Using machine learning to identify thyroid diseases logbook

Reindert Visser

19-11-2021

Exploratory Data Analysis

Research topic

For this project thyroid disease is the main area of research. Problems with the thyroid will result in mental health problems and fatigue. This can be a severe problem and hard to track down. Luckily machine learning can help figure out and calculate patterns. Patterns that otherwise wouldn't be found otherwise.

The data was originally gathered from Garvan Institute. Consisting of 9172 records from 1984 to early 1987. There are 30 columns and a lot of instances. All of the thyroid related hormones have been measured, but also external factors are accounted for. Such as medication, a swollen thyroid or a tumour. These factors affect the hormone cycle of the thyroid, changing a persons mental health. The data has a lot of instances, all of these columns contain a true or false. The patient uses medication or not for example. There are only a few numeric columns, these are for the hormone expression values. If the specific hormones (T3 and T4) are measured, the expression value is noted under the _measured column.

Which will bring the research question:

Is it possible to predict if a person has thyroid disease by using machine learning, looking at expression values, medicine usage and external factors?

With machine learning, great amounts of data can be loaded up and a model can be constructed. When the algorithm encounters new data, these patterns can be identified and will help improve the model. Since the data has a lot of instances, machine learning will be a great tool to use. Since the data has a Sick column, it's either True or False. So this is supervised learning with classification on Sick. Because this is what a patient will look for. More data will help improve the model. But since there are nine thousand instances, an additional instance won't impact the model.

Readying the data

Before the research can start, the data needs to be readied. The original data set needs to be loaded up and readied for usage.

Loading the data

When initially downloaded, the data had an additional column. This was the patient's case, before starting with machine learning; this needs to be removed as soon as possible. The column `Diagnose Letter` was combined with an id of the patient. This was removed using the regular expression `\[[0-9]+\]`, in order to create a true machine learning algorithm. So the algorithm won't score 100% correct while training.

The labels from the variable `my_labels` were used from the disclosed information in `data/thyroid0387.names`. Only the labels were used, not the additional text and explanation. These are used for readability and make the upcoming graphs better.

```
my_data <- read.table("data/thyroid0387.data", sep = ",", header=F, na="?")
my_labels <- read.table("data/labels.txt", sep="\n", header=F)
```

```
colnames(my_data) <- as.vector(my_labels[[1]])

dplyr::as_tibble(head(my_data, n=5))
```

```
## # A tibble: 5 x 30
##     age sex   on_thyroxine query_on_thyroxi~ on_antithyroid_medi~ sick  pregnant
##   <int> <chr> <chr>        <chr>             <chr>                <chr> <chr>
## 1    29 F     f            f                 f                    f     f
## 2    29 F     f            f                 f                    f     f
## 3    41 F     f            f                 f                    f     f
## 4    36 F     f            f                 f                    f     f
## 5    32 F     f            f                 f                    f     f
## # ... with 23 more variables: thyroid_surgery <chr>, I131_treatment <chr>,
## #   query_hypothyroid <chr>, query_hyperthyroid <chr>, lithium <chr>,
## #   goitre <chr>, tumor <chr>, hypopituitary <chr>, psych <chr>,
## #   TSH_measured <chr>, TSH <dbl>, T3_measured <chr>, T3 <dbl>,
## #   TT4_measured <chr>, TT4 <dbl>, T4U_measured <chr>, T4U <dbl>,
## #   FTI_measured <chr>, FTI <dbl>, TBG_measured <chr>, TBG <dbl>,
## #   referral_source <chr>, Diagnose_letter <chr>
```

The data will be loaded using the `read.table` R build-in function. The data does not have a header and uses a `,` as separator. NA (Not available) is defined with a `?` in the data. A lot of external factors such as pregnancy, medication, tumours or a psychiatrist have been taken into account. For most of the instances, this is not the case, so it is labelled with `f`. For the hormone measurement columns, these columns form a combination of name and expression value. The hormone name is simply a true or false, while the measured value is the exact amount of hormone expression measured. If there is no hormone expression measured, it will default to NA. This will not form a big issue since the hormone column gives enough information to work with.

For the label file, the data does not have a header either. The separator is a `\n` and there are no NA's. This data is used to overwrite the current column names with the more easy to understand and practical names.

