Introduction into machine learning and analysis of Breast Cancer Proteomes

Theme09 - Introduction to Machine Learning

Mats Slik
344216
BFV3
November 13, 2022
Dave Langers (LADR)
Bart Barnard (BABA)

Introduction into machine learning and analysis of Breast Cancer Proteomes

Theme09 - Introduction to Machine Learning

Mats Slik
344216
Bioinformatics
Institute for Life Science & Technology
Hanze University of Applied Sciences
Dave Langers (LADR)
Bart Barnard (BABA)
November 13, 2022

Abstract

the data were used to assess how the mutations in the DNA are affecting the protein expression landscape in breast cancer. Genes in our DNA are first transcribed into RNA molecules which then are translated into proteins. Changing the information content of DNA has impact on the behavior of the proteome, which is the main functional unit of cells, taking care of cell division, DNA repair, enzymatic reactions and signaling etc. my question is: Are there different ways to categorize breast cancer based on protein expression data, with machine learning being used to classify them without using the pam50 proteins?

Table of Contents

List of Abbreviations

EDA	Exploratory Data Analysis
TCGA	The cancer Genome Atlas Program
CPTAC	Clinical Proteomic Tumor Analysis Consortium
DNA	Deoxyribonucleic Acid
RNA	Ribonucleic Acid

List of Figures

List of Tables

1 Introduction

In today's world although breast cancer has reach a high rate of awareness and research its still a significant problem [2]. With 1 in 8 women getting a form of brest cancer. so it is still important to get a quick diagnosis of the stage of the cancer. the stages used in this report ar T1 to T4. T1 in this stage the cancer has not grown deeply in surrounding tissue yet, and hasn't reached a lymph node or other parts of the body. T2 and T3 in these two stages the cancer has grown more deeply in the surrounding tissue and may have reached a lymph node in the body but not other parts of the body. T4 means that in this stage the cancer has reached and gropwn into other organs or parts of the body. There are more type of cancer classification but these arent applicable or used in this report and analayses for simplicity. for classification of these stages we are looking at protein expression data. the data we used assess how the mutations in the DNA are affecting the protein expression landscape in breast cancer. Genes in our DNA are first transcribed into RNA molecules which then are translated into proteins. Changing the information content of DNA has impact on the behavior of the proteome, which is the main functional unit of cells, taking care of cell division, DNA repair, enzymatic reactions and signaling etc. normally This data containing is filtered for proteins originating from a list of 50 genes called PAM50. and these are selected to further classify the type/subtypes of brest cancer on. PAM50 is a list of 50 genes that classify breast cancers into one of five intrinsic subtypes from formalin-fixed, paraffin-embedded tissues by real time polymerase chain reaction (RT-PCR) (62). These 50 genes identified were refined from a list of 1,906 genes, which were found in four previous microarray studies. my question is: Are there different ways to categorize breast cancer based on protein expression data, with machine learning being used to classify them without using the pam50 proteins?

2 Material & Methods

All created and used files are stored in this Github repository: here.

Data The data set contains published iTRAQ proteome profiling of 77 breast cancer samples generated by the Clinical Proteomic Tumor Analysis Consortium (NCI/NIH). It contains expression values for ~12.000 proteins for each sample, with missing values present when a given protein could not be quantified in a given sample. This data was sampled from 105 originally from the TCGA (The Cancer Genome Atlas Program - NCI), which was further filterd to 77 samples containing high quality protein expression data. Normally a filter is applied over protein expression dat like this to make more sense out of it like the PAM50. Since my research question is about finding new proteins and or ways to classify them without this PAM50 , it won't be used.

