# data exploration

Jorick Baron

14/9/2021

## Data.

### Intro.

GpC sites are areas in eukaryotic DNA where a G followed by a C get connected with a phosphate atom. These sites are often found in promoters near transcription start sites of genes. The sites are regulated by methylation and seem to have roles in cancer and gene silencing and maybe even memory. The methylation is influenced by age and smoking is also claimed to have influence.

### Codebook.

Below here you find a codebook explaining each variable.

```
codebook <- read.delim("codebook.txt", sep = ";")
kable(codebook, caption = "Table1: Codebook")
```

Table 1: Table1: Codebook

| Abriviation | Full.Name | Class | Unit | Description |
|---|---|---|---|---|
| GSM | GEO GSM sample number | string | NA | the number that can be used to obtain the full sample on NCBI. |
| Smokes | Smoking Status | string | curent/ never | if the patient smokes or not the value can be current or never. |
| Gender | Patient Gender | char | NA | The patient's gender. |
| Age | Patient Age | int | Years | The patient's age. |
| CG******* | CG******** Methylation Ratio | double | Ratio | Methylation rate of site ******* is replaced by it's number. |

### Reading the data.

Here we load in the data. We noticed when downloading not all genders had the same case so we made all genders upper-case.

```
my_data <- read.csv("data/Smoker_Epigenetic_df.csv")
my_data$Gender <- toupper(my_data$Gender)
```

Now we do a 5 number summary this can be very useful when trying to find faults in the data. For readability we will look at the first 8 values.

```
pander(summary(my_data[1:8]))
```

Table 2: Table continues below

| GSM | Smoking.Status | Gender | Age |
|---|---|---|---|
| Length:683 | Length:683 | Length:683 | Min. :18.00 |
| Class :character | Class :character | Class :character | 1st Qu.:47.00 |
| Mode :character | Mode :character | Mode :character | Median :56.00 |
| NA | NA | NA | Mean :53.82 |
| NA | NA | NA | 3rd Qu.:62.00 |
| NA | NA | NA | Max. :80.00 |
| NA | NA | NA | NA |

| cg00050873 | cg00212031 | cg00213748 | cg00214611 |
|---|---|---|---|
| Min. :0.1186 | Min. :0.00695 | Min. :0.0000 | Min. :0.01247 |
| 1st Qu.:0.4131 | 1st Qu.:0.06317 | 1st Qu.:0.3635 | 1st Qu.:0.06946 |
| Median :0.5052 | Median :0.36554 | Median :0.4713 | Median :0.41575 |
| Mean :0.5600 | Mean :0.30960 | Mean :0.5191 | Mean :0.34106 |
| 3rd Qu.:0.8144 | 3rd Qu.:0.45981 | 3rd Qu.:0.7278 | 3rd Qu.:0.49745 |
| Max. :0.8989 | Max. :0.70999 | Max. :0.9236 | Max. :0.80606 |
| NA's :62 | NA's :62 | NA's :62 | NA's :62 |

Furthermore we noticed that there are 62 NA Methylation Ratios whilst doing the summary and decided to remove these rows.

```
my_data <- na.omit(my_data)
```

