these individuals using attributes including age, location, and datasource. In total, there are thirty attributes and 284,142 instances. Each instance represents an individual respondent. The dimension of the data is 284,142. The thirty attributes, as well as their descriptions are:

| RowId | Dataset row identifier |
| --- | --- |
| YearStart | Year Start |
| YearEnd | Year End |
| LocationAbbr | Location Abbreviation |
| LocationDesc | Location Description |
| Datasource | Data Source |
| Class | Class description |
| Topic | Topic description |
| Question | Question |
| Data\_Value\_Unit | The unit, such as "%" for percentage |
| DataValueTypeID | Identifier for the Data Value Type |
| Data\_Value\_Type | The data value type, such as age-adjusted prevalence or crude prevalence |
| Data\_Value | Data Value, such as 14.7 |
| Data\_Value\_Alt | Equal to data value, but format is numeric |
| Data\_Value\_Footnote\_Symbol | Footnote Symbol |

| Data\_Value\_Footnote | Footnote Text |
| --- | --- |
| Low\_Confidence\_Limit | Low Confidence Limit |
| High\_Confidence\_Limit | High Confidence Limit |
| StratificationCategory1 | Stratification grouping e.g. Age group, Race/ethnicity group |
| Stratification1 | Stratification value e.g. 18-24yrs |
| StratificationCategory2 | Stratification grouping e.g. Age group, Race/ethnicity group |
| Stratification2 | Stratification value e.g. 18-24yrs |
| Geolocation | The exact coordinates for the location |
| ClassID | Identifier for Class |
| TopicID | Topic Identifier |
| QuestionID | Question or Indicator Identifier |
| LocationID | Location number value corresponding to geographic location like state |
| StratificationCategoryID1 | Identifier for the first category stratification |
| StratificationID1 | Identifier for the first stratification |
| StratificationCategoryID2 | Identifier for the second category stratification |

There are six possible values for the class, each of which represent a heath indicator category the questionnaire is exploring. The distribution of the instances among these values are as follows:

* Screenings and Vaccines 62153
* Nutrition/Physical Activity/Obesity 33194
* Caregiving 25493
* Mental Health 22184
* Smoking and Alcohol Use 22183
* Cognitive Decline 22182

From the above distribution, we note that the data is skewed towards one category, ‘Screenings and Vaccines’.

There are 818,325 values that are missing from the data source in total there are 8,434,260 values. We can easily remove this using a filtering method. This is mainly in the Data\_Value\_Footnote and Data\_Value\_Footnote\_Symbol attributes and we can easily ignore them.

# Part 3 – Preprocessing

## 3.1 Cleaning Up Redundant Columns

In order to reduce redundant data, fourteen columns were removed, ['Data\_Value\_Footnote\_Symbol', 'Data\_Value\_Footnote', 'ClassID', 'QuestionID', 'TopicID', 'RowId', 'Data\_Value\_Alt', 'StratificationCategoryID1', 'StratificationCategoryID2', 'Data\_Value\_Alt', 'LocationAbbr', 'LocationID', 'StratificationID1', 'StratificationID2']. They were irrelevant because they were annotations used to describe other columns in the dataset. Most of the removed columns were ID’s that describe other columns.

## 3.2 Processing Strings; Fixing Specific Values

There were several issues with processing the strings.

**Problem 1: Commas**

Weka uses commas as indicators of a separate item in a row. However, some of the values in the csv were text responses and thus contained commas. "White, non-Hispanic" is an example. To fix this we read in the columns containing these text values and replaced the commas with spaces.

df['Stratification2'] = df['Stratification2'].str.replace(',', '', regex=False)

df['DataValueTypeID'] = df['DataValueTypeID'].str.replace(',', '', regex=False)

df['Question'] = df['Question'].str.replace(',', '', regex=False)

Figure 1.2.1

**Problem 2: Quotation Marks**

Similar to the previous issue, Weka has trouble with quotation marks. This is solved by replacing the ‘ “ ‘ with a space.

**Problem 3: Blank Attribute Name**

When you process with Pandas it adds an extra column for the row number. However this column has no name and interferes with Weka’s file reading. This is solved by manually putting a column name or just by removing the column.

## 3.3 Replacing Nans/Null values

To have a complete final data, for every missing value we fill it in. If the value has a quantitative datatype, the value is filled by the mean of its class. If the value is qualitative, the value is filled by the mode of the class. This is done by first checking if the column has null values and then for each value in the column we replace it with its respective statistic.

## 

## 3.4 Normalizing Features

Excluding the quantitative attributes that are categorical, e.g. YearStart, YearEnd, we normalize the attributes ['Data\_Value', 'Low\_Confidence\_Limit', 'High\_Confidence\_Limit']

.We use Z-score normalization using the zscore function from scipy.stats.

## 3.5 Transform Non-Numerical data into Numerical Data

When the data is not transformed into numerical values, the attribute selection is limited in Weka. Thus, we transform the data, sorting each class into a specific value using pandas factorize function.

## 3.6 Splitting the Dataset to Train/Validation/Test

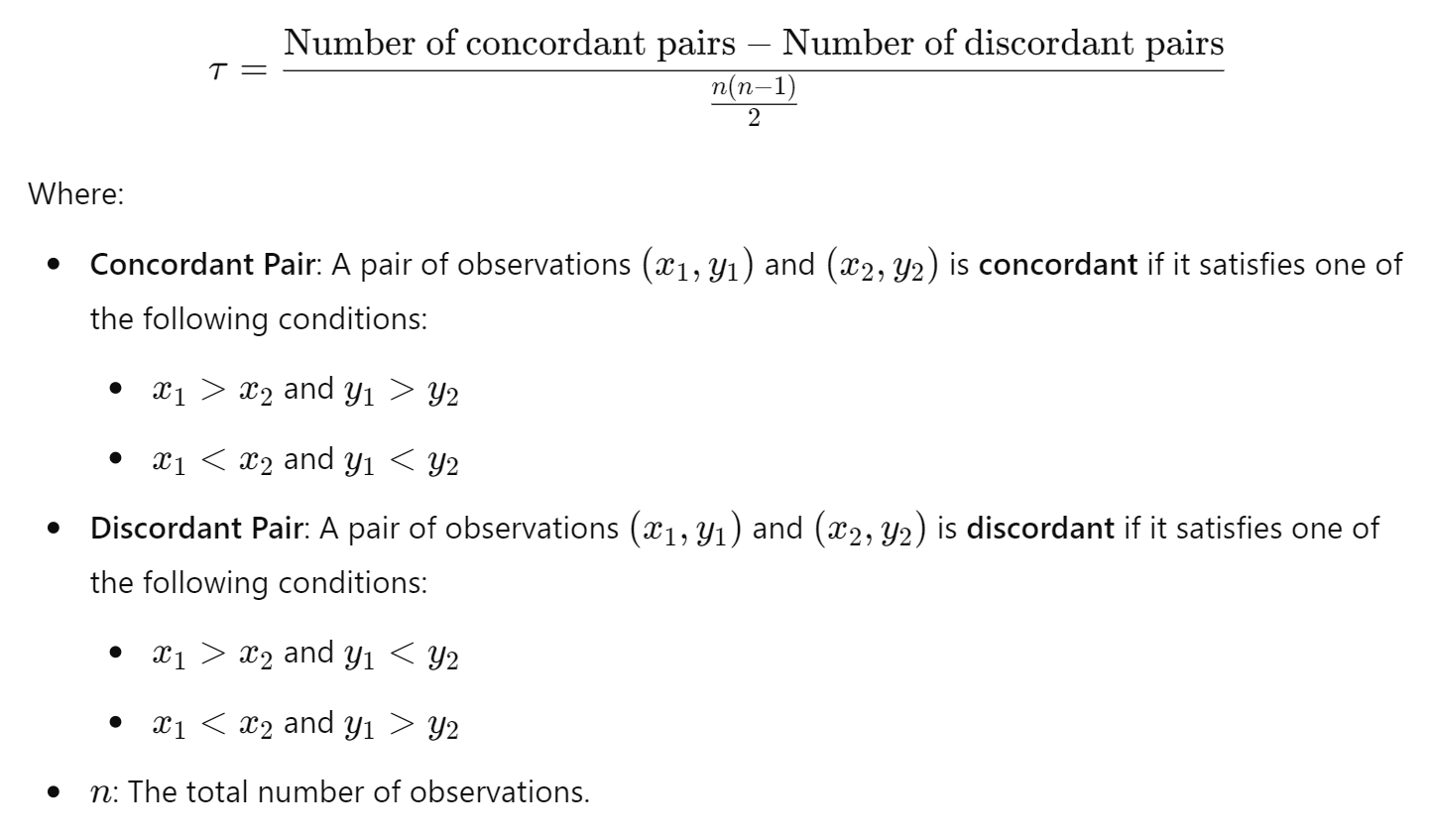
We will do a 70%/20%/10% split for the train and test datasets. We will use sklearn’s train\_test\_split to split the data with the stratify argument to ensure the ratios of the class are the same. The lengths are 198899, 57112, and 28131 respectively. 284,142

# **Part 4 – Attribute Selection**

## Following preprocessing, we conducted attribute selection to filter out attributes that do not contribute significantly to the class, as to reduce dimensionality. All files post-attribute selection can be found in a folder labeled accordingly in the drive.

## 4.1 Kendall Rank Correlation Coefficient Feature Selection (Non-Weka)

We will use the Kendall Rank Correlation Coefficient. Correlation is finding the strength of association between separate variables. The difference between the Kendall Rank’s correlation coefficient and Pearson’s is that Pearson relies on a linear relationship and Kendall is non parametric.



**Concordant Pair**: (x1,y1)(x\_1, y\_1)(x1​,y1​) and (x2,y2)(x\_2, y\_2)(x2​,y2​) is **concordant** if it satisfies one of the following conditions:

x1>x2x\_1 > x\_2x1​>x2​ and y1>y2y\_1 > y\_2y1​>y2​

x1<x2x\_1 < x\_2x1​<x2​ and y1<y2y\_1 < y\_2y1​<y2​

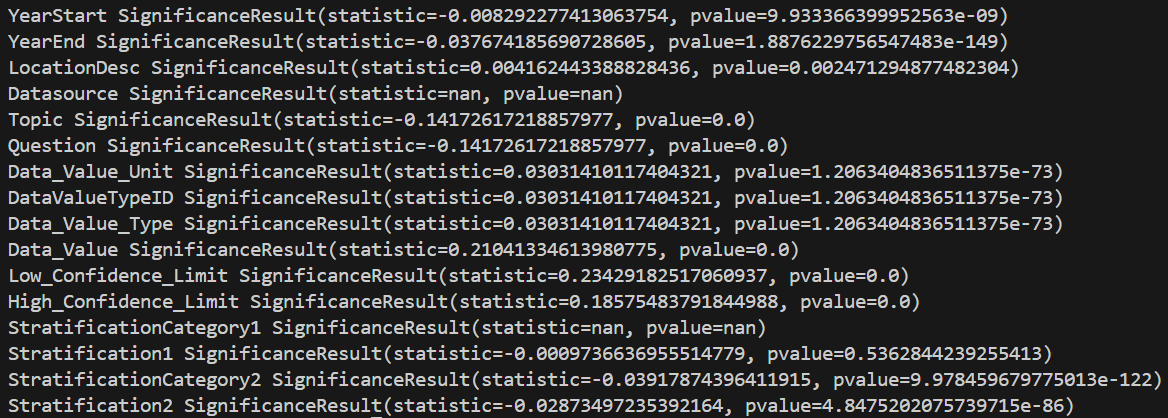
**Discordant Pair**: A pair of observations (x1,y1)(x\_1, y\_1)(x1​,y1​) and (x2,y2)(x\_2, y\_2)(x2​,y2​) is **discordant** if it satisfies one of the following conditions:

x1>x2x\_1 > x\_2x1​>x2​ and y1<y2y\_1 < y\_2y1​<y2​

1<x2x\_1 < x\_2x1​<x2​ and y1>y2y\_1 > y\_2y1​>y2​

n: The total number of observations.

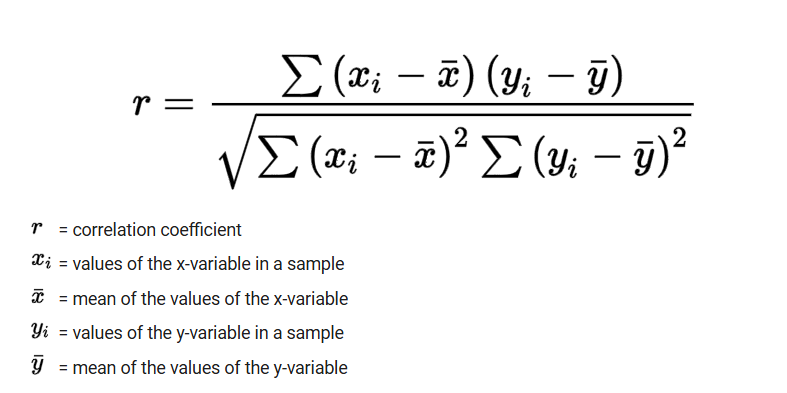
Running kendalltau from scipy, we get the following results.