First the columns with Gene_symbols and gene_name where stripped from it so the data set now only contained numerical data. After that step the data from the clinical data is merged with the samples from the protein expressions, but to first do this the names of the samples must correspond with the names/ID used in the clinical data set. For example in the protein dat we have : `A0.A12D.01TCGA` and in the clinical dat we had `TCGA-A0-A12D`. For this a regex and split method was used to search on `"TCGA"` and split the names/ID's on `_ and - and .` and at last it was rearranged to the clinical data format for its name/ID. Thirdly the dataset containing the log2 protein expression data is transposed and its RefSeq accession number is used as the column names, so that each row is a new instance(samples) and the column are te attributes(proteins). After those first cleaning and formatting steps proteins with more than 10% of there values missing where removed from the data set.

this yielded the final data set that was used for the machine learning. this was exported to a .arff file created with the Rweka library in the logbook `Thema_09_log_Mats_Slik.rmd` found in the repository of this research.

machine learning For the machine learning`Weka 3.8.6` was used. Weka is a open source collection of machine learning algorithms for data mining tasks. It contains tools for data preparation, classification, regression, clustering, association, rules, mining, and visualization. it is build on the java platform The data set was first loaded in the explorer function of Weka and the column containing the different tumor stages was selected as a class attribute, so it could be trained on this column.

then the data set was used in the experimenter function of weka for a rough image on how different algorithms performed on the data. after this step further algorithms where tested and used in the explorer function since that gave better feedback on the performance of the algorithms used and the models it created. To see how these performed please look at the Results section namely tables 2 to 13. After comparing all the results and models created by trying different setting and parameters and algorithms one was selected to be exported. The model was selected in the Weka explorer and was exported to the `Models` folder in the repository. This model was used in the building of a Java jar file to classify new instances.

java wrapper For creating a java application to classify new instances `gradle 7.4` with the Java `JDK 17`. Further dependencies used where `Weka version 3.8.6` and `shadowJar version '7.1.2'`. these where used to create a single Jar with all dependencies and the trained model in it.

3 Results

When looking at the dimensions of the data set we can see there are a lot of proteins see table 1

Row.names	Tumor	NP_958782	NP_958785	NP_958786	NP_000436
blcdb9.I.CPTAC	Healthy	-0.1913	-0.1839	-0.186	-0.186
c4155b.C.CPTAC	Healthy	0.567	0.5787	0.5767	0.5767
TCGA-A2-A0CM	T2	0.6834	0.6944	0.6981	0.6871
TCGA-A2-A0D2	T2	0.1075	0.1042	0.1075	0.09751
TCGA-A2-A0EQ	T2	-0.9127	-0.928	-0.928	-0.9318
TCGA-A2-A0EV	T1	0.453	0.4726	0.4726	0.4586

```
## number of rows: 80 number of columns: 9201
```

With this distribution we can clearly see that there are a lot of attributes per instance, normally u want around ten percent of N instances as attributes. in this data set that corresponds with around 7 attributes. This limit our options on the machine learning part. /newline

After this first assessment of the data we started looking at the number of missing values as seen in the figures' fig 1 and 2 below, this is done to ascertain if there are a lot of missing values in the data set en if we neet to filter them out for better result later on in with the machine learning part