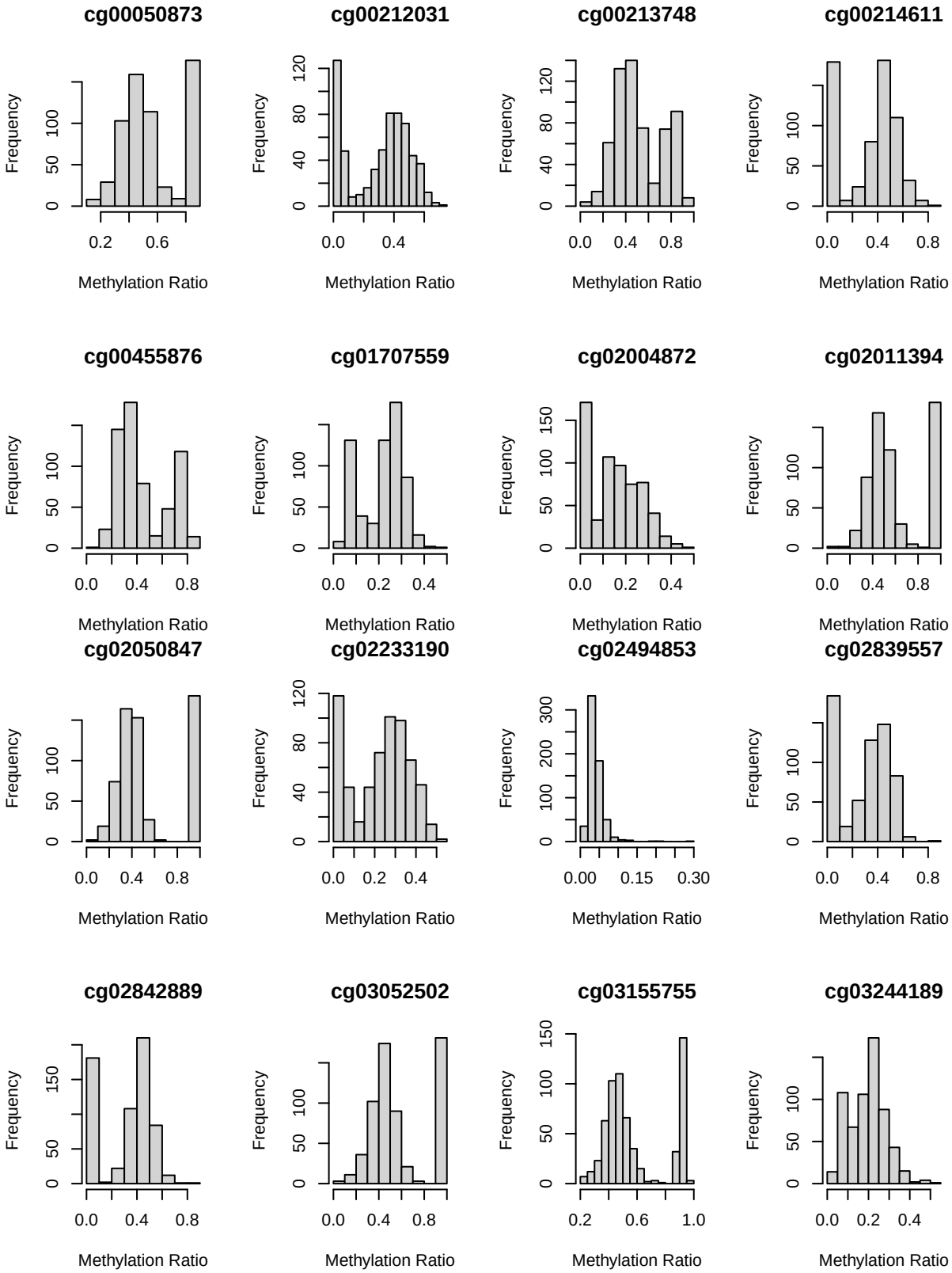
Looking more in depth at the summaries of the Methylation Ratios we notice that the median and mean not always align. This could be an indicator of skewed data.