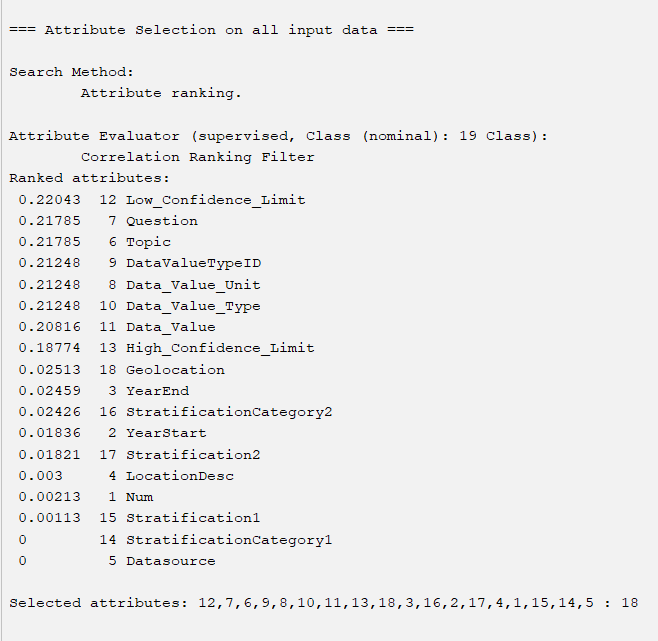
Values closer to 1 and -1 are kept. Thus we make 0.10 and -0.10 as our bounds. This keeps Topic, Question, Data\_Value, Low\_Confidence\_Limit, and High Confidence\_Limit.

## 4.2 Selection by Pearson’s Correlation

CorrelationAttributeEval in WEKA selects attributes based on the Pearson’s correlation calculated between an attribute and the class. The formula for Pearson’s correlation is as follows:



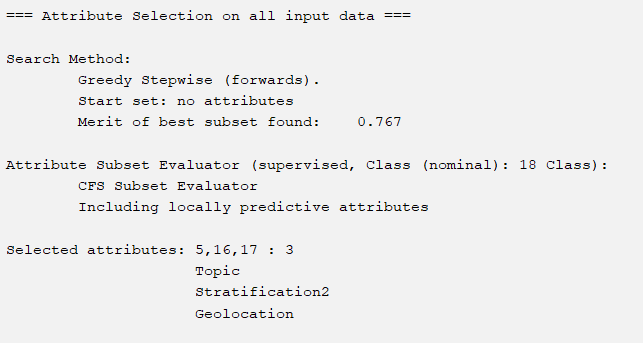
Running this correlation selection yielded the following results:



With a cutoff value of 0.20, all the values following 10 (Data\_Value) would be filtered out.

## 4.3 Selection by Subset Analysis

CfsSubsetEval in WEKA selects attributes based on firstly grouping them as subsets, then considering the attributes individually by how well they can predict the class, as well as how redundant they may be with respect to other attributes in the subset. Running this with a greedy stepwise search that searches through the space of attribute subsets gives:

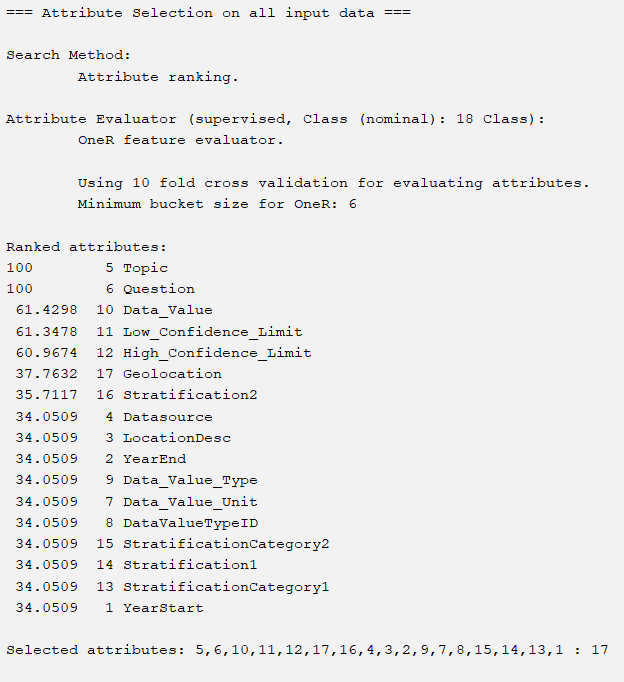


Based on these results, the new dataset from this selection algorithm would only include the attributes of Topic, Stratification2, Geolocation.

## 

## 4.4 Selection by the OneR Algorithm

OneRAttributeEval in WEKA uses the OneR classifier to evaluate the worth of an attribute, which can then be used to rank attributes with the Ranker search method. As discussed in class, the OneR classifier operates by selecting a set of “rules” for each unique value for each attribute, which are based on the least amount of classification error. Running this alongside the Ranker search ranks the attributes as such:

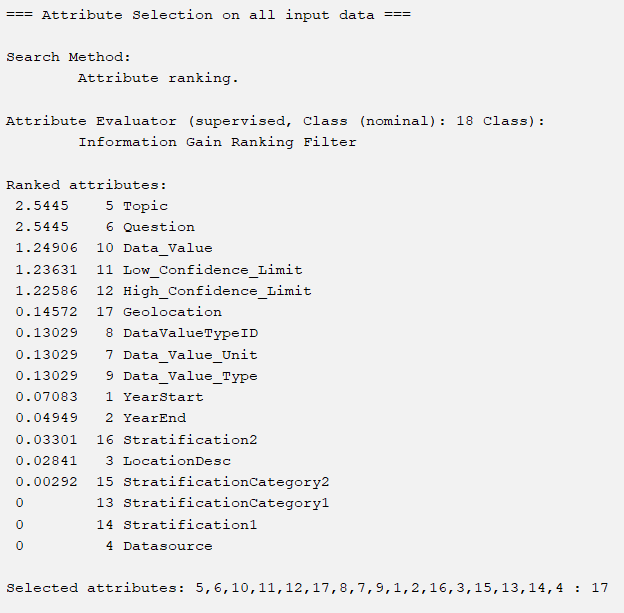


Originally, we considered filtering off all accuracies lower than 60, due to the great jump in ranking from 12 High\_Confidence\_Limit to 17 Geolocation. However, we noticed that using this filter would give exactly the same attributes selected by Pearson’s Correlation coefficient. Therefore, for the sake of producing a unique set of selected attributes, we filtered off accuracies less than 37, removing all attributes below 17 (Geolocation).

## 

## 4.5 Selection by Information Gain

InfoGainAttributeEval in WEKA selects attributes based on the amount of information gained with respect to the class. This was used with the Ranker search method, generating the following ranked attributes;



Filtering off any values less than 0.13, we can remove attributes below 9 (Data\_Value\_Type).

# 

# Part 4 - Classifier Models

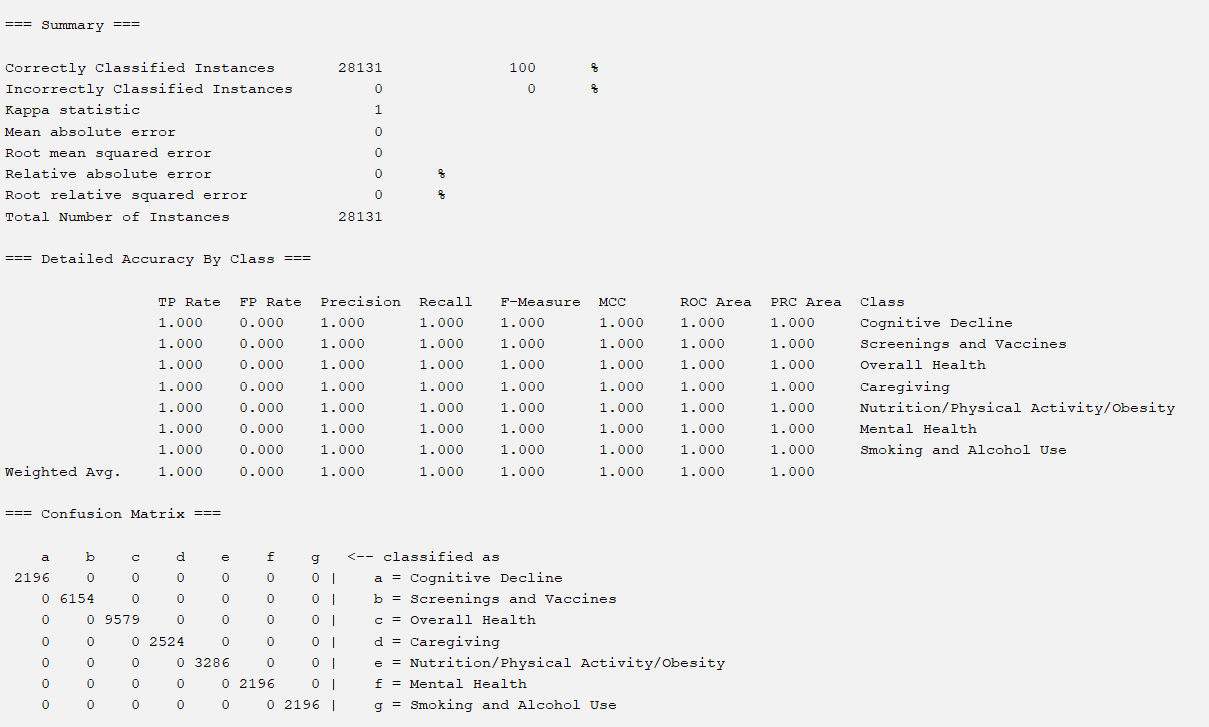
Out of the four classification models, we chose the rule-based methods of OneR and Decision Table, as well as the tree-based methods of J48 and Random Forests.