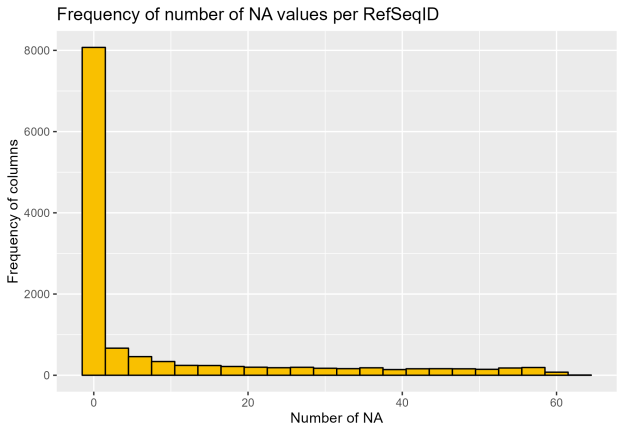

Figure 1: NA count histogram

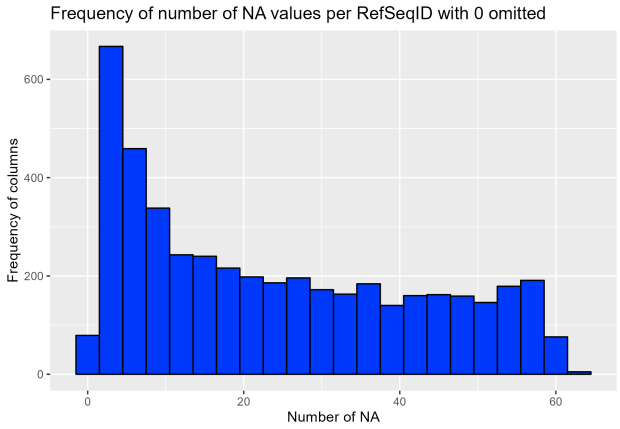

Figure 2: NA count histogram

As we can see in these figures 1 and 2 the distribution is very much to the left where a lot of proteins have one or only two missing values, further more there are still a couple of proteins that have a high number of missing values these are to be filtered out because this can create a false set of results when we are using them in our machine learning algorithm for classifying them on their cancer stage. so to further visualize the data we took the distribution of a couple of proteins in a multi boxplot as seen in figure 3.

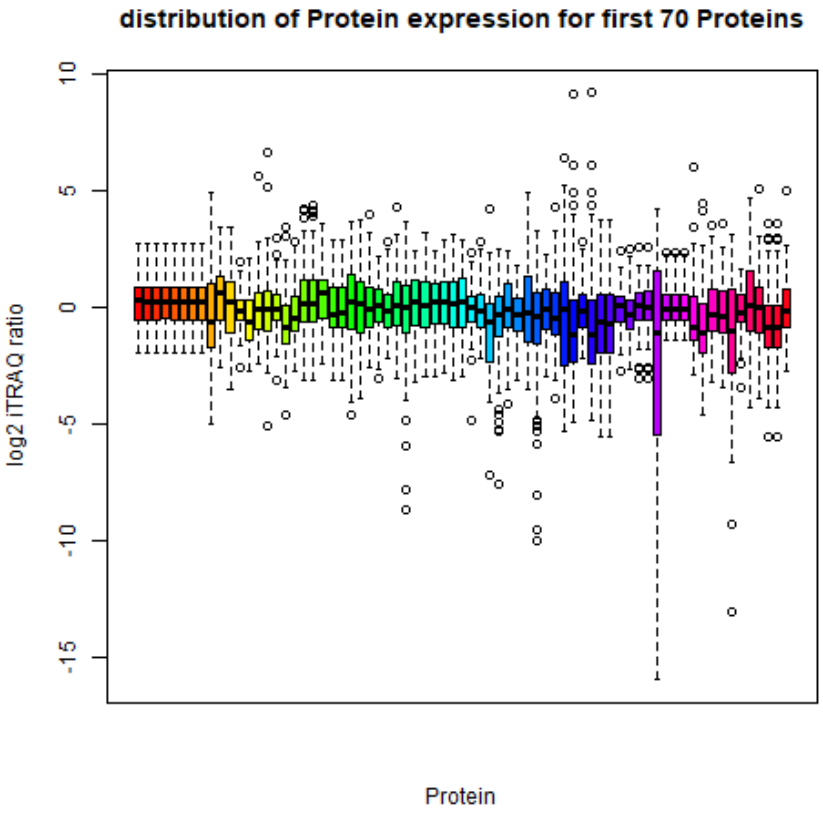

Figure 3: attribute distribution

In this figure 3 we can clearly see that for the first 70 protein that most have a distribution of their log2 itraq expression between 5 and -5 but there are some that have higher numbers. To further make sense of all the 12 to 9 thousand proteins in the data we calculated the standard deviation of them see figure 4

Figure 4: attribute standard deviation, comparison between NA filtered and non filterd data set

In this figure 4 we compared the normal data set and the one filtered that has had protein with more tha 10% of their values missing removed. In it, we can clearly see that a lot of proteins with high deviation are removed from the data. To make a further analyse of these samples