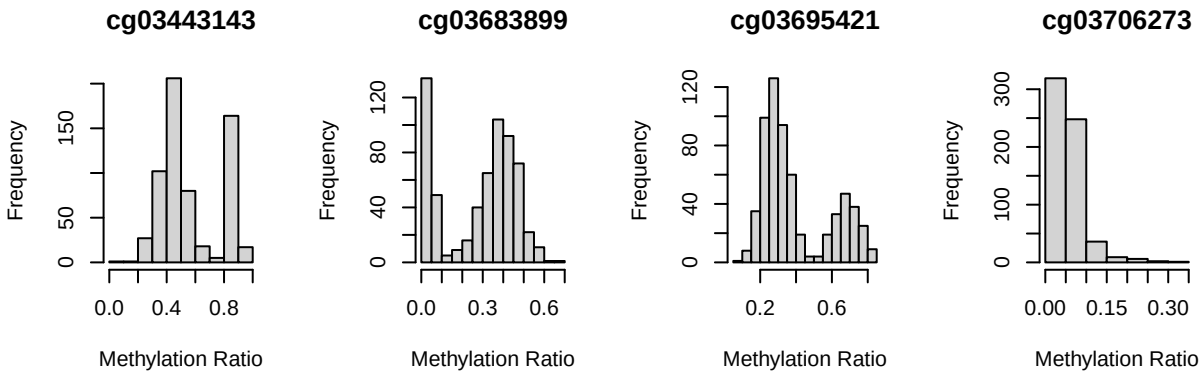
# Visualisation.

## Data Shape.

To determine if the data is skewed we will visualize the Methylation Ratio using a histogram.

```
par(mfrow = c(2,4))
for (name in colnames(my_data[5:24])) {
  hist(my_data[, name], main = name, xlab = "Methylation Ratio")
}
```


cg00050873
cg00212031
cg00213748
cg00214611
cg00455876
cg01707559
cg02004872
cg02011394
cg02050847
cg02233190
cg02494853
cg02839557
cg02842889
cg03052502
cg03155755
cg03244189
Frequency
Methylation Ratio