# **Part 5 – Results and Analysis**

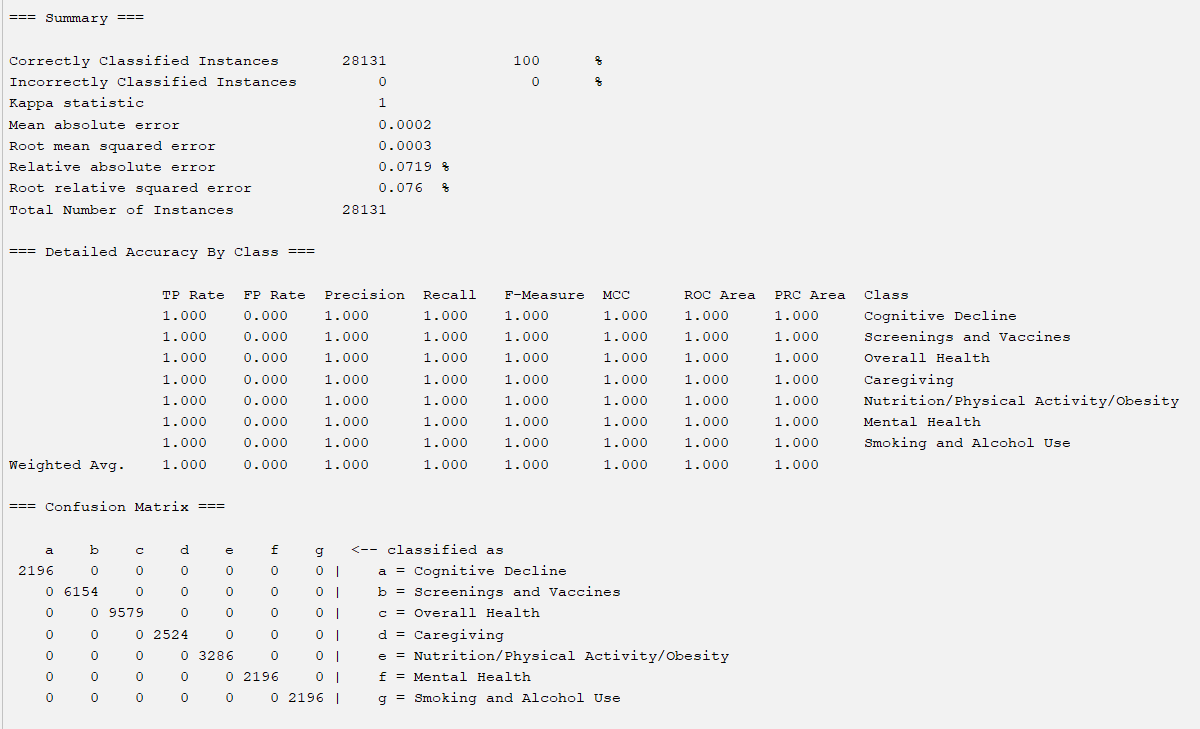
## 5.1 - Results

### **CLASSIFICATION WITH KENDALL’S CORRELATION-SELECTED ATTRIBUTES:**

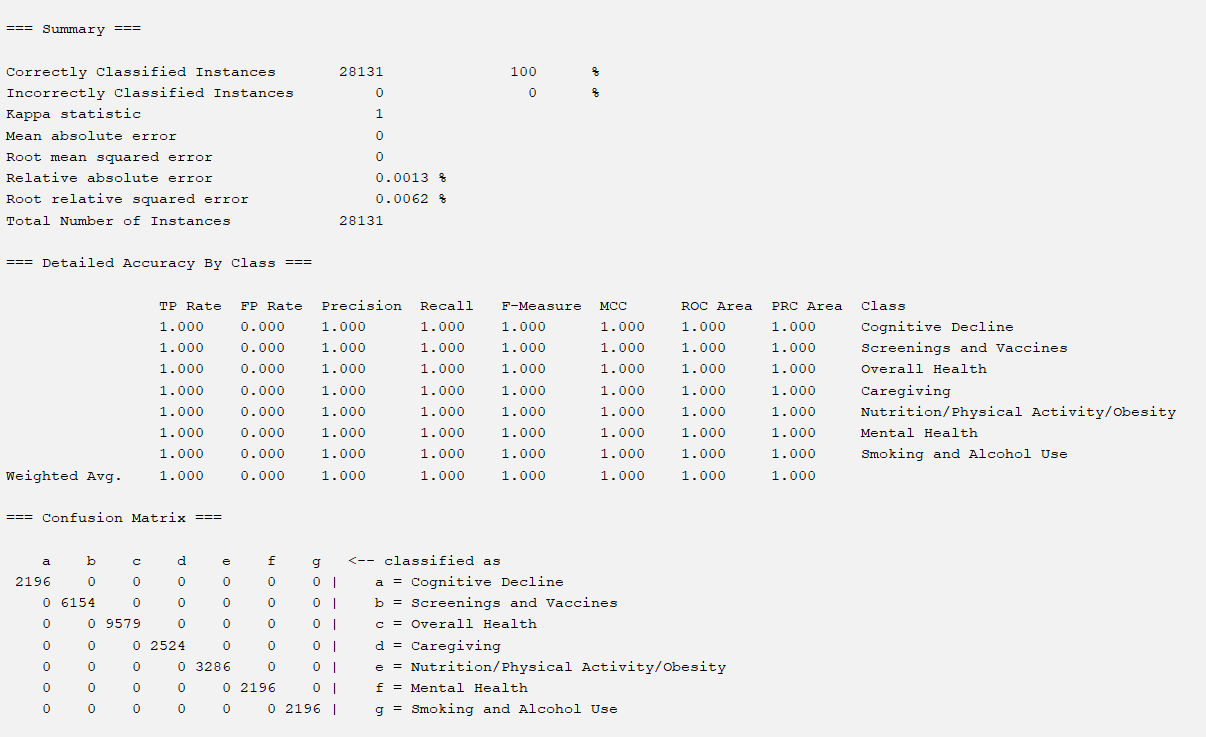
**Kendall’s Correlation (Non-Weka Attribute Selection) with OneR:**



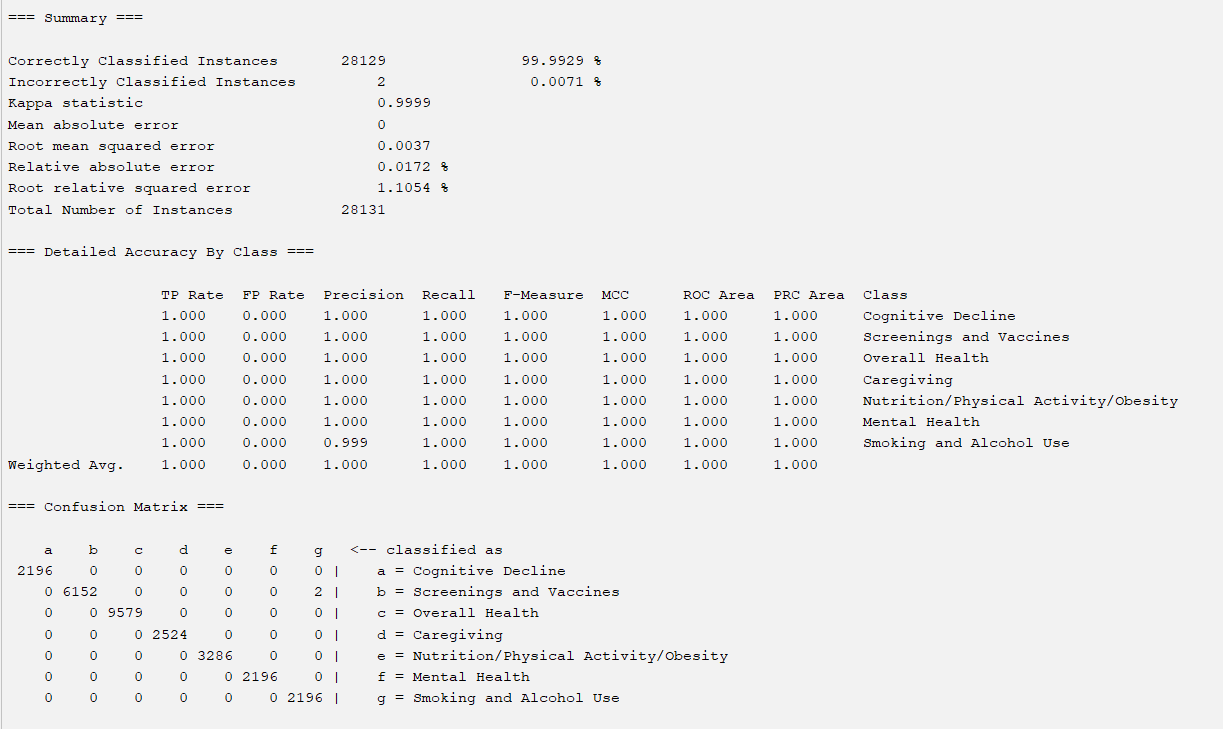
**Kendall’s Correlation (Non-Weka Attribute Selection with DecisionTable**

****

**Kendall’s Correlation (Non-Weka Attribute Selection with J48**

****

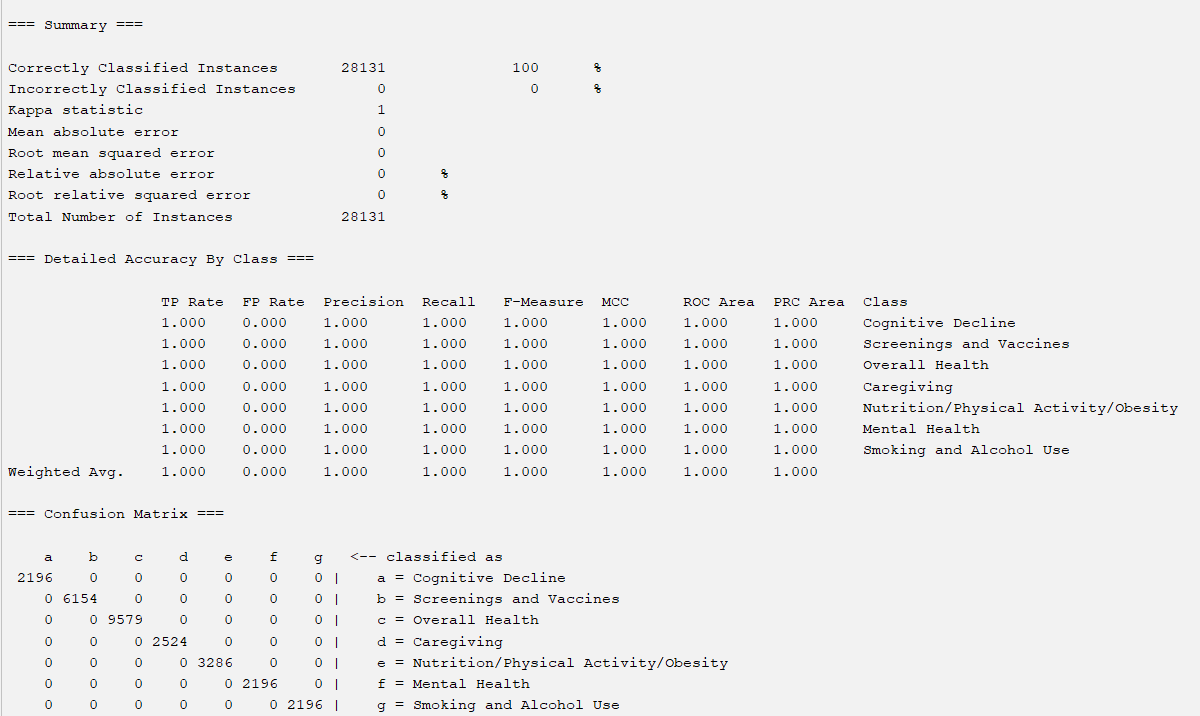
**Kendall’s Correlation (Non-Weka Attribute Selection with Random Forest**