NA values can be a problem for some algorithms, it is important to keep this in mind when preparing the data. Using `sapply` the NA values can be summed up.

```
knitr::kable(sapply(my_data, function(x) sum(is.na(x))), caption = "Amounts of NA found in the data." )
```

Table 1: Amounts of NA found in the data.

	x
age	0
sex	307
on_thyroxine	0
query_on_thyroxine	0
on_antithyroid_medication	0
sick	0
pregnant	0
thyroid_surgery	0
I131_treatment	0
query_hypothyroid	0
query_hyperthyroid	0
lithium	0
goitre	0
tumor	0
hypopituitary	0

	x
psych	0
TSH_measured	0
TSH	842
T3_measured	0
T3	2604
TT4_measured	0
TT4	442
T4U_measured	0
T4U	809
FTI_measured	0
FTI	802
TBG_measured	0
TBG	8823
referral_source	0
Diagnose_letter	0

Only the sex column has a lot of NAs, this could be because people wouldn't like to include their gender. There are also a lot of NAs in the hormone columns.

Exploring the data

For the majority of the data, the data is categorical binary data. Which indicates it's either a yes or a no, 1 or 0, True or False. In the dataset these are labelled with either `t` or `f`. But not all, there are also a few numeric values. These numeric values are intervals, and can be used for counting but also used as "in between" values such as decimal numbers.

The thyroid plays a big part in the endocrine system, it gives off multiple hormones to regulate the body. There are multiple hormones that the thyroid can give off. The values are tests in order to measure the amount of hormone in the blood. Since most of the hormones transport with blood cells, the thyroid values are testable. If there is a value measured, the default false will become true. With the matching value in the hormone_"Measured" column. [1]

The complete column description is made under `codebook.xlsx`, which describes the column, name, data type, unit and description.

```
Openend_codebook_results <- readxl::read_xlsx("resources/codebook.xlsx")
knitr::kable(Openend_codebook_results, caption = "Overall overview of the dataset")
```

Table 2: Overall overview of the dataset

Name	Data Type	Unit	Description
age	integer	years	The age in years of the person.
sex	char	NA	The sex of the person.
on thyroxine	bool	NA	If the person uses thyroxine medication at this point.
query on thyroxine	bool	NA	If the person is on a waiting list for thyroxine medication or in the medical process of getting thyroid medication.

Name	Data Type	Unit	Description
on antithyroid medication	bool	NA	If the person uses antithyroid medication, lowering the effect of the thyroid in general.
sick	bool	NA	If the person is sick, having a thyroid issue which is medically proven.
pregnant	bool	NA	If the person is carrying a child, since this can effect your mental health. Because the body is working extra hard.
thyroid surgery	bool	NA	If the person has had surgery related to the thyroid, in the past or future.
I131 treatment	bool	NA	A ratiation treatment where the radioactive isotope iodine (I-131) is used to effect the thyroid.
query hypothyroid	bool	NA	If the person has hypothyroid where the thyroid will producte too little hormones, causting an overreaction
query hyperthyroid	bool	NA	If the person has hyperthyroid where the thyroid produces too much hormones and causing an overreaction.
lithium	bool	NA	Lithium is an element that circulates in the body and helps different processes. Patients can take extra to stimulate the thyroid and becoming more active. Which will result in true in the data.
goitre	bool	NA	A goitre is a major increase in size of the thyroid or local swelling. Which can be diagnosed, if this is the case then the value is true in the data.
tumor	bool	NA	If the person has a tumor or not, since that will drastically effect your body and mental health. Leading to the same symptons with a different cause.
hypopituitary	bool	NA	Hypopituitary is a rare case of the pituitary gland does not produce enough hormones.
psych	bool	NA	If the person is seeing a psychologist, since thyroid issues will have an effect on mental health.
TSH measured	bool	NA	Thyroid stimulating hormone measured: will measure the amount of thyroid stimulating hormones in the blood.
TSH	float	continuous	Thyroid stimulating hormone: this hormone helps the thyroid release the T3 and T4 hormones. Without TSH, the thyroid would not be able to function at all. If there is too much TSH, the thyroid isn't releasing enough T3 and T4. TSH is released from the hypothalamus and targets the thyroid.
T3 measured	bool	NA	Triiodothyronine measured: will measure the amount of T3 in the blood. Since all transport of the thyroid hormones is done via blood cells, this will give an accurate amount of hormones.
T3	float	continuous	Triiodothyronine: is one of the two thyroid hormones, the hormones will influence each other and affect a lot of different processes in the body. Such as metabolism of glucose, the breakdown of cholesterol or increasing the heart rate.
TT4 measured	bool	NA	Thyroxine measured: will measure the total amount of free T4 and bound to blood T4. Since there are two types of T4, the way that it's used can differ. It can be bound to the blood or roam in the fluid. There for it is also provable by the test.
TT4	float	continuous	Thyroxine: this is the other hormone that will cooperate with the T3 hormone. They have the same function and are involved in the same general processes of the body: Metabolism, growth and increased catecholamine.

