Membrane composition prediction using machine learning

Kasper Notebomer

10/2/2021

Contents

1 Short introduction **1**

2 Introduction **2**

2.1 Objective 2

2.2 Theory 2

3 Materials and methods **3**

3.1 Materials 3

3.2 Existing methods 3

3.3 Developed methods 3

4 Results **4**

4.1 Data exploration 4

4.2 Data cleaning 10

4.3 Machine learning 11

5 Discussion and Conclusion **16**

5.1 Results summary 16

5.2 Discussion 16

5.3 Conclusion & Future research 18

6 Project proposal for minor **19**

7 Refereces **20**

8 Apendix **21**

8.1 A, The code 21

8.2 B, The codebook 26

8.3 C, LMT algorithm 26

1 Short introduction

This report describes the results of the exploratory data analysis and data cleaning done on a dataset containing data on membrane composition and characteristics. The data was provided by Tsjerk Wassenaar of the RuG. The data set contains data on membrane composition and characteristics. The data set is not publicly available. The data set also does not have a publication linked to it. There is no paper linked to the dataset and there was little information given in general.

The data set contains 14 variables and 2843 different instances. The goal is to use measurement/membrane characteristics like the bending rigidity to predict the composition of the membrane. Typically you would use independent variables to predict one dependent variable. In this data set this doesn't seem to be the case, seeing as the variables that are given as parameters are seen as class variables, this means that these would be considered the labels. The parameters are: Temperature, Sterol type, Sterol concentration, Other (phospho)lipids in membrane, Aliphatic tails, Saturation index, Phosphatidyl choline concentration and Ethanol concentration. So the biggest question that the EDA should answer is which of these variables is most interesting to use as the label, or could we even predict multiple of them using the given variables.

The main goals of the EDA where to give insight into the data, and to prepare the data for machine learning experiments. Giving insight into the data means looking for relations between variables and finding clusters. Cleaning the data consists of things like transforming values and scaling the data.

2 Introduction

2.1 Objective

The aim of this research is to try and find a way to predict membrane composition given the characteristics. This is done with the use of machine learning algorithms. The found model can then be made into an application that can easily predict the membrane composition when given specific membrane characteristics as input by the user. This leads to the following research question: “Is it possible to predict membrane composition based on membrane characteristics, with a high degree of accuracy, using machine learning?” The H0 hypothesis is as follows: It is possible to predict membrane composition with membrane characteristics. To answer the research question thoroughly, the question was split up into three sub questions: “Is it possible to predict the sterol that is present in a membrane based on the membrane’s characteristics?” “Is it possible to aliphatic tails that are present in the lipids of a membrane given the membrane characteristics?” and “Is is possible to predict the sterol concentration of a membrane given the membrane characteristics?”

2.2 Theory

In some research it is necessary or interesting to create a membrane that has specific characteristics, like a specific area per lipid or bending rigidity. However, due to the many different possible combinations of membrane components, creating a specific membrane can be quite a daunting task. There is a lot of research that shows what kind of an effect certain components have on a membranes characteristics, but there isn’t really anything that tries to predict what components are present in a membrane, given the characteristics. A problem like this is most likely solvable using a combination of machine learning algorithms. Machine learning algorithms are algorithms that get more accurate as they are exposed to more data. A few examples of machine learning algorithms are Naive Bayes, RandomForest and LMT (Landwehr, Hall, and Frank 2005).

3 Materials and methods

The code used for this research can be found at the following repository: https://github.com/Skippybal/Thema9. The application that was made using this research can be found at: https://github.com/Skippybal/Project-Actaeon.