****

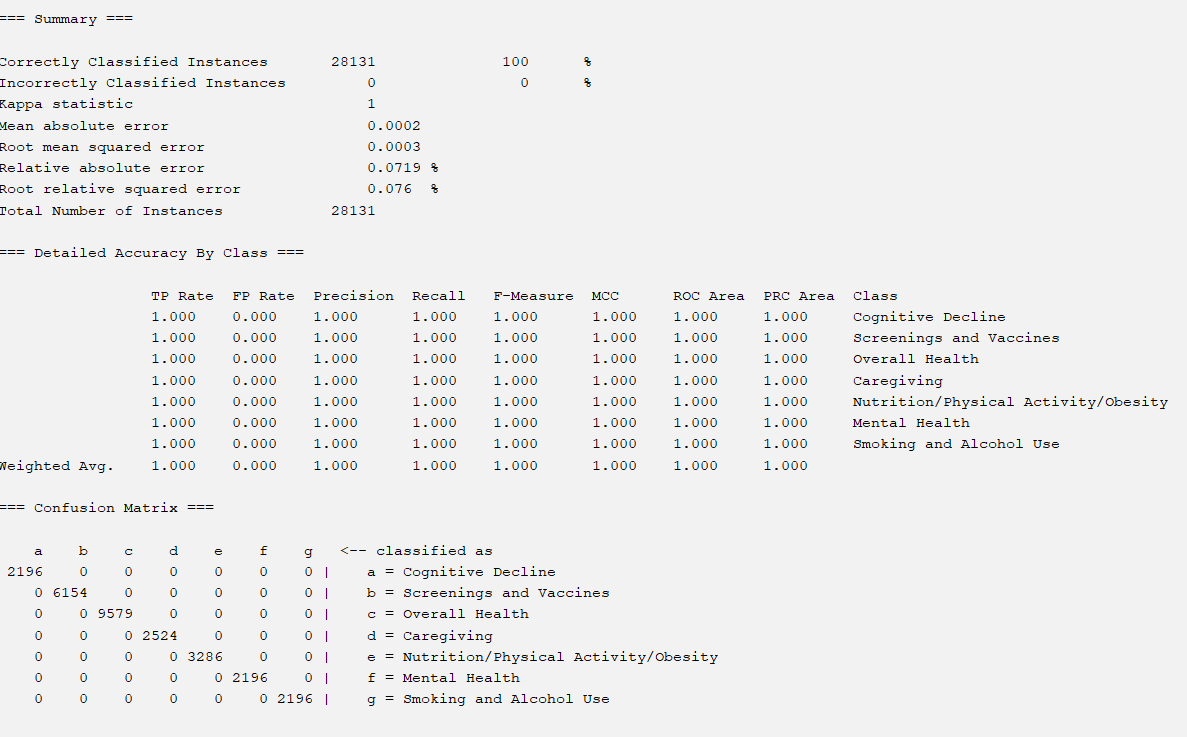
### 

### **CLASSIFICATION WITH PEARSON’S CORRELATION-SELECTED ATTRIBUTES:**

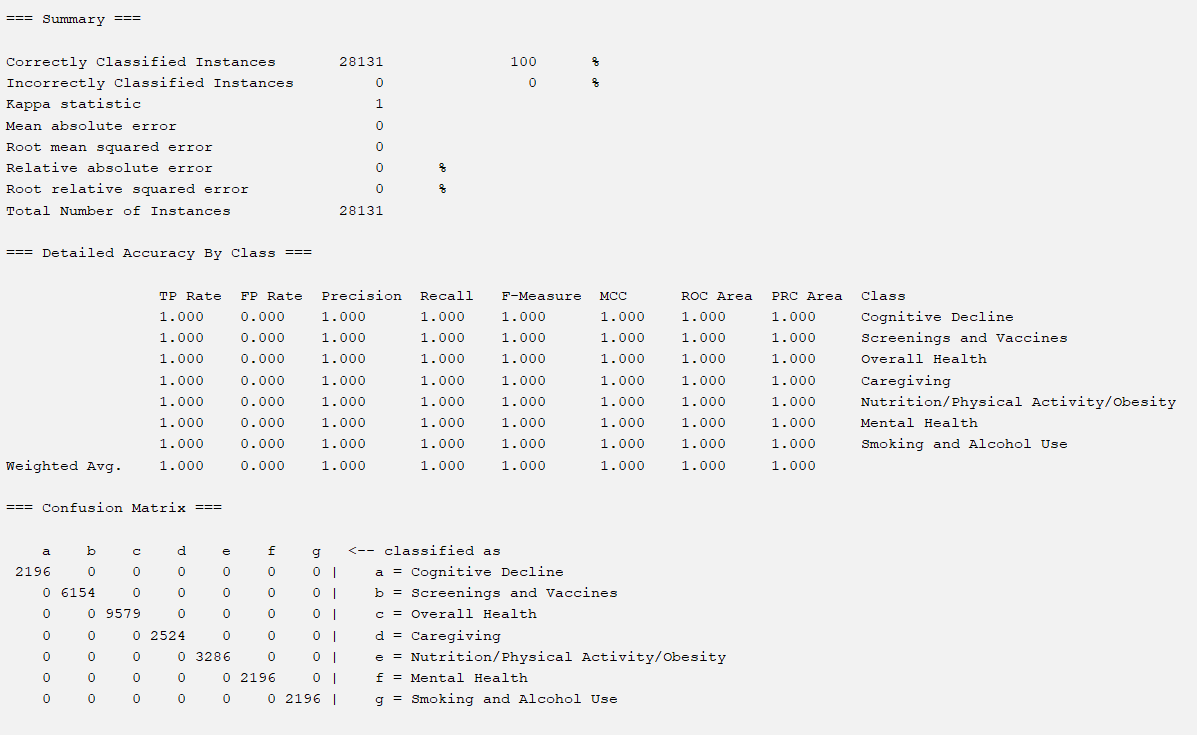
**CorrelationAttributeEval with OneR:**

****

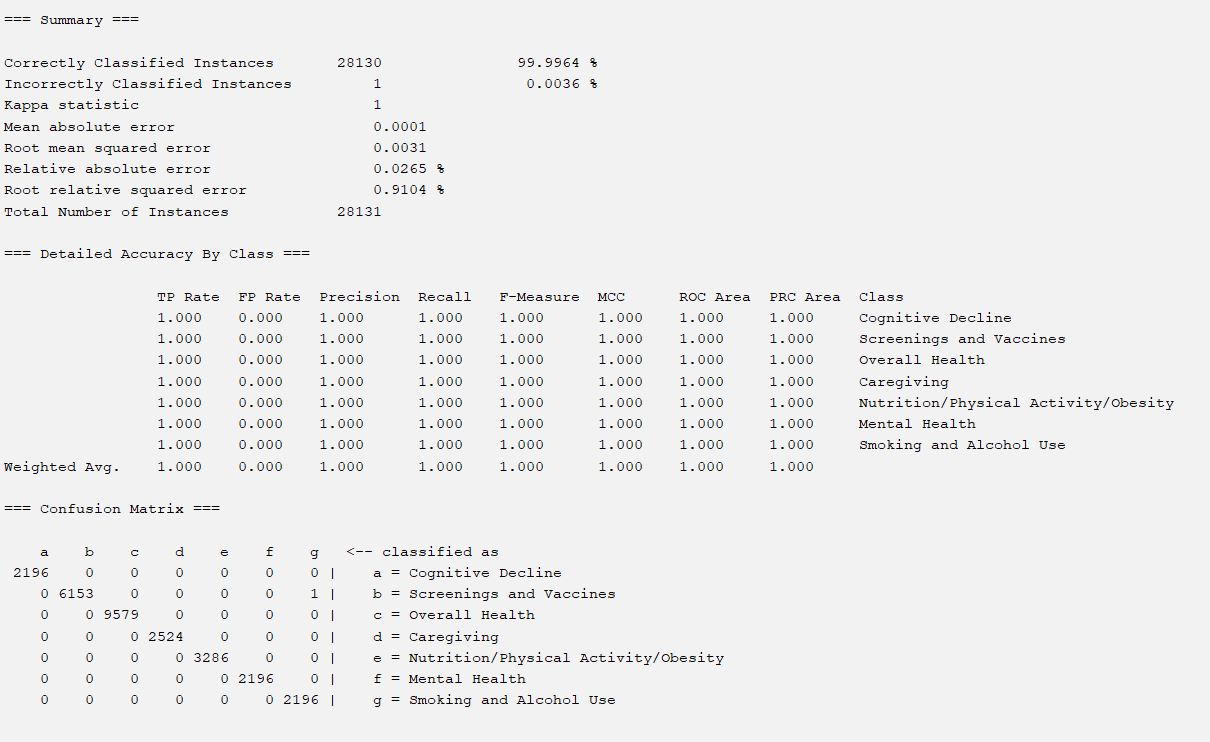
**CorrelationAttributeEval with DecisionTable**