Name	Data Type	Unit	Description
T4U measured	bool	NA	Thyroxine unbound measured: will measure the amount of T4 Uptake, which tests the usage of available T4.
T4U	float	continuous	Thyroxine unbound: T4 wich is free T4 hormone in the cell. This is transported via the blood so it can be measurable.
FTI measured	bool	NA	Free T4 Index measured: will measure the amount of Free T4, Total Free (T4) Index. This can range between 0.7-1.9 nanogram per decilitre of blood.
FTI	float	continuous	Free T4 Index: amount of free T4 which roams free in the body, being unbound and roaming free. Helping other processes in the body.
TBG measured	bool	NA	Thyroxine binding globulin measured: will measure the amount of TBG enzymes. Which means this is an indirect test to measure T4.
TBG	float	continuous	Thyroxine binding globulin: this is the amount of helper enzymes which is cleaved to make T4 hormones.
referral source	String	NA	External factors which are sets of strings in the data, indicating diagnosed conditions. A diagnosis "-" indicates no condition requiring comment.
diagnosis letter	char	NA	Additional diagnose with the letter with the patient's condition.

Hormone information and testing is found on: [2]

All these hormone expression values can be found with tests, in order to prove activity, compare the expression values or prove absence. The test results will be displayed under the measured column in the data. [3]

The remaining columns are about external factors. Where the body has either abnormalities or lack of hormones/body resources. But since thyroid effects a person's mental health, a psychiatrist is also included. But also pregnancy or a tumour, since these factors can change your life drastically. [4]

Now that the data is fully understood, know what all columns do and why it's necessary to include them, the visualization process can begin.

Visualizing the data

Plotting the data is the fastest way to give a visual view of the data. This can be done by all sorts of ways, since the dataset is mostly categorical; it will be hard to make graphs. Since it is mostly either a `t` or `f` in the dataset. Most of the visualization will be done through tables, due to the lack of numeric values.

```
ggplot(data = my_data, mapping = aes(x = sick)) +
  geom_bar(aes(fill = sick)) +
         labs(caption = "Data source: sick column") +
         scale_x_discrete(labels=c("Healthy instances", "Sick instances")) +
         scale_fill_discrete(labels=c("Healthy instances", "Sick instances")) +
         xlab("Classification label") +
         ylab("Frequency in dataset")
```

The great majority of the instances in the data are not patients. Only around 3% of the instances are patients. Which might become tricky to correctly diagnose the real patients according to their thyroid values. Since an algorithm will easily achieve a high accuracy by simple saying that all instances are healthy.

Another factor in the dataset is the sex of the instance. Since thyroid involves many mental problems, fatigue and disorders, there will be a difference between sex. The pregnancy column is also not relevant for all men in the dataset.