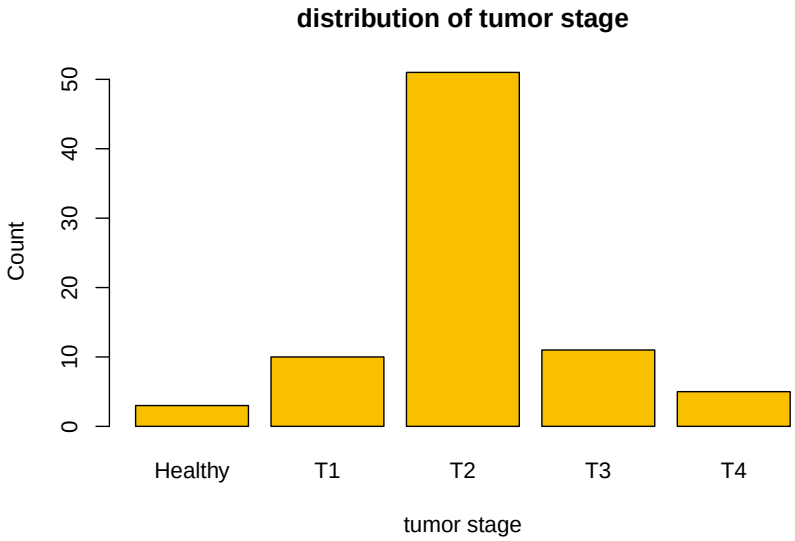

Figure 5: distribution of amount of samples per tumor stage

In this figure 5 we can see how the different samples are spread according to there cancer stages. here we can see a clear bias towards T2.

Table 2: Tabel with the algorithms with default settings used in initial comparison .

(1) rules.OneR '-B 6' -3459427003147861500
(2) trees.RandomTree '-K 0 -M 1.0 -V 0.001 -S 38' -9051119597407395800
(3) trees.RandomForest '-P 100 -I 100 -num-slots 1 -K 0 -M 1.0 -V 0.001 -S 38' 1116839470751428740
(4) trees.J48 '-C 0.25 -M 2' -217733168393644448
(5) meta.AttributeSelectedClassifier '-E ĊfsSubsetEval -P 6 -E 6ⁿS ĠreedyStepwise -T -1.7976931348623157E308 -N -1 -num-slots 1ⁿW trees.J48 – -C 0.2
(6) meta.AttributeSelectedClassifier '-E ĊfsSubsetEval -P 6 -E 6ⁿS ḂestFirst -D 2 -N 5ⁿW trees.J48 – -C 0.25 -M 2' -1151805453487947520
(7) meta.AttributeSelectedClassifier '-E ĊfsSubsetEval -P 6 -E 6ⁿS ḂestFirst -D 2 -N 5ⁿW trees.RandomForest – -P 100 -I 100 -num-slots 1 -K 0 -M 1.0 -
(8) meta.AttributeSelectedClassifier '-E ĊfsSubsetEval -P 6 -E 6ⁿS ḂestFirst -D 2 -N 5ⁿW trees.RandomTree – -K 0 -M 1.0 -V 0.001 -S 38' -11518054534

machine learning

the results are as seen in table 3 below:

Table 3: Tabel with a T test performed on the percentage each algorithm correctly predicted.

Dataset	(1) rules.On	(2) trees	(3) trees	(4) trees	(5) meta.	(6) meta.	(7) meta.	(8) meta.
percentage correctly classified	59.74	50.65	66.23	35.06 *	45.45	42.86 *	62.34	46.75
Significance	(v = better/ Same /* = Worse)	(0/1/0)	(0/1/0)	(0/0/1)	(0/1/0)	(0/0/1)	(0/1/0)	(0/1/0)

As can be seen from these results in table 3 nothing really stands out from the rest, they all perform quit bad especially if u compare it with the OneR and ZeroR. For further, because of these result and the dimensions of the data set as seen in tabel 1 and figure 5, further machine learning will be focused on the AttributeSelectedClassifier

So for further testing we firstly make a good baseline with the zeroR and further make comparisons with mutiple combinations of the meta.AttributeSelectedClassifier and its parameters.