**CorrelationAttributeEval with J48**

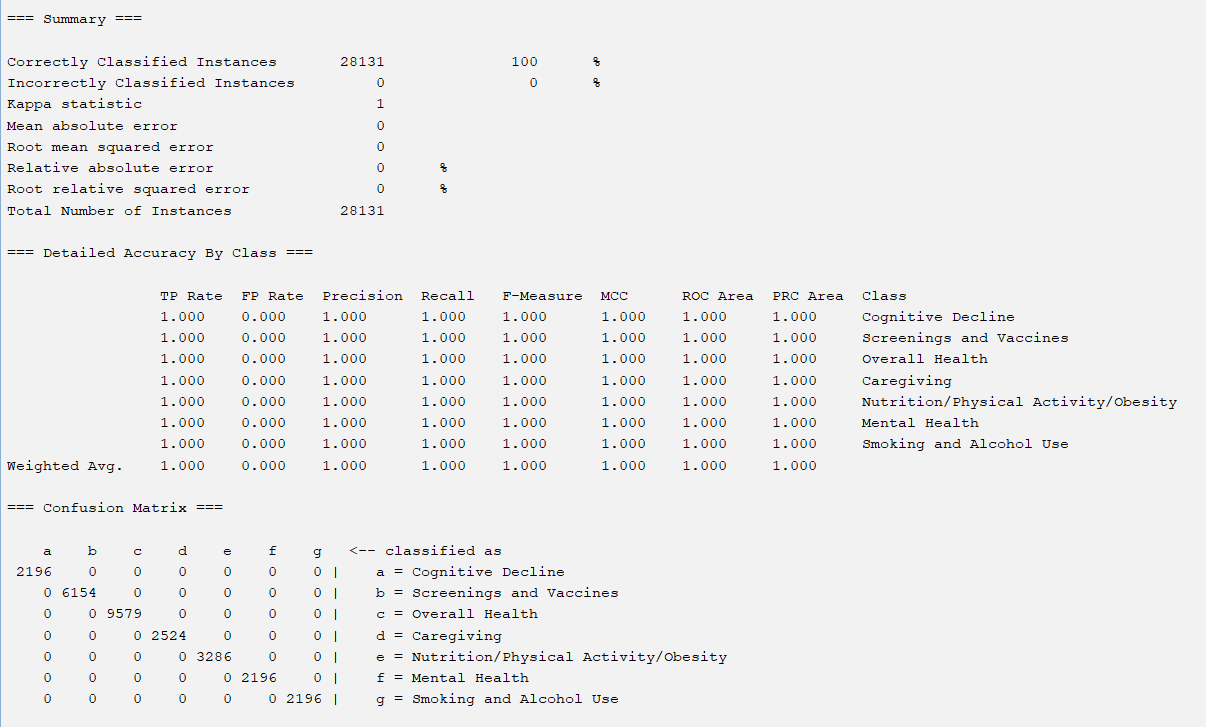
****

**CorrelationAttributeEval with Random Forest**

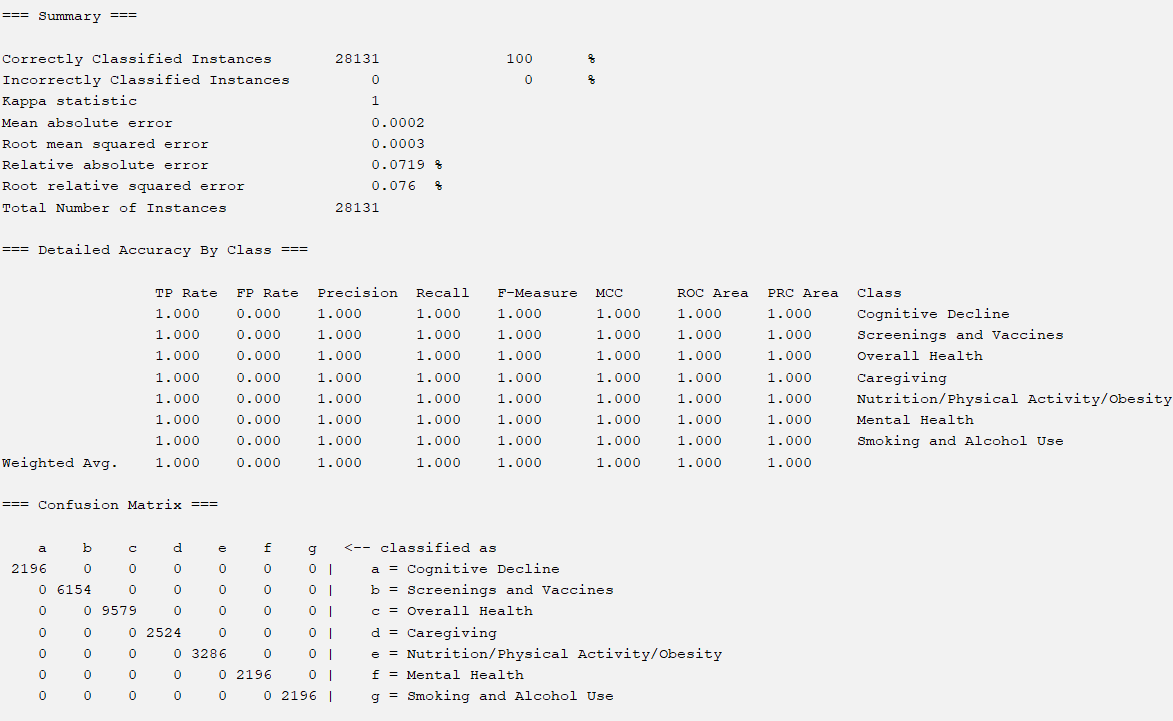


### **CLASSIFICATION WITH SUBSET ANALYSIS-SELECTED ATTRIBUTES:**

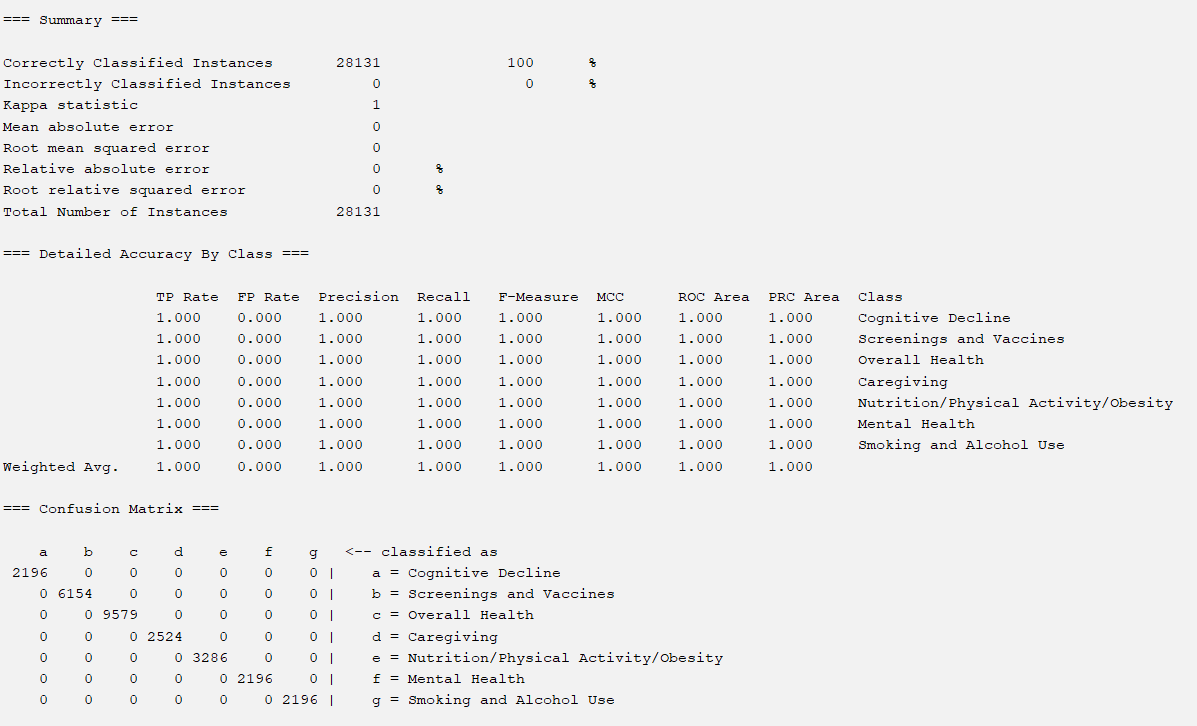
**CfsSubsetEval with OneR:**

****

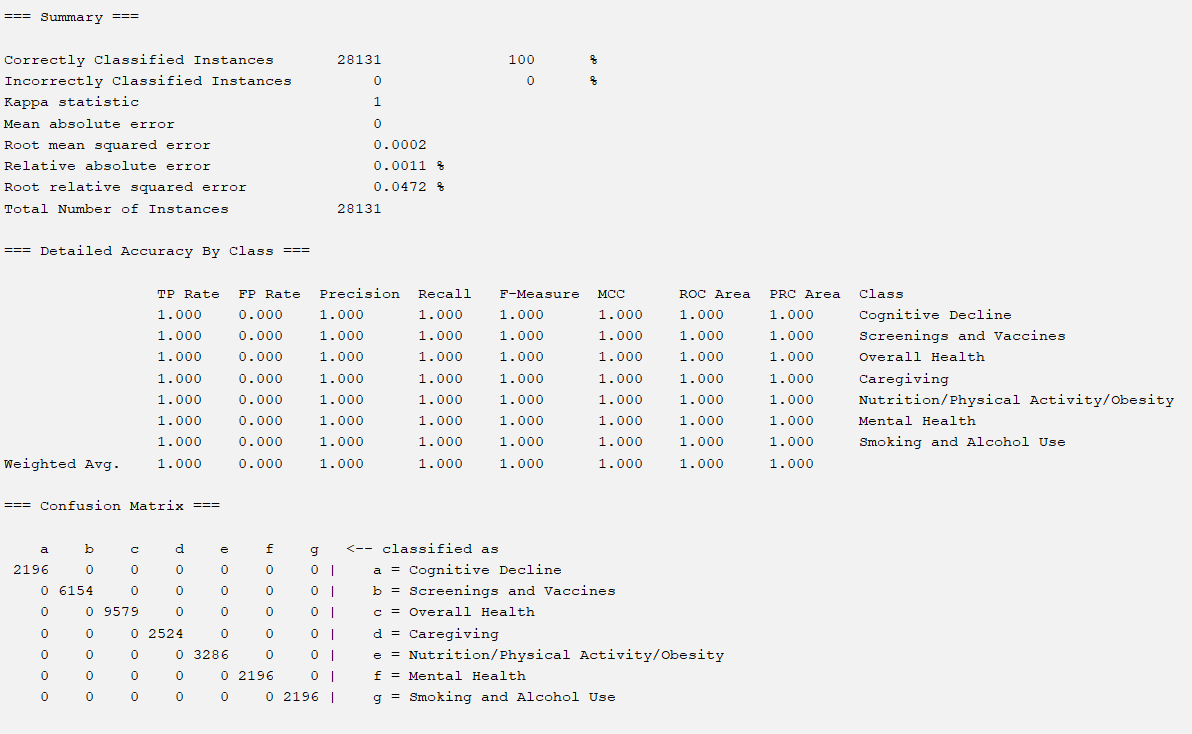
**CfsSubsetEval with DecisionTable:**



**CfsSubsetEval with J48:**

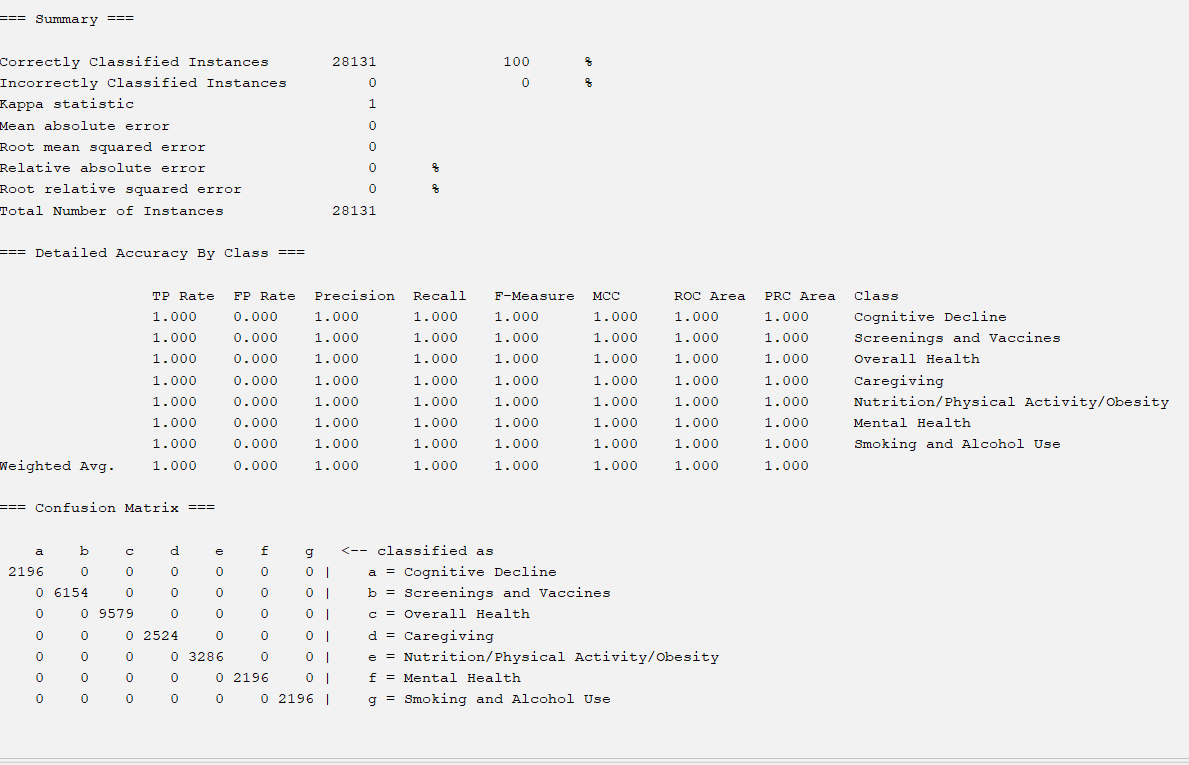


**CfsSubsetEval with Random Forest:**

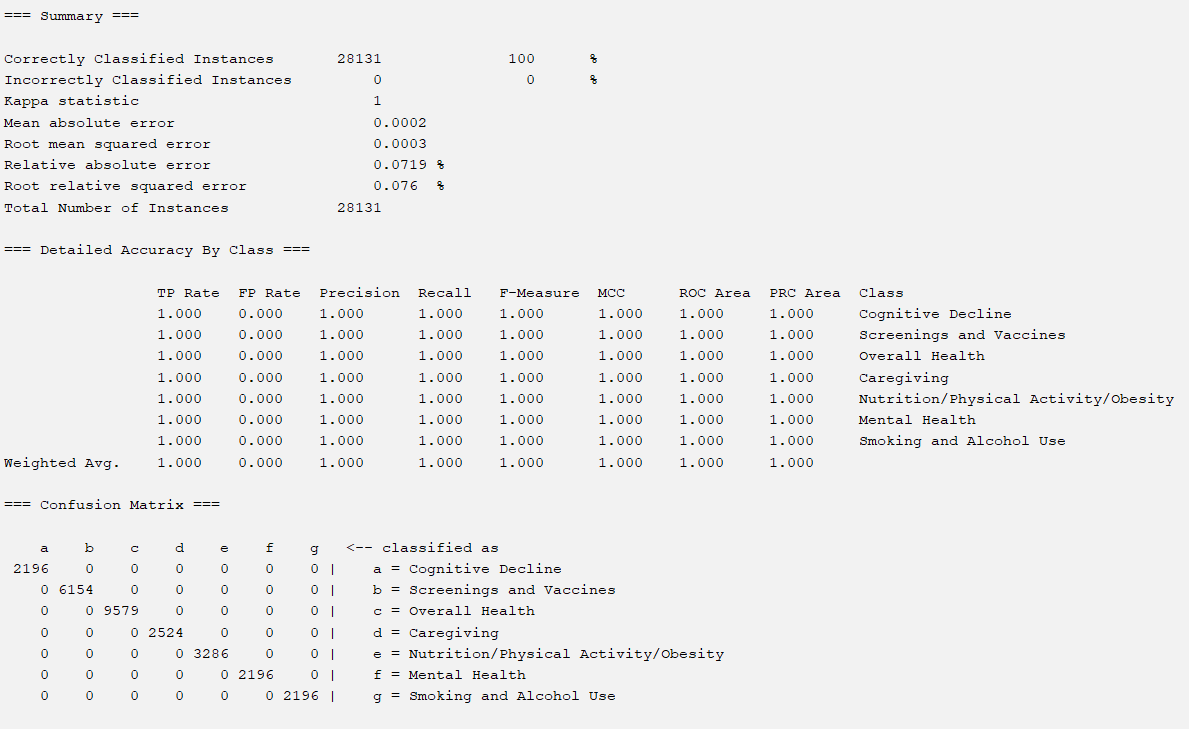