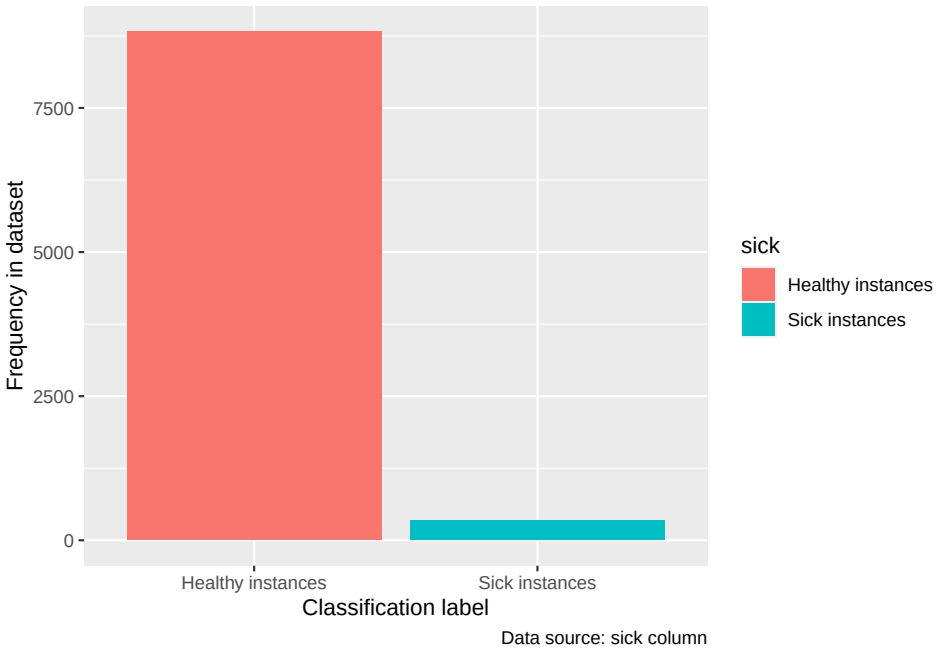


Figure 1: The distribution of sick instances in the thyroid dataset.

```
ggplot(data = my_data, mapping = aes(x = sex)) +
  geom_bar(aes(fill = sex)) +
         xlab("Sex of the instance") +
         ylab("Frequency in dataset") +
         labs(caption = "Data source: patients' sex in the dataset") +
         scale_x_discrete(labels=c("Female", "Male", "Not included")) +
         scale_fill_discrete(labels=c("Female", "Male", "Not included"))
```

As predicted there are more female instances in the data. There are also people that did not want to specify their sex, these instances are labelled under the 400 NA in the data. Combining this with the lack of male instances, this hurts the credibility. Where ideally it would be an even distribution.

There are many types of diagnosis, in varying degree of frequency. This will give insight in the distribution of the dataset.

```
knitr::kable(table(my_data$Diagnose_letter))
```

Var1	Freq
-	6771
A	147
AK	46
B	21
C	6
C|I	12
D	8
D|R	1
E	1
F	233
FK	6
G	359

Var1	Freq
GI	10
GK	49
GKJ	1
H|K	8
I	346
J	30
K	436
KJ	11
L	115
LJ	1
M	111
MI	2
MK	16
N	110
O	14
OI	1
P	5
Q	14
R	196
S	85

As shown above, not everyone in the dataset is a patient. Some diagnosis are more common than others, a few cases are really rare on a scale of nine thousand instances.

Since there are differences between sexes, not only body but also in hormones. The thyroid might act different, this might affect the effects people feel and how they respond to this.

```
ggplot(data = my_data,
  mapping= aes(x = sex, y= Diagnose_letter)) +
  geom_point(col= "blue", alpha = 0.2) +
  geom_smooth(method="loess", se=FALSE) +
  xlab("Sex of the instance") +
  ylab("Diagnose letters") +
  labs(caption = "Data source: sex and diagnose columns") +
  scale_x_discrete(labels=c("Female", "Male", "Not included")) +
  scale_fill_discrete(labels=c("Female", "Male", "Not included")) +
  theme_minimal()
```

```
## `geom_smooth()` using formula 'y ~ x'
```

Not all males have the same problems as females, some more frequent then others. NA is added this time to give a general overview of the data.

Another common medicine that helps thyroid patients with their problems is thyroxine. This is additional TT4 hormones that will help the body maintain or keep the cycle it is currently in. But some people use it to cope with problems, these people might falsely think that they are a thyroid patient.

```
ggplot(data = my_data,
  mapping= aes(x = on_thyroxine, y= TT4_measured)) +
  geom_jitter(width = 0.4, height = 0.4) +
  geom_point(col= "blue", alpha = 0.2) +
  geom_smooth(method="loess", se=FALSE) +
  xlab("Patient using thyroxine medication") +
  ylab("Measured thyroxine in testing") +
```

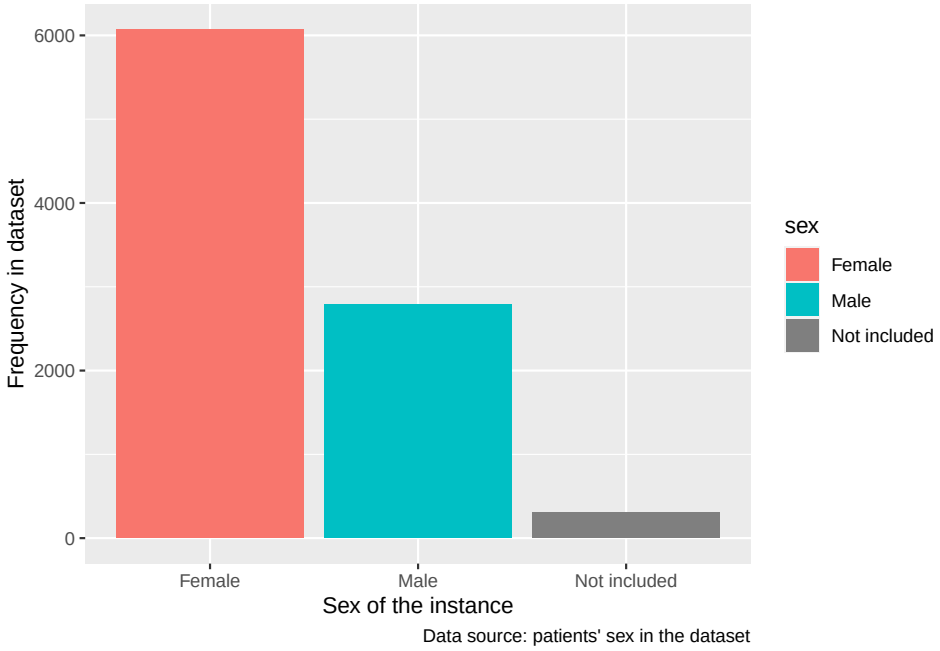


Figure 2: Distribution of sexes in the dataset.