The next set of test where done in the weka explorer gui.

The first of these is a ZeroR and the rest are the results of a few of the best AttributeSelectedClassifier, since most of them are extremely poor performing and not worthy of mentioning the results at all. After showing these results we shall make a conclusion about which algorithm performs the best. All of the following test runs have been made with crossvallidation with the leave one out method to maximise the limited number of instances in the data.

Table 4: Tabel with the summary of results from zeroR

Correctly Classified Instances	51	66.2338
Incorrectly Classified Instances	26	33.7662
Kappa statistic	0	x
Mean absolute error	0.2665	x
Root mean squared error	0.3613	x
Relative absolute error	100	x
Root relative squared error	100	x
Total Number of Instances	77	x

Table 5: Confusion matrix

a	b	c	d	<– classified as
0	10	0	0	a = T1
0	51	0	0	b = T2
0	11	0	0	c = T3
0	5	0	0	d = T4

ZeroR

As we can see from the results of ZeroR in *tabel 5* that even classifying everything as T2 scores 66% good, thus using that as a base of evaluating the classifiers used is not very reliable, and we shall take the confusing matrix as the first metric to see if a particular model has any merit to for further analyses and testing/

AttributeSelectedClassifier with cost sensitive J48

This is the second algorithm used, and this is using attribute selection with sub set evaluation based on the best first method. as a cost matrix in I assigned every wrongly classified instance as class T2 extra heavy since that class is overrepresented, furthermore it is added that every clas that is wrongly classified is weight little heavier than a T2 class that is wronly classified.

Relation: R_data_frame
Instances: 77
Attributes: 9200
Test mode: 77-fold cross-validation
Evaluation cost matrix:

0 5 2 2
1 0 1 1
1 5 0 2
2 5 2 0

=== Attribute Selection on all input data ===

Search Method:
Best first.
Start set: no attributes
Search direction: forward
Stale search after 5 node expansions
Total number of subsets evaluated: 211334
Merit of the best subset found: 0.601

Attribute Subset Evaluator (supervised, Class (nominal): 9200 data.class):
CFS Subset Evaluator
Including locally predictive attributes

Selected attributes: 338,905,1188,1230,1555,2172,2277,2821,3196,3333,3719,3932,5844,6802,7234,7490,7959,8149,8538
: 19

Table 6: Tabel with the summary of results from zeroR

Correctly Classified Instances	22	28.5
Incorrectly Classified Instances	55	71.4
Kappa statistic	-0.1735	x
Total Cost	135	
Average Cost	1.7532	
Mean absolute error	0.3473	
Root mean squared error	0.5733	
Relative absolute error	129.1465	
Root relative squared error	156.9896	
Total Number of Instances	77	

Table 7: Confusion matrix

a	b	c	d	<– classified as
1	7	2	0	a = T1
8	19	19	5	b = T2
2	8	1	0	c = T3
0	4	0	1	d = T4

in the above seen confusion matrix *table 7* we are looking for a nicely made line from the top left to the bottom right, but we can clearly see that that is not the case. So this algorithm doesn't have a lot of merit for further exploration

AttributeSelectedClassifier HoeffdingTree

In this third algorithm a HoeffdingTree was used as the base algorithm, although this algorithm is normally used for web based streaming inputs and thus not really applicable to this data. it was one of the better ones according to its accuracy but worse on other aspect as well see further on in its results.