### **CLASSIFICATION WITH ONER-SELECTED ATTRIBUTES:**

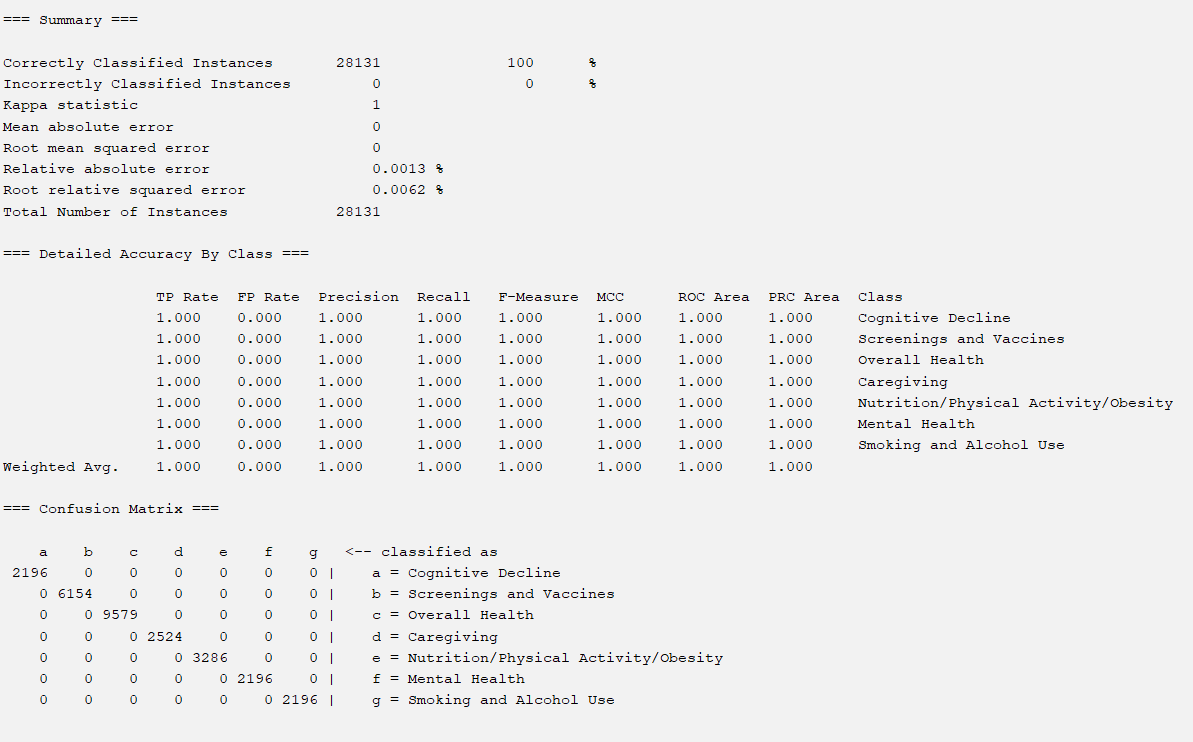
**OneRAttributeEval with OneR:**

****

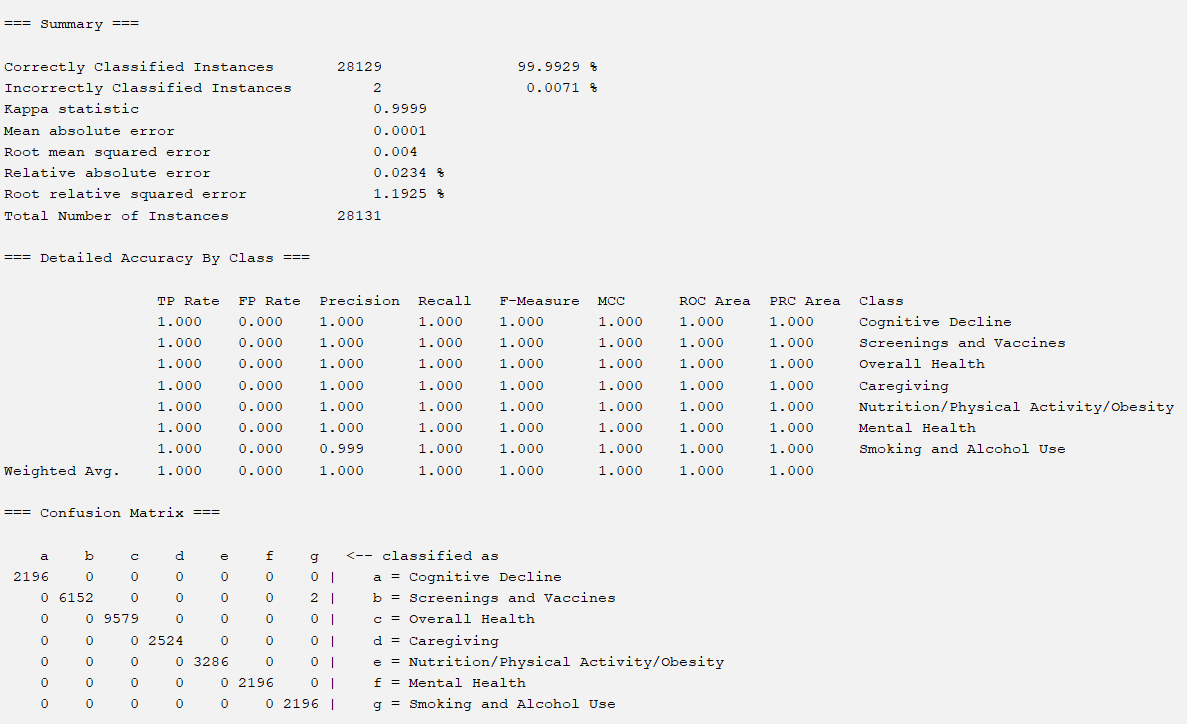
**OneRAttributeEval with DecisionTable**



**OneRAttributeEval with J48**

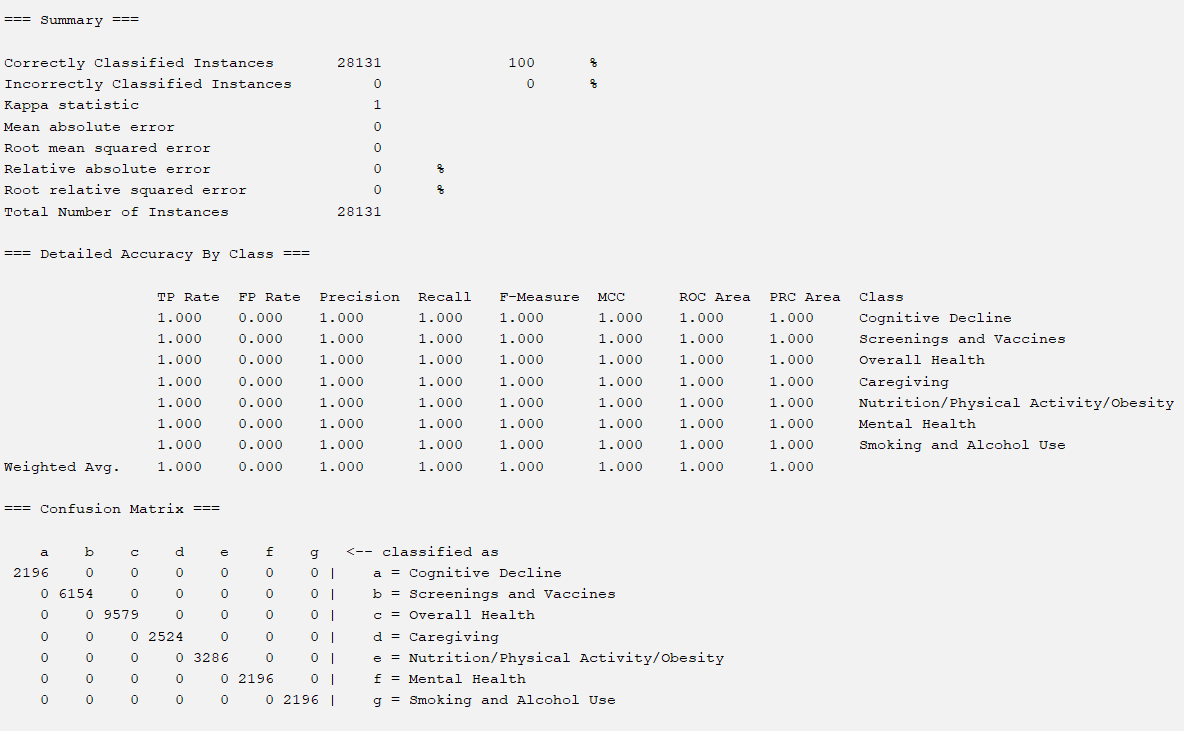


**OneRAttributeEval with Random Forest**

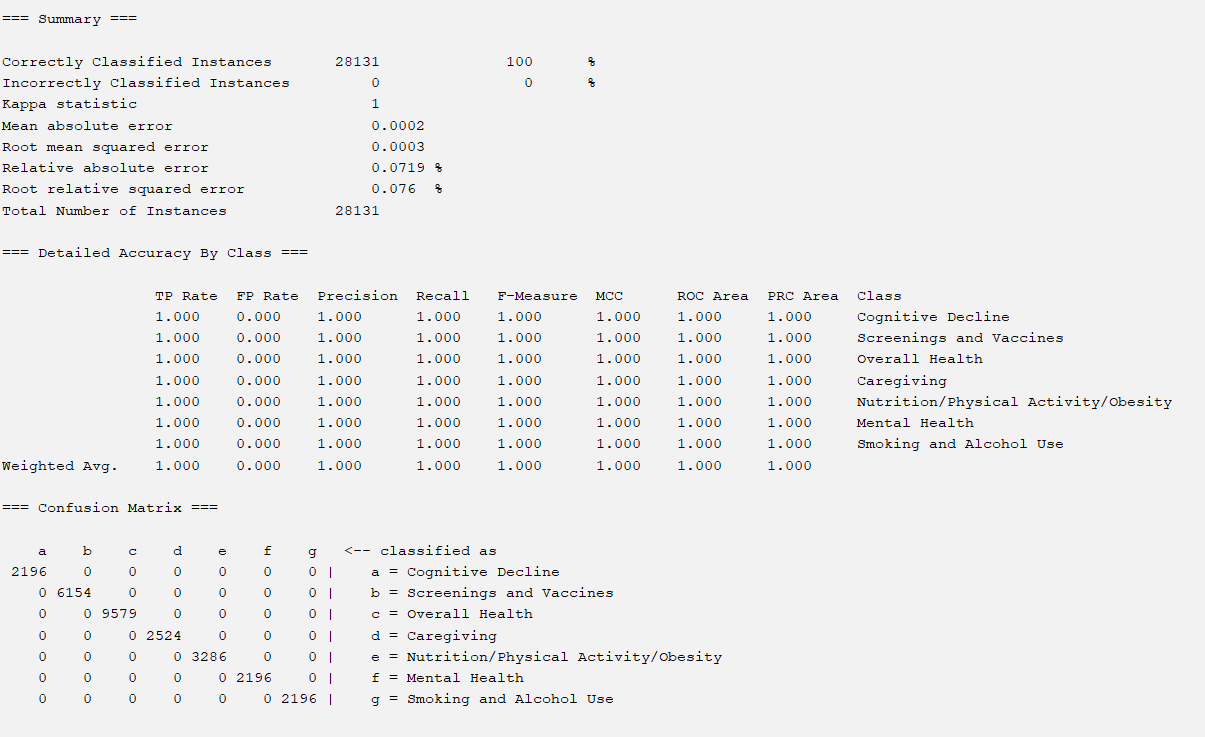
****

### **CLASSIFICATION WITH ENTROPY GAIN-SELECTED ATTRIBUTES:**

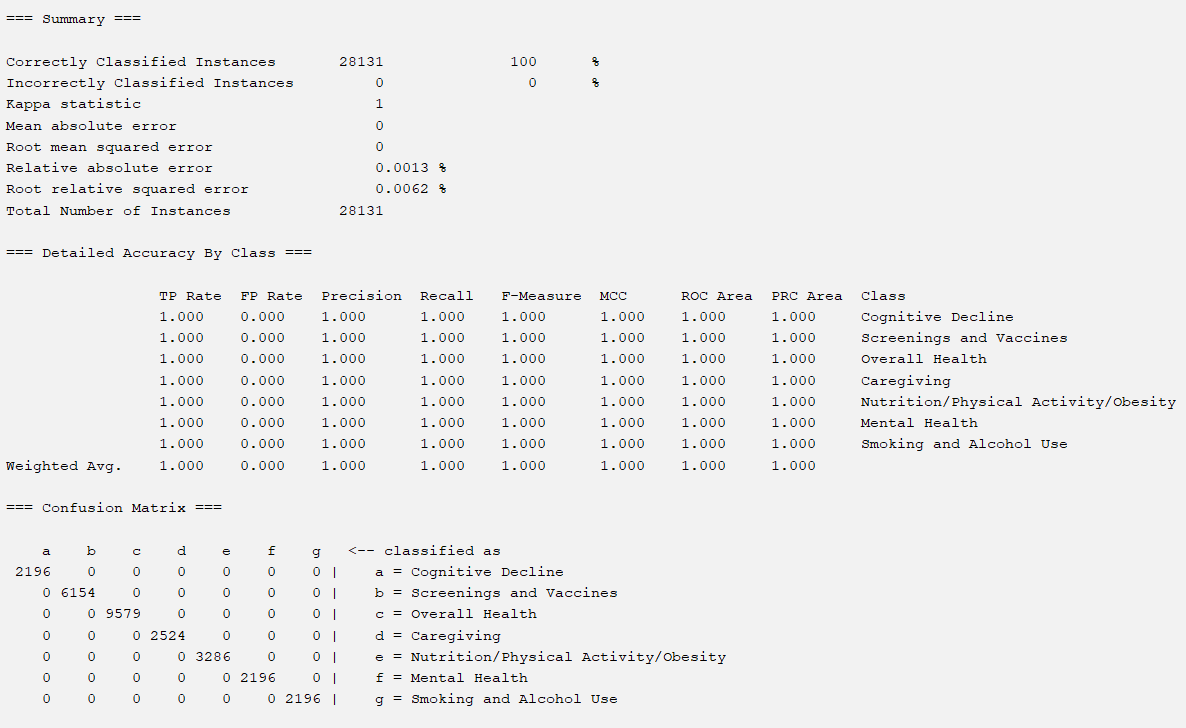
**InfoGainAttributeEval with OneR**



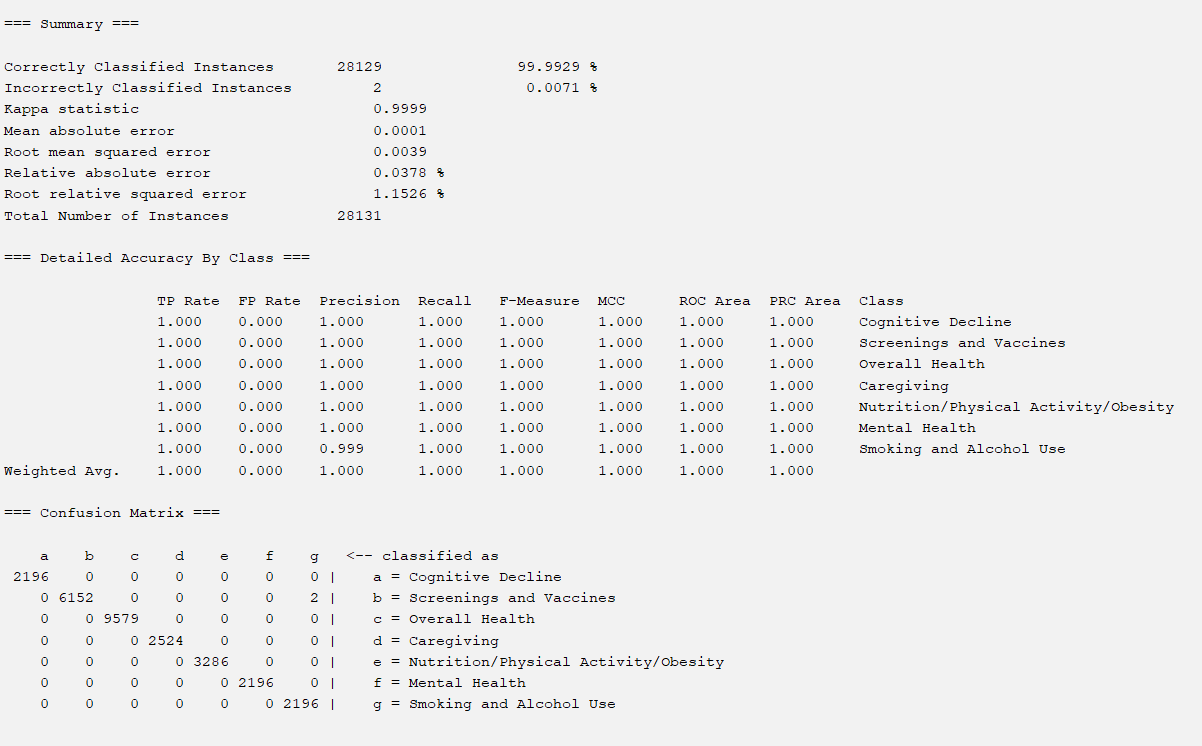
**InfoGainAttributeEval with Decision Table**