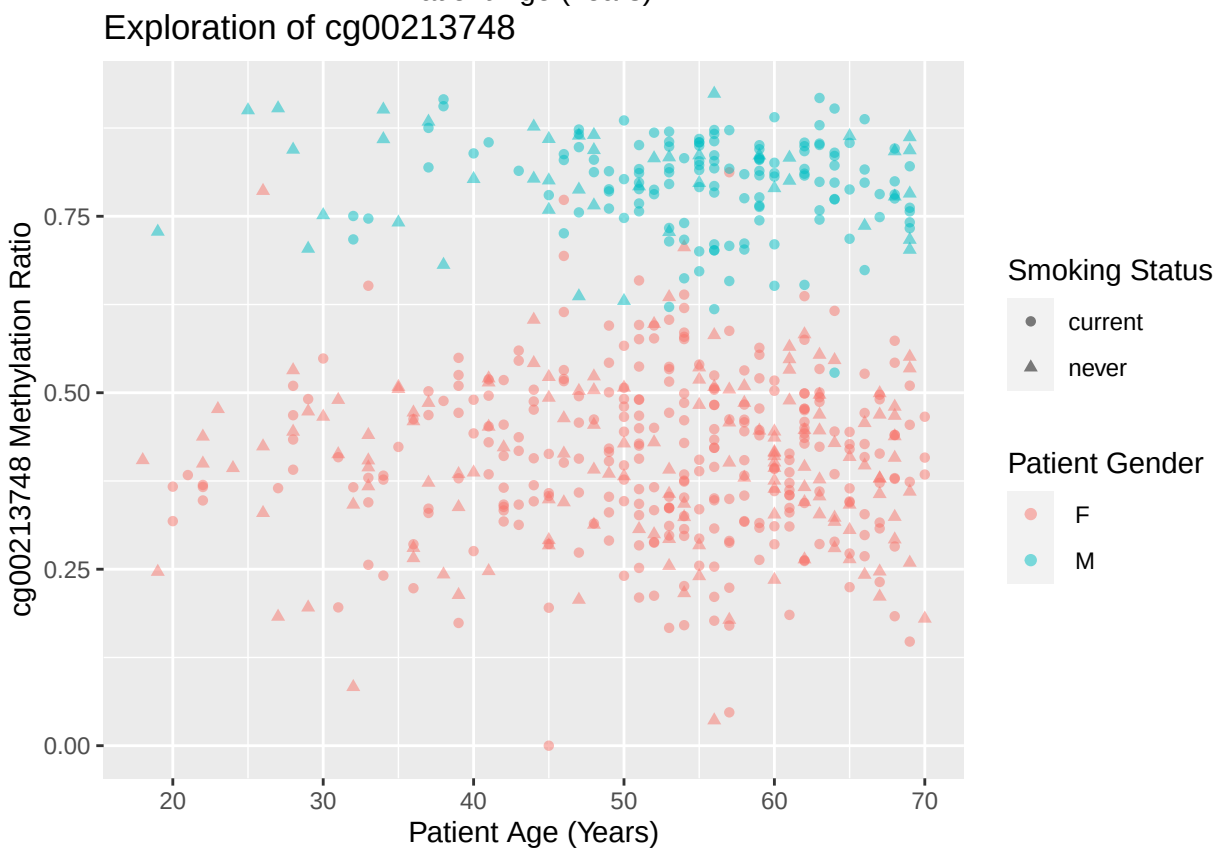



These histograms indicate that most data is not skewed but is in fact bimodal, this requires no further action now. However a couple of Methylation Ratios (cg02494853 & cg03706273) are skewed, further research is required to determine what actions will be taken if any.

## Relationships.

Now to see if the data contains any patterns we will make 4 dimensional scatter-plots containing the first 5 Methylation Ratios.


Exploration of cg00212031
cg00212031 Methylation Ratio
0.0
0.2
0.4
0.6
20
30
40
50
60
70
Patient Age (Years)
Smoking Status
current
never
Patient Gender
F
M