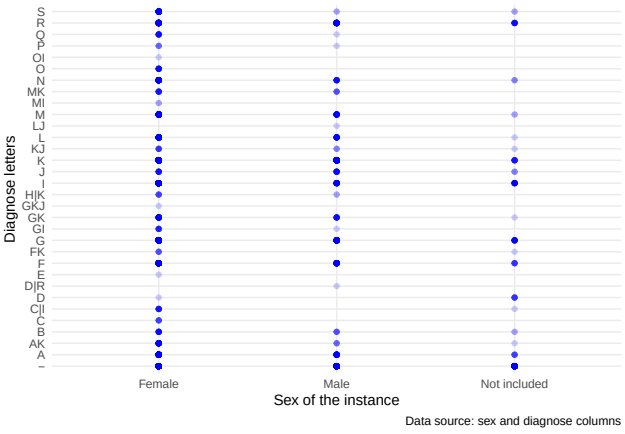


Figure 3: Different types of diagnosis and relation to sex.

```
  labs(caption = "Data source: thyroxine usage") +
  scale_x_discrete(labels=c("False", "True")) +
  scale_y_discrete(labels=c("False", "True")) +
  theme_minimal()
```

```
## `geom_smooth()` using formula 'y ~ x'
```

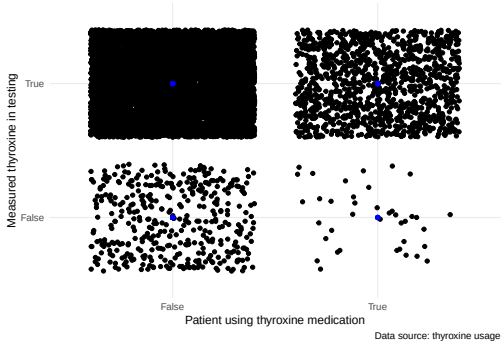


Figure 4: Distribution of thyroxine users in the dataset.

As seen, the patients that use thyroxine actually translates really well to the data. The True-True combination is proven TT4 and extra thyroid medication. Due to the scope of the data and the big amounts of instances, it might look insignificant. But there are a lot of instances where only a few are actually sick, so it is important to identify them. Thyroxine usage might be useful for the future algorithm.

Overall the data looks really good, but if you look a bit closer, there is something strange going on in the data. The age column has three outliers. Two persons are in their nine thousands' and one person is over four hundred years old. There must have gone something wrong with these values, since it's biologically impossible to be this age. These values need to be either removed or edited.

The instances with unnatural ages are now edited to only use the first two digits. If someone would say forty and a half, and entered it like "40.5". The values combined will be 405 which enters the range of one of the instance's age. All three instances are not sick so the effect will be minimal.

Now that the unnatural ages have been modified, the data is now clean and seems accurate with the normal age of people. The best way to visualize this, is via a boxplot. To directly show the mean and the quartiles within the age distribution.

```
ggplot(my_data, aes(y = age, x=as.factor("Patients"))) +
  geom_boxplot() +
  xlab("") +
  ylab("Age spread in years") +
  labs(caption = "Data source: patients' ages from the data") +
  theme_minimal()
```

There are no extreme outliers any more and this is believable. Since it is biological impossible to have such extreme ages. Most of the instances have an age of round 54 years old. While the third quantile leans a bit more towards the older age, since it does not align with the 75 year line.