****

**InfoGainAttributeEval with J48**

****

**InfoGainAttributeEval with Random Forest:**

****

## 5.2 - Analysis

Based on five unique datasets selected by different attribute selection algorithms, four classification algorithms were applied to each dataset. Two of these algorithms were rule-based, which were OneR and decision table classification. The other two classification algorithms, J48 and random forests, were tree-based. This resulted in a total of twenty separate results. Out of these results, many scored 100% accuracy, with others in the 99-100% range. These algorithms that yielded 100% accuracy are as follows:

* Kendall’s Correlation Attributes
  + OneR
  + Decision Table
  + J48
* Pearson’s Correlation Attributes
  + OneR
  + Decision Table
  + J48
* Subset Analysis Attributes
  + OneR
  + Decision Table
  + J48
  + Random Forest
* OneR Algorithm Attributes
  + OneR
  + Decision Table
  + J48
* Entropy/Information Gain Attributes
  + OneR
  + Decision Table
  + J48

| **Att. Selection Algorithm** | **Classification Algorithm** | Accuracy | MAE | RMSE | Precision | Recall |
| --- | --- | --- | --- | --- | --- | --- |
| Kendall’s Rank (Non-Weka) | OneR | 100% | 0 | 0 | 1.0 | 1.0 |
| Decision Table | 100% | 0.0002 | 0.0003 | 1.0 | 1.0 |
| J48 | 100% | 0 | 0 | 1.0 | 1.0 |
| Random Forest | 99.9929% | 0 | 0.0037 | 1.0 (0.99 in one category) | 1.0 |
| Pearson’s Correlation | OneR | 100% | 0 | 0 | 1.0 | 1.0 |
| Decision Table | 100% | 0 | 0 | 1.0 | 1.0 |
| J48 | 100% | 0 | 0 | 1.0 | 1.0 |
| Random Forest | 99.9964% | 0.0001 | 0.0031 | 1.0 | 1.0 |

| Att. Selection Algorithm | Classification Algorithm | Accuracy | MAE | RMSE | Precision | Recall |
| --- | --- | --- | --- | --- | --- | --- |
| Subset Analysis | OneR | 100% | 0 | 0 | 1.0 | 1.0 |
| Decision Table | 100% | 0.0002 | 0.0003 | 1.0 | 1.0 |
| J48 | 100% | 0 | 0 | 1.0 | 1.0 |
| Random Forest | 100% | 0.0002 | 0.0011 | 1.0 | 1.0 |
| OneR Algorithm | OneR | 100% | 0 | 0 | 1.0 | 1.0 |
| Decision Table | 100% | 0.002 | 0.003 | 1.0 | 1.0 |
| J48 | 100% | 0 | 0 | 1.0 | 1.0 |
| Random Forest | 99.9929% | 0.0001 | 0.004 | 1.0 (0.99 in one class) | 1.0 |

| Att. Selection Algorithm | Classification Algorithm | Accuracy | MAE | RMSE | Precision | Recall |
| --- | --- | --- | --- | --- | --- | --- |
| Information Gain | OneR | 100% | 0 | 0 | 1.0 | 1.0 |
| Decision Table | 100% | 0.0002 | 0.0003 | 1.0 | 1.0 |
| J48 | 100% | 0 | 0 | 1.0 | 1.0 |
| Random Forest | 99.9929% | 0.0001 | 0.0039 | 1.0 (0.99) in a class | 1.0 |

From the above, we can immediately notice that the rule-based classification algorithms resulted in 100% accuracy for all given datasets. This makes sense especially when we examine the attributes common in many of the datasets. The table below provides a recap of what attributes specifically were chosen from each attribute selection algorithm :

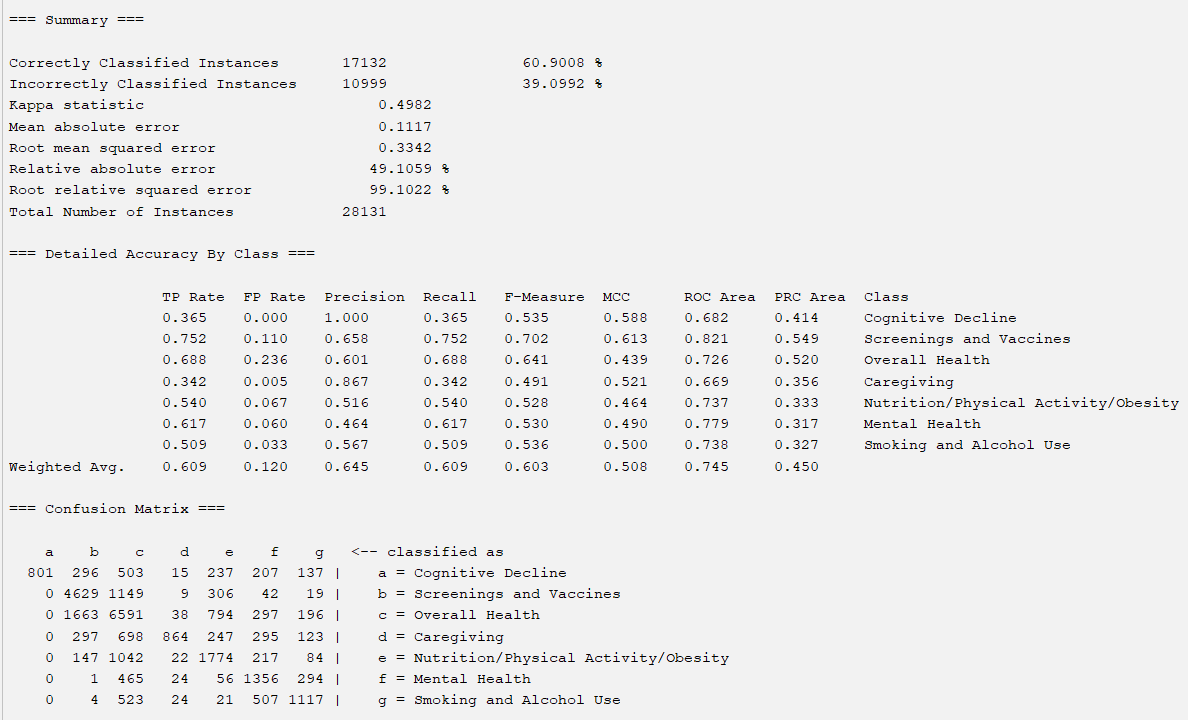
| **Kendall’s Correlation** | **Pearson’s Correlation** | **Subset Analysis** | **OneR Algorithm** | **Entropy/Info Gain** |
| --- | --- | --- | --- | --- |
| Topic | Low\_Confidence\_Limit | Topic | Topic | Topic |
| Question | Question | Stratification2 | Question | Question |
| Data\_Value | Topic | Geolocation | Data\_Value | Data\_Value |
| Low\_Confidence\_Limit | Data\_Value\_Type |  | Low\_Confidence\_Limit | Low\_Confidence\_Limit |
| High\_ConfidencE\_Limit | Data\_Value\_Unit |  | High\_Confidence\_Limit | High\_Confidence\_Limit |
|  | DataValueTypeID |  | Geolocation | Geolocation |
|  | Data\_Value |  |  | DataValueTypeID |
|  |  |  |  | Data\_Value\_Unit |
|  |  |  |  | Data\_Value\_Type |

With the exception of the Subset Analysis, all other attribute selection algorithms used a ranking evaluator, and the above attributes are presented in order of this rank. Highlighted in red in the table above, ‘Topic’ was chosen in all attribute selection algorithms, with relatively- consistent importance in each of the algorithms that considered ranking. At a close second, we can also see that ‘Question’ is selected at a relatively-high importance in all attribute selection algorithms aside from Subset Analysis.

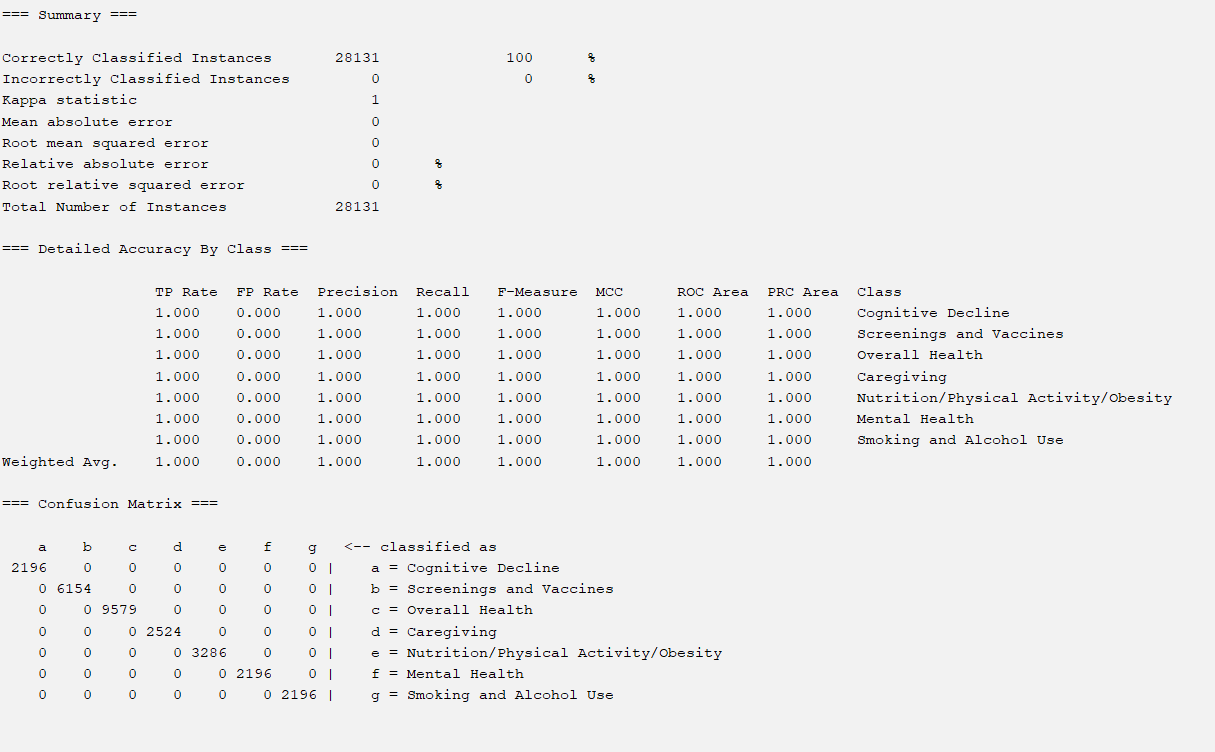
High correlation between ‘Topic’ and ‘Question’ with class is reasonable in context, as while a class may have multiple values for ‘Topic’ and ‘Question’ associated with it, a single topic or question in either attribute will only be associated with one class.

For example, values in ‘Topic’ such as ‘Ever had pneumococcal vaccine’ and ‘Mammogram within past 2 years’ are both associated uniquely with the ‘Screenings and Vaccines’ class. Similarly, values in ‘Question’ such as ‘Percentage of risk adults (have diabetes asthma cardiovascular disease or currently smoke) who ever had a pneumococcal vaccine’ and ‘Percentage of older adult men who are up to date with select clinical preventive services’ are also both associated solely to the ‘Screenings and Vaccines’ class. For a rule-based method such as OneR, it would then be incredibly simple to assign each unique value in an attribute to a class with no error, as each value could only map to one class. In the general context of a survey, such mappings are expected, as questions asked are likely to have been delivered with the intention of inquiring about a specific risk factor or class.