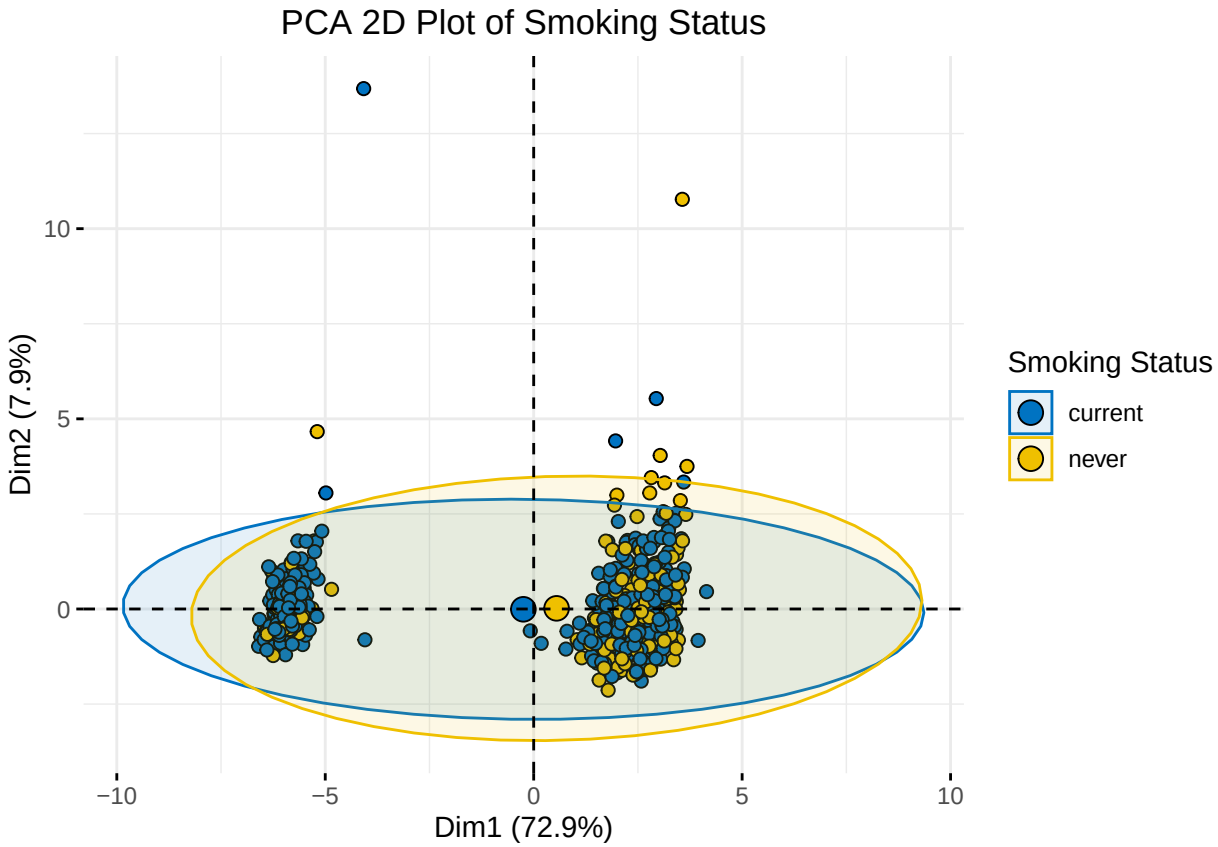

Exploration of cg00213748
cg00213748 Methylation Ratio
0.00
0.25
0.50
0.75
20
30
40
50
60
70
Patient Age (Years)
Smoking Status
current
never
Patient Gender
F
M


Exploration of cg00214611
cg00214611 Methylation Ratio
0.0
0.2
0.4
0.6
0.8
20
30
40
50
60
70
Patient Age (Years)
Smoking Status
current
never
Patient Gender
F
M


Exploration of cg00455876
cg00455876 Methylation Ratio
0.2
0.4
0.6
0.8
20
30
40
50
60
70
Patient Age (Years)
Smoking Status
current
never
Patient Gender
F
M

At a quick glance it seems that gender seems to be a strong predictor for Methylation, we hope that when we take a more in depth look at the data using machine learning that we can find more patterns. Furthermore the fact that the data is divided in two clusters also explains why the histograms display bimodal data.

```
pc_data <- my_data[5:24]
data.pr <- prcomp((pc_data), center = T, scale. = T)
fviz_pca_ind(data.pr, geom.ind = "point", pointshape = 21,
             pointsize = 2,
             fill.ind = my_data$Smoking.Status,
             col.ind = "black",
             palette = "jco",
             addEllipses = T,
             label = "var",
             col.var = "black",
             repel = T,
             legend.title = "Smoking Status") +
  ggtitle("PCA 2D Plot of Smoking Status") +
  theme(plot.title = element_text(hjust = 0.5))
```




# Research.

## Research question.

Is it possible to link smoking to Methylation rates by predicting smoking status from Methylation rates. using machine learning

# Machine Learning.

To make the data useful for machine learning it needs to be "cleaned".

## Data Preparation.

luckily for us the data was already very clean already all we had to to was removing the identifiers (GSM) and the age and gender since they're not Methylation rates. And we also did some cleaning earlier removing the NAs.

```
ml_data <- my_data[5:24]
ml_data <- cbind(ml_data, my_data[2])
write.csv(ml_data, file = "data/ml_data.csv", row.names = F)
```

## Algorithm Comparison.

After running the cleaned data trough weka using standard ML algorithms these where the results.

```
ml_invalid <- read.csv("data/ml_results_not_valid.csv")
ml_valid <- read.csv("data/ml_results_valid.csv")
kable(ml_invalid, caption = "UnCross-validated ML methods", align = "l")
```

Table 4: UnCross-validated ML methods

| Method | Acc.Percent | Speed.Sec | TP | FP | TN | FN |
|---|---|---|---|---|---|---|
| zeroR | 68.9211 | 0.01 | 428 | 193 | 0 | 0 |
| oneR(-B=56) | 68.9211 | 0.00 | 428 | 193 | 0 | 0 |
| J48(-M=50) | 68.9211 | 0.00 | 428 | 193 | 0 | 0 |
| IBk(-K=38) | 69.0821 | 0.70 | 428 | 192 | 1 | 0 |
| SimpleLogistic | 68.9211 | 0.00 | 428 | 193 | 0 | 0 |
| SMO | 68.9211 | 0.00 | 428 | 193 | 0 | 0 |
| NaiveBayes | 46.3768 | 0.01 | 139 | 44 | 149 | 289 |
| RandomForest | 100.0000 | 0.03 | 428 | 0 | 193 | 0 |

```
kable(ml_valid, caption = "Cross-validated ML methods", align = "l")
```