To finalize the analysis, there are the hormone columns. There are six of these columns, with numeric values and they visualize the expression value. With a boxplot all the additional information about the data will be showed.

```
ggplot(my_data, aes(y = TT4, x= as.factor("T4") )) +
  geom_boxplot() +
  xlab("") +
```

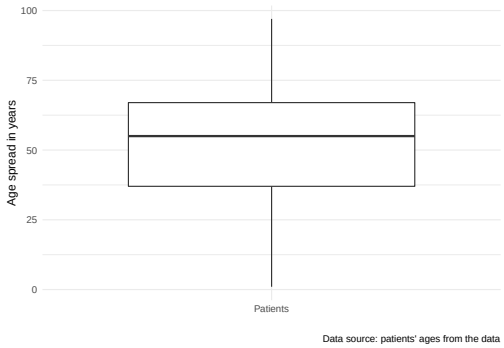


Figure 5: A distribution of age within the data

```
  ylab("Gene expression values") +
  labs(caption = "Data source: patients' expression values from the data")
```

```
## Warning: Removed 442 rows containing non-finite values (stat_boxplot).
```

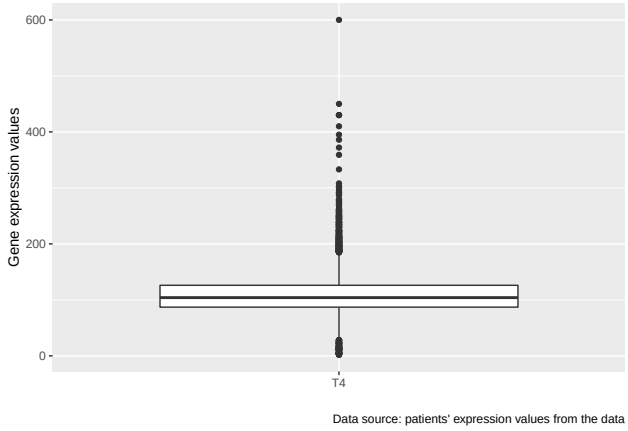


Figure 6: A boxplot of the TT4 expression values.

There are some outliers in this column, but because this is on the big scale of thousands of instances. This effect is not significant. For this reason, the outliers won't be normalized. Because the gained effect won't change the outcome.

Machine learning

This is the most important part of the whole process. To see if an algorithm can identify thyroid patients from their expression values.

Type 1 and type 2 errors

There are two types of errors, the type-1 error and type-2 errors. The type-1 error is simply a false alarm. Where the algorithm incorrectly said `Yes`, when it should've been a `No`. The type-2 error is a underestimation or a simply a miss, where the algorithm incorrectly said `No` where it should have `Yes`. In this case for type-1, it is a patients that is diagnosed sick that is actually healthy. For the type-2 error, it is a patient that is incorrectly declared healthy, while actually being sick.

Since the dataset has a lot of patients, where the sick patients are more important. Not only because of

the minority but also the sick need to get diagnosed properly. Because around the 90% is healthy, the false positives need to be as low as possible. The patients need to be correctly identified, but the algorithm doesn't need to go too harsh on the healthy instances.

Algorithm

Exploring machine learning

The overall result of the machine learning process is not valid. Due to the skewness of the data, ZeroR will get 97% correct by simply saying False. For OneR will make illogical models, for example "if age is not NA, sick is false." Due to the lack of actual patients in the dataset, the algorithm will won't do a good job on actual patients. The J48 tree will make a simple model with everything resulting in False. NaiveBayes will try, but perform significantly less than zeroR. The only way to get an acceptable score with an actual algorithm is by overfitting on the trainings data. Unfortunately, this won't work for new instances. So the data needs to be trimmed down. Since the ratio is around 9000 healthy and 200 sick, the amount of healthy instances needs to be brought down. While the patient amount should get more weight.

The extra weight for the sick class, will be done by making a dedicated class column. So there is a sick column, and an additional class column. The only differences are that the NAs will be taken out, so there is a somewhat clean column. There is still a large number of healthy instances, to bring this down way more; around 60% of the healthy instances will be removed. Now the data has 3772 instances, where around 200 instances are sick. With the extra weight of another class column, the results are now a lot better.

Running experiments

With the newly made dataset, all algorithms are tested with 10-fold cross validation (not training data). ZeroR keeps performing good, ZeroR will now do 93% correct by simply saying healthy to everyone. But now OneR will do a better job and make an actual model, and will have 96% correct and will base the model on T3 expression. But the J48 tree will do 98.6% correct, with 10-fold cross validation and a minimum number of objects of 10. Picking a number higher will lower the correctly diagnosed sick instances. Picking a lower number will cause overfitting and resulting in more incorrectly diagnosed sick instances. So 10 seems to be a perfect option.

Increasing or decreasing the minimal number of objects drastically, gives about the same results but different decimals. But there is some mild variation in correctly diagnosed sick instances. Which proves that this model is surprisingly functional. Due to this not being trainings data, this tree should be valid in a more realistic scenario. From the 231 patients, 28 are diagnosed incorrectly healthy. With the massive skewness of this data, only having 231 sick instances, this is a good result. The cleaned and trimmed data can be found under `data/thyroid_final.arff`.