Relation: R_data_frame
Instances: 77
Attributes: 9200
Test mode: 77-fold cross-validation
Evaluation cost matrix:

0 5 2 2
1 0 1 1
1 5 0 2
2 5 2 0

=== Attribute Selection on all input data ===

Search Method:
Best first.
Start set: no attributes
Search direction: bi-directional
Stale search after 5 node expansions
Total number of subsets evaluated: 248373
Merit of best subset found: 0.604

Attribute Subset Evaluator (supervised, Class (nominal): 9200 data.class):
CFS Subset Evaluator
Including locally predictive attributes

Selected attributes: 338,1188,1230,1555,2172,2277,2844,3196,3333,3719,3932,5844,6802,7234,7490,7959,8149,8538
: 18

Table 8: Tabel with the summary of results from HoeffdingTree

Correctly Classified Instances	45	58.4
Incorrectly Classified Instances	32	41.5
Kappa statistic	-0.0584	x
Mean absolute error	0.2337	x
Root mean squared error	0.4332	x
Relative absolute error	86.9241	x
Root relative squared error	118.6126	x
Total Number of Instances	77	x

Table 9: labelHoeffdingTree confusionmatrixConfusion matrix

a	b	c	d	<– classified as
0	9	1	0	a = T1
5	45	1	0	b = T2
1	10	0	0	c = T3
0	5	0	0	d = T4

in the above seen confusion matrix *table 9* we are still looking for a nicely made line from the top left to the bottom right, but we can clearly see that that is not the case, and it classifies them mostly as t1 and t2.

AttributeSelectedClassifier Ranker RandomTree
Relation: R_data_frame
Instances: 77
Attributes: 9200
Test mode: 77-fold cross-validation
Evaluation cost matrix:

0 5 2 2
1 0 1 1
1 5 0 2
2 5 2 0

Selected attributes: 4526,6015,1443,1312,8523,2172,1230,5384,2277,4206,8329,1188 : 12

Table 10: Tabel with the summary of results from Randomtree

Correctly Classified Instances	42	54.5
Incorrectly Classified Instances	35	45.5
Kappa statistic	0.1126	x
Mean absolute error	0.226	x
Root mean squared error	0.4726	x
Relative absolute error	84.0	x
Root relative squared error	129.4	x
Total Number of Instances	77	x

Table 11: labelRandomTree confusionmatrixConfusion matrix

a	b	c	d	<– classified as
3	4	3	0	a = T1
3	38	7	3	b = T2
5	6	0	0	c = T3
0	4	0	1	d = T4

confusion matrix

in the above seen confusion matrix we are looking for a nicely made line from the top left to the bottom right, but we can clearly see that that is not the case, and it classifies them mostly wrong as t2. So this algorithm doesn't have a lot of merit for further exploration

AttributeSelectedClassifier greedystepwise with OneR
Relation: R_data_frame
Instances: 77
Attributes: 9200
Test mode: 77-fold cross-validation
Evaluation cost matrix:

=== Attribute Selection on all input data ===

Search Method:
Greedy Stepwise (forwards).
Start set: no attributes
Merit of best subset found: 0.601

Attribute Subset Evaluator (supervised, Class (nominal): 9200 data.class):
CFS Subset Evaluator
Including locally predictive attributes

Selected attributes: 338,905,1188,1230,1555,2172,2277,2821,3196,3333,3719,3932,5844,6802,7234,7490,7959,8149,8539 : 19

Table 12: Tabel with the summary of results from OneR

Correctly Classified Instances	45	58.44
Incorrectly Classified Instances	32	41.55
Kappa statistic	-0.0788	x
Total Cost	133	
Average Cost	1.7273	
Mean absolute error	0.2078	x
Root mean squared error	0.4558	x
Relative absolute error	77.2714	x
Root relative squared error	124.8216	x
Total Number of Instances	77	x

Table 13: labelgreedystepwise with OneR confusionmatrixConfusion matrix

a	b	c	d	<– classified as
0	9	1	0	a = T1
2	45	4	0	b = T2
0	11	0	0	c = T3
0	5	0	0	d = T4

in the above seen confusion matrix we are looking for a nicely made line from the top left to the bottom right, but we can clearly see that that is not the case, we can see that is mostly classify them as T2. So this algorithm doesn't have a lot of merit for further exploration.

final model ultimately for a part of the assignment it was needed to creat a Java wrapper to perform classification of new instances for a newly trained model according to this research and these are the result for the model used. since it had the "best" Roc curve and the least amount of cost.