3.1 Materials

The dataset that was used contains data on membrane composition and characteristics. The data was provided by Tsjerk Wassenaar of the RuG. The dataset is, at the time of writing, not publicly available. The data set also does not have a publication linked to it. There is no paper linked to the dataset and there was little information given in general. In the dataset 14 different variables where recorded and there where 2843 instances recorded in total. The recorded varaibles are: temperature, sterol type, sterol concentration, other (phospho)lipids in membrane, aliphatic tails, saturation index, phosphatidyl choline concentration, ethanol concentration, area per lipit (APL), thickness, bending regidity, tilt angle, zorder and compressibility. Some of these variabels are considered paramaters, this means that they are meant to be the labels in machine learning. The parameters are: Temperature, Sterol type, Sterol concentration, Other (phospho)lipids in membrane, Aliphatic tails, Saturation index, Phosphatidyl choline concentration and Ethanol concentration. This means that the other variables should be used to predict one or multiple of the parameters.

Any kind of data manipulation and the explanatory data analysis phase where done in the R programming language (R Core Team 2020). The machine learning experiments where done using the Weka software (Waikato 2020).

3.2 Existing methods

The machine learning experiments where done in Weka. A lot of different algorithms where tested against each other on whether or not their accuracy was significantly better then other algorithms. This was done using a t-test with a p-value of 0.05. The algorithms that where tested include, but aren't limited to: Naive Bayes, J48, SMO, Simple Logisic, ZeroR, OneR, RandomForest and LMT. Some of there algorithms where also used in certain forms of ensemble learning like stacking, boosting and voting.

3.3 Developed methods

A resutlt form this research is the program that can be found at: https://github.com/Skippybal/Project-Actaeon. This repo contains a program, written in java, that can predict the tails, sterol presensce and sterol concentration when given a membrane's characteristics. The program is a wrapper for the Weka models created by this research. The input for the program consists of the following variables: area per lipit (APL), thickness, bending regidity, tilt angle, zorder and compressibility.

4 Results

The results are divided into two different subsection: data exploration, data cleaning and machine learning. This is because all of these sections have different goals and are major subject of discussion. The code for both the data exploration steps and the data cleaning can be found in appendix A. All of the abbreviation are explained in the codebook, shown in appendix B.

4.1 Data exploration

The biggest goal of the exploration phase was to give insight into the data, like relations between variables. Using this insight, the research question could be tweaked to have a higher likelihood of being achievable. Another question the exploration phase was looking to answer was whether or not the dataset was suitable for machine learning.

Table 1: Five number summary of the original dataset

	temperature	sterol.conc	satur.index	PC.conc	ethanol.conc	APL	thickness	bending	tilt	zorder	compress
Minimum	298.0	0.00	0.0000	0.00	0	0.4514	3.003	0.2025	6.774	0.02736	1.286
Q1	298.0	10.00	0.0000	25.00	5	0.6350	3.627	9.8579	14.490	0.18604	23.097
Median	298.0	20.00	0.5000	50.00	15	0.6885	3.812	12.1000	18.157	0.25860	31.346
Mean	305.2	16.55	0.6208	50.02	15	0.6807	3.934	22.0256	35.732	0.33269	48.714
Q3	298.0	30.00	1.0000	75.00	25	0.7394	4.238	25.5712	35.933	0.45559	43.348
Maximum	328.0	30.00	2.0000	100.00	30	0.8924	5.101	125.3820	227.813	0.93013	538.446
Number of NA's	0	0	0	0	0	104	103	104	104	153	103

Table 1 shows the five number summary of all of the numerical columns that are present in the dataset. Inspecting this table closely, a deviation in the mean and the median of some of the columns can be observed. This is an indication that our data might be skewed. This is an important observation, seeing as some algorithms expect the data to have a Gaussian(normal) distribution. The results of calculating the exact skew of all of the numeric columns can be found in table 2. The table shows that there is indeed a moderate (|Skew|> 0.5) to heavy skew (|Skew| > 1.0) in almost every column. Another thing to take note of in table 1 is the extremely high maximum values of the bending and the tilt columns. This tells us that there are probably some outliers in these columns. To confirm this, a boxplot is made. The results of this are shown in figure 1.