Considering that in context ‘Topic’ and ‘Question’ may have an abnormally-high level of correlation with the class, we experimented with removing these attributes from one dataset and testing its performance with the same model. We randomly chose to test this on the Kendall’s Correlation dataset on the OneR classification model. After removing ‘Topic’ and ‘Question,’ we are left with the attributes of ‘Data\_Value’, ‘Low\_Confidence\_Limit’, and ‘High\_Confidence\_Limit’. In other words, we are left with purely numerical data, with no context or background of where the numbers may have come from. Running OneR on this new dataset gives the following:



We automatically see a large drop in accuracy: 100% to 60.9008%. This decrease is not shown if only one of ‘Topic’ OR ‘Question’ is removed, such as in the below run, when ONLY ‘Topic’ was removed, and the OneR classification algorithm was run:



From these results, and by knowing the context of the data in ‘Topic’ and ‘Question’, we can assume that much of the high accuracies shown throughout all twenty classification algorithms are caused by the unbalanced importance of either ‘Topic’ or ‘Question’ on the class.

Although many algorithms reached 100% accuracy, a subset of these algorithms can be judged to be better than others in this group based on other error metrics such as MSE, RMSE, and RAE. Particularly, some algorithms can be judged to be the best based on having 0 error across MSE, RMSE, and RAE. These algorithms also reached perfect accuracy based on other metrics such as TP Rate, FP Rate, precision and recall. Keeping in mind that certain attributes such as ‘Topic’ and ‘Question’ may be disproportionately-correlated with the class as compared to other attributes, we also repeated evaluation on all datasets when removing ‘Topic’ and ‘Question,’ as we suspected their high correlation may have contributed to providing a misleading accuracy.

## 5.3 - Results Without Topic & Question

| **Att. Selection Algorithm** | **Classification Algorithm** | Accuracy | MAE | RMSE | Precision | Recall |
| --- | --- | --- | --- | --- | --- | --- |
| Kendall’s Rank (Non-Weka) | OneR | 60.9002% | 0.1117 | 0.3342 | 0.645 | 0.609 |
| Decision Table | 62.2445% | 0.1254 | 0.2505 | 0.626 | 0.622 |
| J48 | 62.2516% | 0.1193 | 0.2646 | 0.621 | 0.623 |
| Random Forest | 61.5655% | 0.1153 | 0.2705 | 0.615 | 0.616 |
| Pearson’s Correlation | OneR | 60.9008% | 0.1117 | 0.3342 | 0.645 | 0.609 |
| Decision Table | 63.8904% | 0.1203 | 0.2447 | 0.649 | 0.639 |
| J48 | 63.318% | 0.1172 | 0.2556 | 0.632 | 0.633 |
| Random Forest | 61.5691% | 0.1144 | 0.2683 | 0.616 | 0.616 |

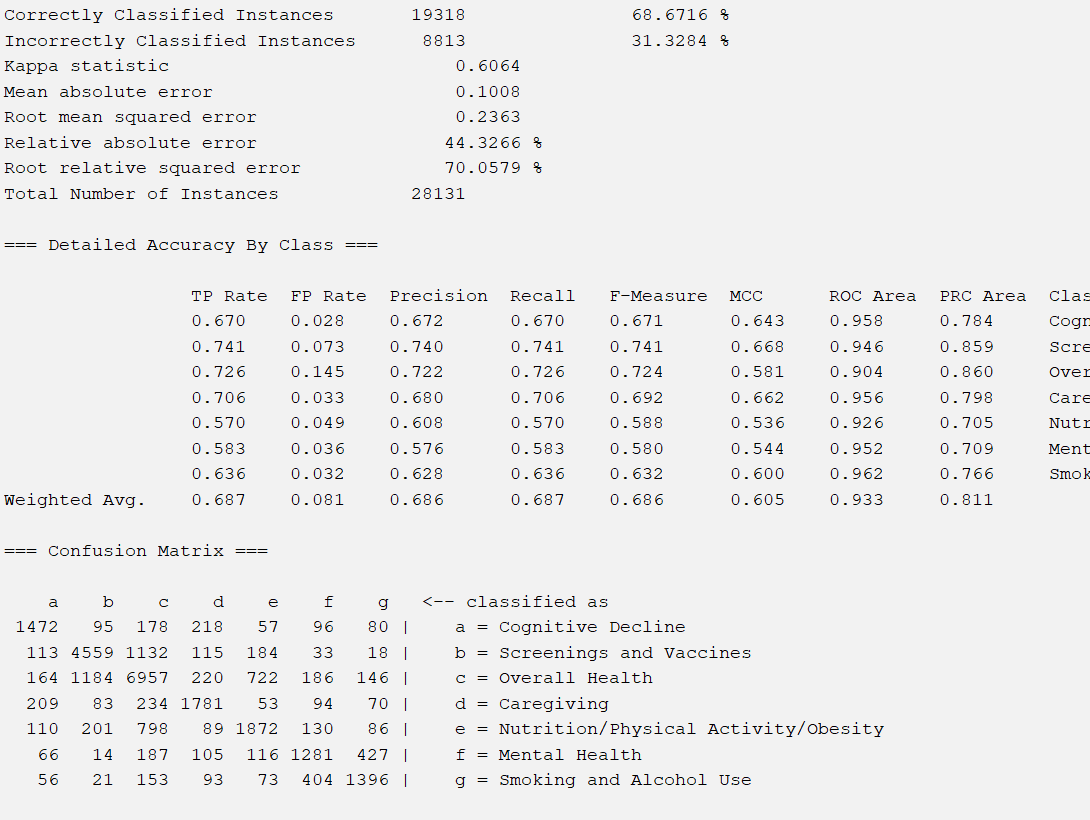
| **Att. Selection Algorithm** | **Classification Algorithm** | Accuracy | MAE | RMSE | Precision | Recall |
| --- | --- | --- | --- | --- | --- | --- |
| Subset Analysis | OneR | 37.7022% | 0.178 | 0.4219 | ? | 0.377 |
| Decision Table | 39.085% | 0.2152 | 0.3279 | ? | 0.391 |
| J48 | 39.085% | 0.2175 | 0.3287 | ? | 0.391 |
| Random Forest | 39.117% | 0.2151 | 0.3282 | ? | 0.391 |
| OneR Algorithm | OneR | 66.0268% | 0.1117 | 0.3342 | 0.645 | 0.609 |
| Decision Table | 66.0268% | 0.1237 | 0.243 | 0.661 | 0.660 |
| J48 | 67.5767% | 0.1026 | 0.2588 | 0.675 | 0.676 |
| Random Forest | 68.405% | 0.1017 | 0.2391 | 0.683 | 0.684 |

| Att. Selection Algorithm | Classification Algorithm | Accuracy | MAE | RMSE | Precision | Recall |
| --- | --- | --- | --- | --- | --- | --- |
| Information Gain | OneR | 60.9008% | 0.1117 | 0.3342 | 0.645 | 0.609 |
| Decision Table | 66.9724% | 0.1218 | 0.24 | 0.672 | 0.670 |
| J48 | 68.2663% | 0.1003 | 0.2554 | 0.682 | 0.683 |
| Random Forest | 68.6716% | 0.1008 | 0.2363 | 0.686 | 0.687 |

# 

Initially, when running the results, we noticed that only algorithms run on the SubsetEval attributes had a ? for Precision. Looking at the above confusion matrix, we suspected this was due to precision being calculated over TP Rate, which is impossible to compute for 0.0 TP and FP Rate.

Furthemore, although it had slightly lower accuracy in the dataset with Topic and Question, we choose Information Gain-selected attributes with Random Forest as our best model, as it had the highest accuracy, precision, and recall and lowest error rates when removing attributes that may have led to “false” accuracies.



The above shows the results of our best model made with the classification algorithm RandomForest and the attribute selection model Information gain. We believe that this model performed the best because of its ability to account for more complex relationships, which is to be expected as more direct relationships expected from ‘Topic’ and ‘Question’ are removed.

# **Part 6 – Conclusion**

At the start of this project we were trepidatious newcomers, excited to learn more about classification. Throughout this project we were able to successfully create various models. We learned how to deal with qualitative data, maintain a stratified dataset, and perform correlation with a non-linear relationship. Our results show we need to observe data carefully. Achieving 100% accuracy is abnormal and could be avoided with close inspection. Another area of improvement is in preprocessing. When we normalized we used the zScore, but we could also use min-max and decimal scaling to replicate the exact scale. Future work could lead to further analysis of performance, model, and feature selector.

**Steps for Reproducibility:**

1. In Weka, load the dataset titled Alzh\_train.csv. This dataset automatically includes all preprocessing steps done in Python, as outlined above.
2. Under Select Attributes, under AttributeEvaluator, choose InfoGainAttributeEval. Weka should automatically select the Ranker Search method.
3. On the left side of the screen, select the down arrow by “No class” and select “(Nom) Class” as the class variable. Click start to begin attribute selection.
4. Noting down the attributes with a value of more than 0.13, go to the Preprocess tab and remove all attributes below 0.13 by selecting such attributes and removing them. Do not remove the class. The remaining attributes should be identical to those found in the file Alzh\_InfoTrain.arff.
5. Open Alzh\_test.csv and remove the same attributes removed in Alzh\_train.csv. The resulting dataset should be the same found in Alzh\_InfoTestarff.
6. Furthermore remove ‘Topic’ and ‘Question’ from both of the above training and testing sets if you wish to analyze without those two attributes. The resulting datasets should be equivalent to Alzh\_InfoTrain2.arff and Alzh\_InfoTest2.arff
7. Return to the Alzh\_InfoTrain2.arff dataset. Click on the Classify tab, and under Supplied Test Set, choose Alzh\_InfoTest2.arff.
8. Repeat step 3, choosing the correct class to be “Class”.
9. Select the Random Forest model under rules.
10. Click Start. This model can be found at Info\_RandomForest\_BestModel.model

# Part 7 - Team Members and Tasks Performed

**Finding the Data & Building Proposal:** Leah + Victoria

**Preprocessing Initial Attempt:** Leah

**Preprocessing & Project Update:** Leah + Victoria

**Non-Weka Attribute Selection Algorithm:** Leah

**Weka Attribute Selection Algorithms and Classifiers:** Victoria

**Results Output:** Victoria

**Results Analysis:** Victoria

**Re-doing Results with Revised Attributes:** Leah + Victoria

**Conclusion:** Leah

**Reproducibility:** Victoria

**Building Final Report:** Leah + Victoria

# Part 8 – Appendix and Sources:

**Data Source Website:** <https://chronicdata.cdc.gov/Healthy-Aging/Alzheimer-s-Disease-and-Healthy-Aging-Data/hfr9-rurv/about_data>

**Files Attached with the Report:**

* preprocess.py – Python code used to run through pre-processing
* non\_weka\_selection.py – Python code to run correlations between features for the non-Weka attribute selection.
* Alzheimer\_s\_Disease\_and\_Healthy\_Aging\_Data.csv – Before splitting train, test, and validation
* Alzh\_no\_na.csv - Data after splitting train, test, and validation.
* Alzh\_train.csv, Alzh\_test.csv, Alzh\_val.csv – Train, test, validation datasets before feature selection algorithms
* “Leah, Victoria - Q1 Project Report.docx”, “Leah, Victoria - Project Proposal”, “Leah, Victoria - Intermediate Report” - Final report, proposal, and intermediate report respectively
* ARFF Weka Files for all 5 attribute selection algorithms in Weka located in the folder “ML Q1 Post-Attribute Selection”.
* ARFF Weka Files for ALL training and testing sets used for all attribute selection models located in the folder “ML Q1 All Train/Test Sets”
* ARFF Weka files for ALL training and testing sets AFTER removing ‘Topic’ and ‘Question’ located in the folder “ML Q1 Revised Train/Test Sets”
* PearsonBest.model – the best model chosen, using attributes selected by Pearson’s Correlation with the OneR classification algorithm

# **References**

Alzheimer's Association. (2023). 2023 Alzheimer’s Disease Facts and Figures. *Alzheimer’s & Dementia*, *19*(4), 1598–1695. https://doi.org/10.1002/alz.13016

CDC Division of Population Health. (2024). *Alzheimer’s Disease and Healthy Aging Data | Data | Centers for Disease Control and Prevention*. Chronicdata.cdc.gov. https://chronicdata.cdc.gov/Healthy-Aging/Alzheimer-s-Disease-and-Healthy-Aging-Data/hfr9-rurv/about\_data

*CfsSubsetEval*. (n.d.). Weka.sourceforge.io. https://weka.sourceforge.io/doc.dev/weka/attributeSelection/CfsSubsetEval.html

GeeksForGeeks. (2020, May 15). *Python | Kendall Rank Correlation Coefficient*. GeeksforGeeks. https://www.geeksforgeeks.org/python-kendall-rank-correlation-coefficient/

*GreedyStepwise*. (2022, January 28). Sourceforge.io. https://weka.sourceforge.io/doc.dev/weka/attributeSelection/GreedyStepwise.html

https://www.health.com/condition/alzheimers-overview. (2022, August 15). *What Is Alzheimer’s Disease?* Health. https://www.health.com/condition/alzheimers-overview