Table 5: Cross-validated ML methods

| Method | Acc.Percent | TP.Avg | FP.Avg | TN.Avg | FN.Avg | Avg.Area.Under.ROC |
|---|---|---|---|---|---|---|
| zeroR | 68.92 | 428.0 | 193.0 | 0 | 0 | 0.50 |
| oneR(-B=56) | 68.92 | 428.0 | 193.0 | 0 | 0 | 0.50 |
| J48(-M=50) | 68.76 | 426.50 | 192.5* | 0.5 | 1.5 | 0.50 |
| IBk(-K=38) | 68.79 | 426.0 | 191.8* | 1.2 | 2 | 0.53 |
| SimpleLogistic | 68.78 | 426.4 | 192.3 | 0.7 | 1.6 | 0.51 |
| SMO | 68.92 | 428.0 | 193.0 | 0 | 0 | 0.50 |
| NaiveBayes | 46.22* | 138.3* | 44.3* | 148.7v | 289.7v | 0.55 |
| RandomForest | 67.02 | 405.0* | 181.8* | 11.2v | 23v | 0.53 |

**: significantly lower compared to ZeroR.*

*v: significantly higher compared to ZeroR.*

*significance determined by a (corrected) paired T Test $\alpha = 0.05$*

From these results we can gather that IBk is the best preforming algorithm with the highest although not significantly different from ZeroR accuracy, and the second highest area under the ROC-curve, IBk is also one of the fastest with 0.00 seconds taken.

However these accuracies are still low and close to ZeroR, for this reason it seems wise to remove classifiers that yield the lowest amount of information. This might actually raise the accuracy of certain algorithms because, paradoxically too much information can be detrimental to accuracy.

Weka has the select attributes feature and after using it we came to the conclusion that keeping: cg00050873, cg00212031, cg01707559 and cg02839557 will add the most information.

```
ml_data_2 <- ml_data[c(1,2,6,12,21)]
write.csv(ml_data_2, file = "data/ml_data_pruned.csv", row.names = F)
```

After cutting away the less useful data we came to these results.

```
ml_short <- read.csv("data/ml_results_shortend.csv")
kable(ml_short, caption = "Cross-validated ML methods on selected data", align = "l")
```

Table 6: Cross-validated ML methods on selected data

| Method | Acc.Percent | TP.Avg | FP.Avg | TN.Avg | FN.Avg | Avg.Area.Under.ROC |
|---|---|---|---|---|---|---|
| ZeroR | 68.92 | 428.0 | 193.0 | 0 | 0 | 0.50 |
| OneR(-B=42) | 68.92 | 428.0 | 193.0 | 0 | 0 | 0.50 |
| J48(-M=6) | 69.94 | 426.2 | 191.1 | 1.9 | 1.8 | 0.51 |
| IBk(-k=40) | 68.78 | 425.8 | 191.7 | 1.3 | 2.2 | 0.54v |
| SimpleLogistic | 68.89 | 427.8 | 193.0 | 0 | 0.2 | 0.51 |
| SMO | 68.92 | 428.0 | 193.0 | 0 | 0 | 0.50 |
| NaiveBayes | 68.20 | 422.2 | 191.7 | 1.3 | 5.8 | 0.55v |
| RandomForest | 65.36* | 369.2* | 156.3* | 36.7v | 58.8v | 0.54v |

**: significantly lower compared to ZeroR.*

*v: significantly higher compared to ZeroR.*

*significance determined by a (corrected) paired T Test* $\alpha = 0.05$

looking at the results from the above table using J48 at -M 6 would yield the best results.