Table 2: Skew in numeric columns

	temperature	sterol.conc	satur.index	PC.conc	ethanol.conc	APL	thickness	bending	tilt	zorder	compress
Skew	1.125717	-0.1719109	0.9075759	0.0008403	0.0149497	-0.6069614	0.7652531	2.674385	2.632133	1.146385	3.93279

Figure 1 shows that there are indeed a lot of outliers to be found in the dataset, especially in the bending rigidity, compressibility and tilt ange columns. Outliers are shown as black dots in the plot. Some variables also have very clean distributions without whiskers. This is mostly because some variables are numerical but only contain a few different levels because of the experiment that they where gathered from.

Any type of normalization or transformation was dealt with in the data cleaning. The NA's that are present were also be removed in the data cleaning.

The results of plotting some variables and coloring them based on certain class variables can be seen in figure 2. These density plots are made to see whether or not it might by possible to distinguish between class variables based on one specific variable.

Plot A in figure 2 show that there is basically no distinction between cholesterol and ergosterol based on the bending rigidity. However, distinguishing between no sterol or a sterol does seem to be possible based on the bending rigidity seeing as the peaks of these classes only overlaps a small bit. There does seem to be an odd peak in the no sterol class at around 50 kB/t bending rigidity.

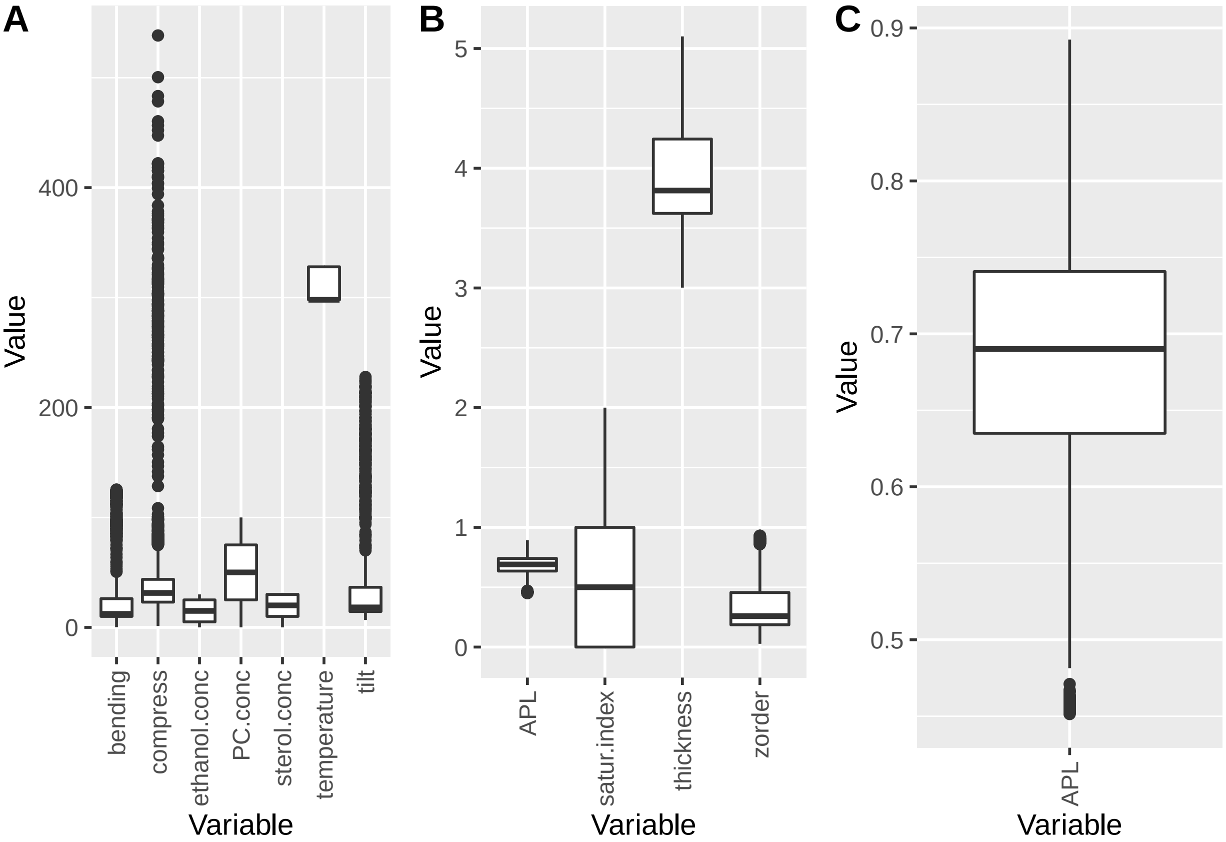

Figure 1: Distribution of data per variable.