Running the J48 algorithm with the optimal options in the Weka experimenter, with 100 runs and 10-fold cross validation, the score will be in the range of 98% and 99% instances correct. This algorithm performs the best, since it diagnoses the most sick instances correct without a lot of False Positives. The False Positives are a lot worse, since the instance is wrongfully judged healthy. A False Negative also has effects, but after a hypothetical medical investigation; the patient will be proven healthy. While a real patient's condition can worsen. The tree also seems logical, since it uses the most obvious columns. An actual doctor would also check hormone expression, then medicine usage and finally the external factors. So the false positives (incorrectly classified healthy) will be weighted more, in order to force the algorithm to classify them correctly.

Algorithm performance

Multiple test results have been run, the most important factors are displayed under the columns.

```
Openend_ML_results <- readxl::read_xlsx("resources/ML_results.xlsx")
knitr::kable(Openend_ML_results, caption = "Results of various machine learning algorithms")
```

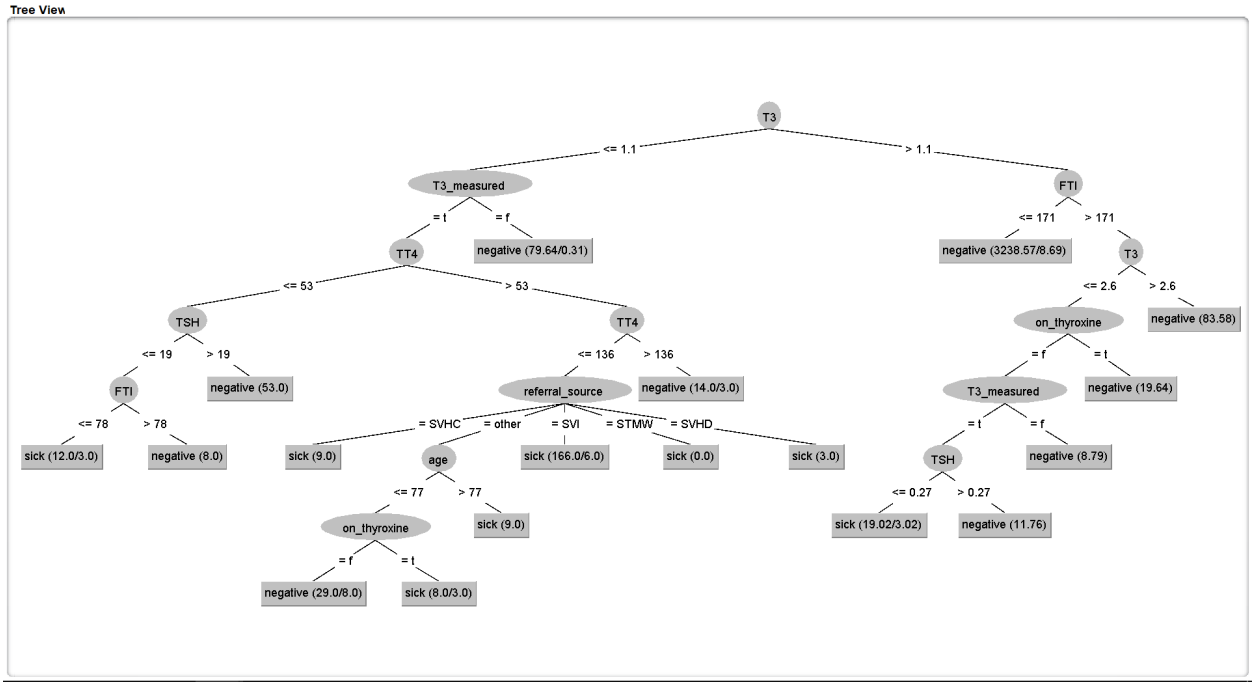


Figure 7: The J48 tree visualized by the Weka 3.8.5 Tree visualize function.