Table 14: Tabel with the summary of results

Correctly Classified Instances	37	48.05
Incorrectly Classified Instances	40	51.94
Kappa statistic	-0.0052	
Total Cost	116	
Average Cost	1.5065	
Mean absolute error	0.2582	
Root mean squared error	0.5062	
Relative absolute error	96.	
Root relative squared error	138.6	
Total Number of Instances	77	

Summary

Table 15: Cost matrix

0	5	2	2
1	0	1	1
1	5	0	2
2	5	2	0

Table 16: Cost matrix

0	20	20	20
1	0	1	1
20	20	0	20
20	20	1	0

Table 17: labelFinal model accuracy final model accuracy

	TP Rate	FP Rate	Precision	Recall	F-Measure	MCC	ROC Area	PRC Area	Class
	0,100	0,134	0,100	0,100	0,100	-0,034	0,494	0,133	T1
	0,647	0,692	0,647	0,647	0,647	-0,045	0,535	0,709	T2
	0,273	0,167	0,214	0,273	0,240	0,096	0,594	0,177	T3
	0,000	0,028	0,000	0,000	0,000	-0,043	0,631	0,112	T4
Weighted Avg.	0,481	0,502	0,472	0,481	0,476	-0,023	0,545	0,520	

cost matrix

Table 18: labelcost sensitive Random TreeConfusion matrix

a	b	c	d	<– classified as
1	7	2	0	a = T1
8	33	8	2	b = T2
0	8	3	0	c = T3
1	3	1	0	d = T4

confusion matrix

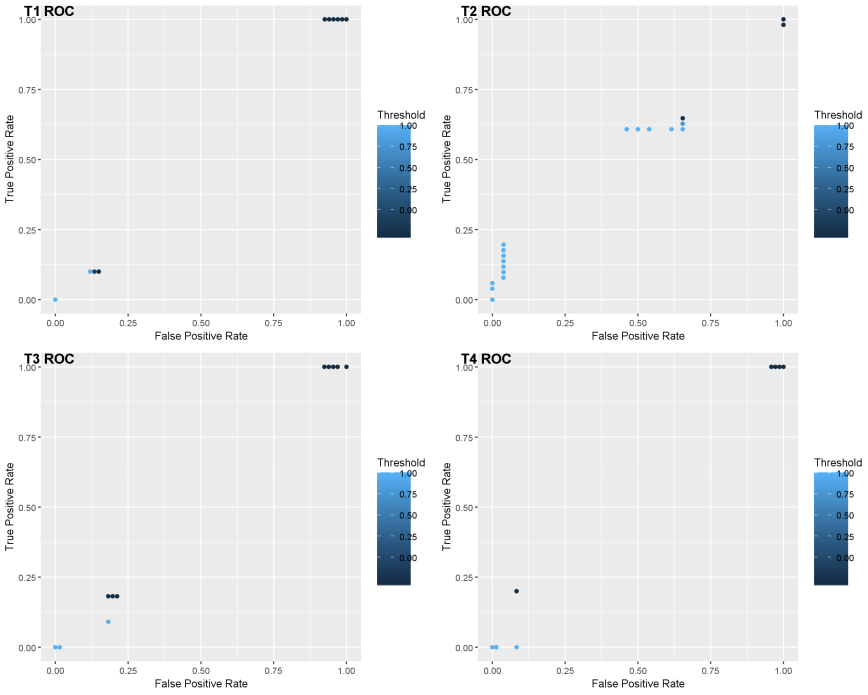

Figure 6: Roc curves per class

Here in the Last figure 6 the Roc curves are shown for the model ultimately used in the java wrapper that was build for easy classification of new samples. It is clearly visible that the model isn't good at all. With almost a straight line between bottom left and top right between the points added. With these Roc curves u want as much area under the curve and the ~50% of the area is under the "curve"