In plot B it looks like there is no way of distinguishing different types of sterol based on the area per lipid, seeing as the peaks are basically in the same place. But just to be certain we'll still plot it against some other variables to make sure, seeing as it might be a very useful variable when paired with something like bending rigidity. The results depicted in plot C also don't look very promising seeing as every peak is around the same place again. Plot D on the other hand seems a lot more interesting seeing as all of the peaks are in slightly different locations, the DO, PO and the PI seem to overlap a lot but the DP and DI tails seem to have little overlap with the rest. It seems like APL could possibly be used as a variable to distinguish between aliphatic tails when paired with another variable.

When looking at all of the plots it looks like most of the data is skewed in one way or another, confirming the numbers given by the five number summary.

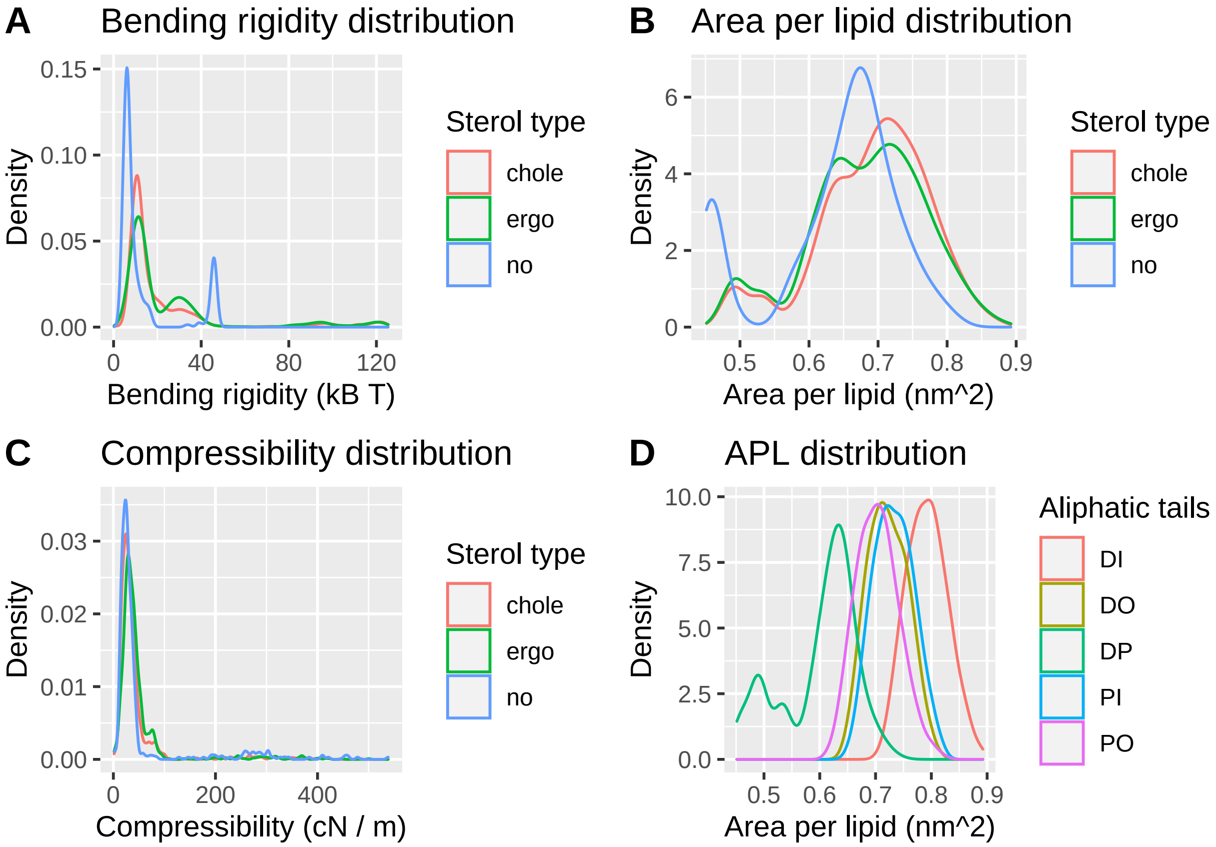

Figure 2: Distribution of variables, coloured by membrane components

An interesting correlation that was found in the data exploration is shown in figure 3. It shows that there is an inverse logarithmic correlation between the area per lipid of a membrane and the bending rigidity. An even more interesting observation is the fact that, when colored on sterol type, it shows that there is a big difference between the no sterol group and the other two groups. This means that this ratio could be very important in the prediction of sterol type based on the membrane characteristics. It does however seem like these variables aren't useful when it comes to discerning between ergosterol and cholesterol.

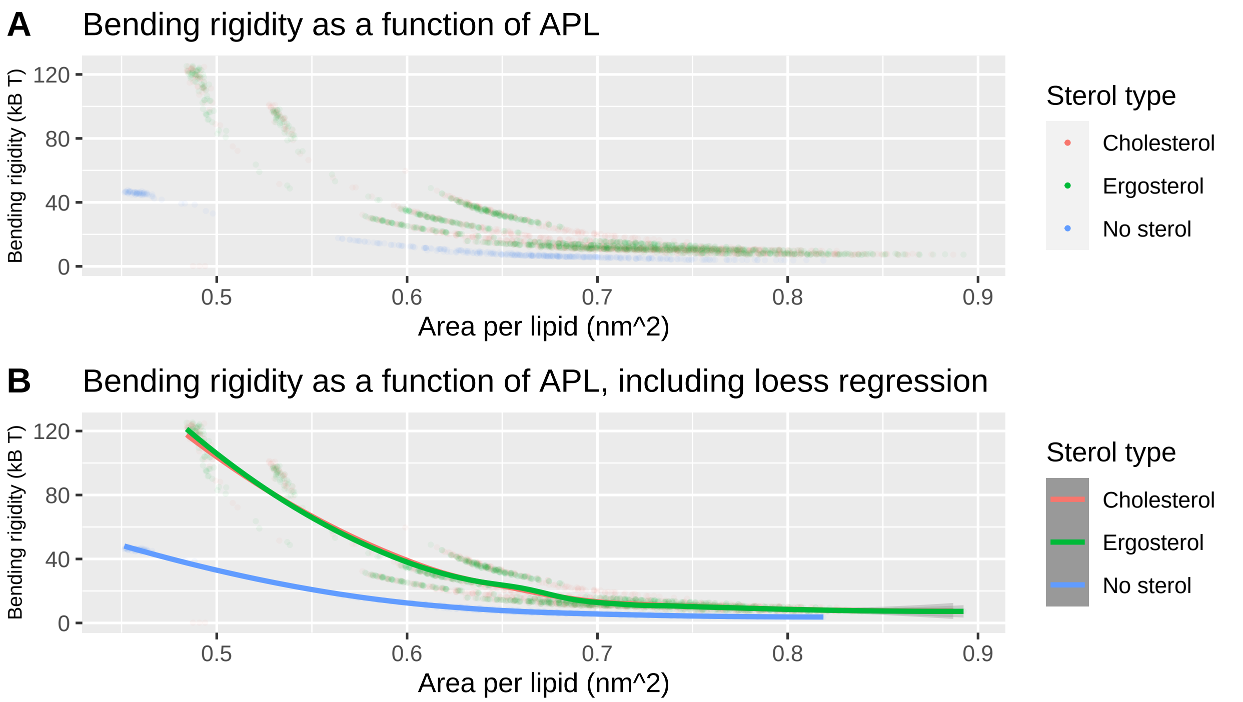

Figure 3: Correlation between APL and bending regidty, (B) includes a loess regresison line. Loess regression shows an inverse logorithmic corrolation based on sterol presence.