Table 4: Results of various machine learning algorithms

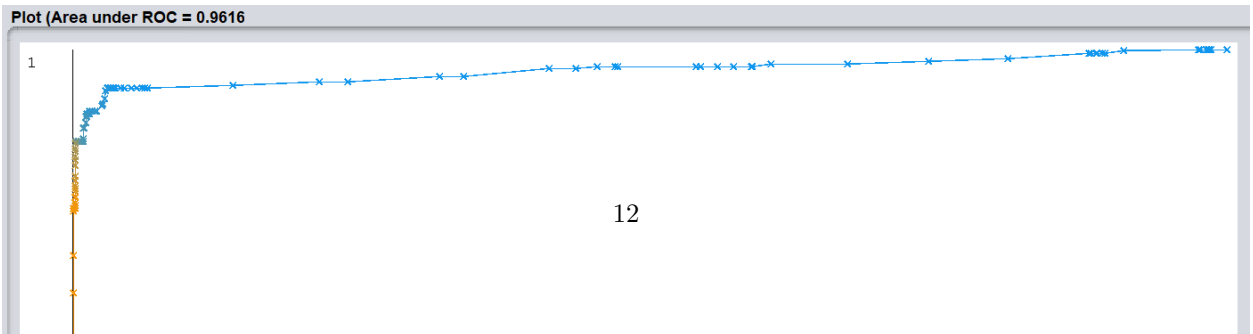
algorithm_name	speed	accuracy	error_rate	true_positives	false_positives	false_negatives	true_negatives
Zero R	0.01	0.94	0.062	3541	231	0	0
One R	0.01	0.96	0.040	3466	53	75	178
Naïve Bayes	0.02	0.93	0.070	3314	52	227	179
J48-Tree	0.14	0.99	0.010	3524	28	17	203
Ibk	0.00	0.96	0.040	3484	87	57	144
SMO	0.24	0.94	0.060	3540	231	1	0
Random Forest	0.42	0.98	0.020	3533	49	8	182

Using the function ‘read_xlsx“ there is a simple way to open an Excel file and displaying the file as a tibble. The J48-Tree proves to be best, since it has the lowest amount of false negatives and false positives. On top of that, it also runs the best with true negatives.

ROC-Curve

Visualizing the algorithm’s performance results can be done via the ROC-Curve. This is a graph of sensitivity as function of specificity, showing how well the algorithm can judge. With 1 being absolutely perfect, judging 100% correctly. But this is not realistic for the algorithm on this specific dataset.

Running the J48-algorithm with 10-fold cross validation, using the optimal 10 minimal number of objects. With the Weka 3.8.5 Experimenter function, rerunning this test 100 times. Generating a total of 1000 trees,



Meta-learners

Combining multiple algorithms and bundling the combined results, can give a very diverse view on the data. With voting, multiple results are joined together and picking the most frequent label. While boosting uses multiple algorithms in an completely different way. With an algorithm being trained on the others mistakes, weighting the wrong instances more. In order to not make the same mistakes again.

Bagging with bootstrapped data, using REPTree will generate multiple trees with each slightly different data. The bagging algorithm performs well, because it uses the tree structuring, just like J48. Bagging will get an accuracy of 98.4% and an error rate of 1.6%. Using the bagging algorithm with REPTree and it's default settings.

Stacking is using the same algorithm with slightly tweaked seeds, the same algorithm will be run multiple times. Using stacking with OneR with default settings, will give an accuracy 93.8% and an error rate of 6.2%. This sounds high, but it is the same output of ZeroR and diagnoses everyone incorrectly healthy. Stacking J48 will give the same output with judging everyone incorrectly healthy.

Boosting will use multiple of the same algorithms designed to fix each other's mistakes. If an algorithm performs 50% correct and 50% wrong, another algorithm will be specialized to fix the 50% wrong instances. Running this on the thyroid data, an accuracy of 95.8% with OneR. OneR algorithm is trained to fix the other OneR's mistakes. So this is a good way to get valid results with no luck involved.

References

[1] Boron, W. F., Boulpaep, E. L. (2012). *Medical Physiology Chapter 49, "Synthesis of Thyroid Hormones"* (2nd ed.). Elsevier/Saunders. ISBN 9781437717532.

[2] Basile, L. M. (z.d.). *What are T3, T4, and TSH?* EndocrineWeb. used on 23 September 2021, from https://www.endocrineweb.com/thyroid-what-are-t3-t4-tsh

[3] Shahid, A. H., Singh, M. P., Raj, R. K., Suman, R., Jawaid, D., Alam, M. (2019). *A Study on Label TSH, T3, T4U, TT4, FTI in Hyperthyroidism and Hypothyroidism using Machine Learning Techniques* 2019 International Conference on Communication and Electronics Systems (ICCES). Published. https://doi.org/10.1109/icces45898.2019.9002284

[4] James R Mulinda, Arthur B Chausmer, Francisco Talavera *Hypopituitary Hypopituitarism Causes, Symptoms and Treatment* 2018, January 3rd. EMedicineHealth. *https* : *//www.emedicinehealth.com/hypopituitary/article*$_e$*m.htm*

[5] Swets, J. A. (1996): Signal detection theory and ROC analysis in psychology and diagnostics: Collected papers. Scientific psychology series. Hillsdale, NJ, US: Lawrence Erlbaum Associates, Inc.