Machine learning summary /newline these are but the best of the multiple different settings that where tried, but all results where the same as these presented in the chapters here above. one if not the first things that stands out for every one of these results is that none of these results scored an accuracy of correctly predicting the class of the data better than ZeroR with its 66.2%. One got close but that was a RandomTree model without attributeselection, so it is questionable how accurate its truly is. But accuracy of correctly prediction the class is not everything, so we must also look at the confusion matrix's that where produced. As can be seen from the confusion matrix's (tables) there are two trends visible that firstly there is a huge bias towards t2 Thus, most of the models that where teste where with the meta learning AttributeSelectedClassifier , and yielded not much as explained above. The main thing that can be said of all the methods and algorithms used is they performed all bad or equally bad as just picking a random class. One more thing is that a significant portion the models that where tried with the AttributeSelectedClassifier is that they almost all chose these attributes seen in table 14

Table 19: labelProtein Selection Most selected protein

Protein RefSeq
NP_008832
NP_056289
NP_001349
NP_001349
NP_055719
NP_001150
NP_079093
NP_000959
NP_004893
NP_004887
NP_065901
NP_058632
NP_002630
NP_060947
NP_065109
NP_848613
NP_001035147
NP_005639
NP_001092102
NP_001017

4 Discussion and Conclusion

In the results' section we can see from the figures 1 and two that although the data set was supplied with the label as high quality there are still proteins in the data with more than 10% of their expression values missing. This combined with the need for using a Regex expressing and string formatting to alter the sample names of the protein expression data set, so it could be merged/ paired with the sample names of the clinical dataset. Wasn't something expected of high quality data. Also in figure 5 it is clearly visible that T2 stage is oversampled and creates a bias in every algorithm used, see the results from ZeroR tables 6 and 7. If we look at the results from the Weka experimenter tabel 2 and 3 we can see that they perform all quite bad with a simple randomForest algorithm produced the highest percentage of correctly classified instances, but the model that resulted from this was majorly over fitted with access to all 9200 attributes in the data. This problem became more apparent when looking at the others results generated with trying more algorithms in the weka explorers using the Attribute Selected Classifier in different configurations as seen in tables 4 to 13. furthermore the sheer amount of proteins recorded in this data is very useful for my purpose of trying to use another classification as the PAM50 protein list, but this also let to some problems. first it forces us to first make a selection on which attributes shall be used for classification, since u want the number of attributes used for selection to be around 10 for the set of 77 samples in this data. This is because the model build can seem to be very good, but over fitted on its training data but when used on any real data perform very bad. We can see this effect more in place when looking at the results in table 2 and 3 and compare that with the same kind of base algorithem used for the final model tabels

The Pam50 list builds upon multiple years of biological knowledge and research [3], and this research is for a school project with a relatively simple machine learning part and usage.

5 Project Proposal

6 References

DeSantis, Carol E., Jiemin Ma, Mia M. Gaudet, Lisa A. Newman, Kimberly D. Miller, Ann Goding Sauer, Ahmedin Jemal, and Rebecca L. Siegel. 2019. "Breast Cancer Statistics, 2019." *CA: A Cancer Journal for Clinicians* 69 (6): 438–51. https://doi.org/https://doi.org/10.3322/caac.21583.

Mertins, Philipp, D R Mani, Kelly Ruggles, Michael Gillette, Karl Clauser, Pei Wang, Xianlong Wang, et al. 2016. "Proteogenomics Connects Somatic Mutations to Signaling in Breast Cancer." *Nature* 534 (May). https://doi.org/10.1038/nature18003.

Raj Kumar, Praveen Kumar, Jeff Liu, Jeffrey Hooke, Albert Kovatich, Leonid Kvecher, Craig Shriver, and Hai Hu. 2019. "PCA-PAM50 Improves Consistency Between Breast Cancer Intrinsic and Clinical Subtyping Reclassifying a Subset of Luminal a Tumors as Luminal b." *Scientific Reports* 9 (May). https://doi.org/10.1038/s41598-019-44339-4.

7 Appendices