The results of plotting the compressibility of the membrane against the bending rigidity can be found in figure 4. In plot A of figure 4 we can't really make out much of a pattern in the dense clout to the left, but when looking at the rest of the data points there does seem to be some grouping based on whether or not no sterol or a sterol is present. When zooming in in plot B it shows the same kind of grouping happen again based on whether there is a sterol present or not. Something else that figure 4 shows it that a log transformation on the compressibility variable might be in order seeing as there are a lot of point in the magnitude of tens but also a lot in the hundreds.

In figure 5 the area per lipid is again plotted against the bending rigidity, only this time, the plot is colored based on the saturation index.

The results shown in figure 5 show an interesting pattern. It seems that a specific saturation index is only found in membranes within a specific range of area per lipid and bending rigidity. This is shown by the fact that there seem to be little clusters of color that have formed in the graph. A saturation index of zero and two seem to be the most well defined, seeing as these clusters are basically separate from the other saturation indexes. The saturation indexes of zero point five and one on the other hand seem to have a lot of overlap between them. All of this means that area per lipid to bending rigidity ratio could also be promising in classifying the saturation index of a membrane.

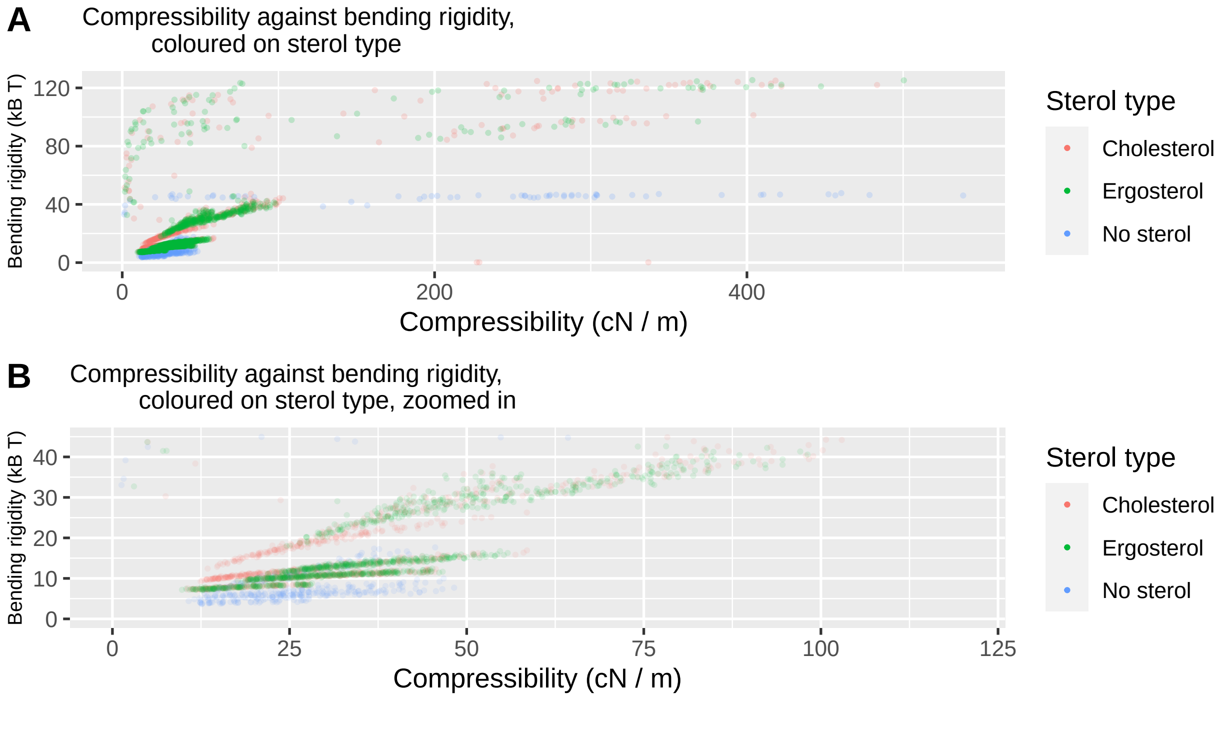

Figure 4: Correlation between bending rigidity and compressibility depending on the sterol type.

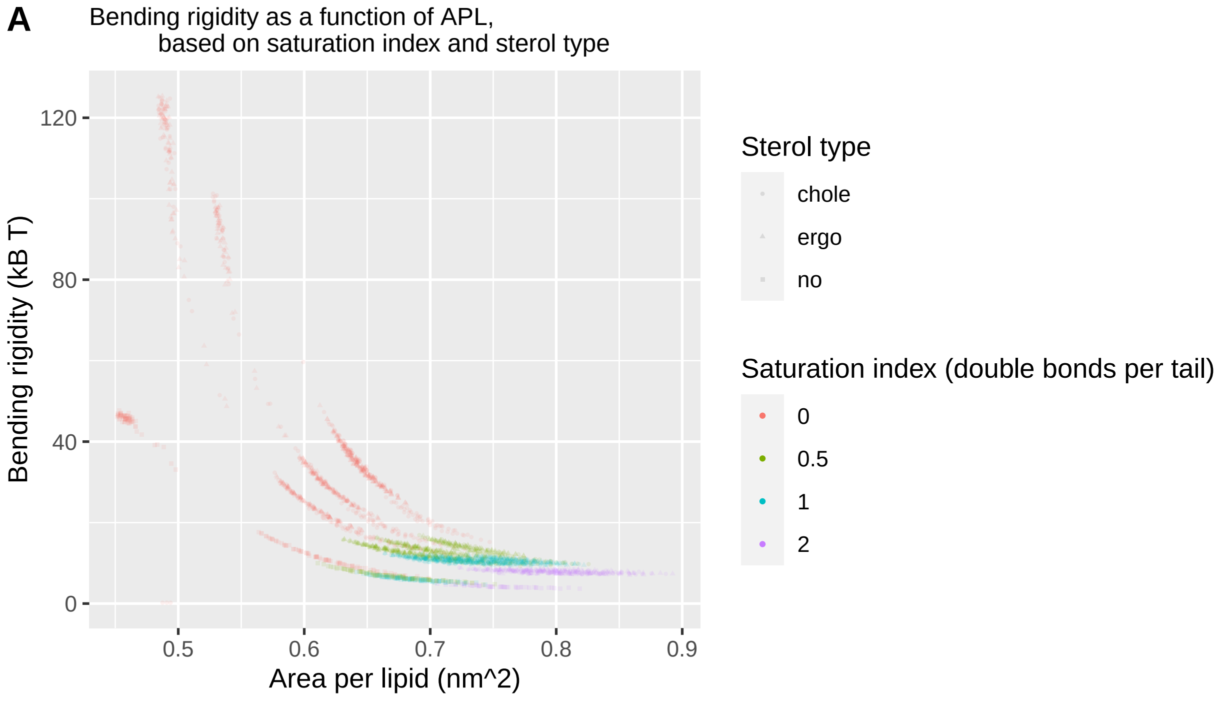

Figure 5: Correlation between APL and bending regidity based on the sterol type and saturation index.

To look for correlation between the variables and to look for clustering a PCA plot was made. This plot is shown in figure 6.

Plot A in figure 6 shows something that is quite strange. It states that over 80% of the variance can be explained by the first principle component, this is quite odd. Plot B shows that some of the variables seem to be highly correlated like z order and thickness just like tilt angle and bending rigidity seeing as their arrows are pointing in the same direction. Area per lipid(APL) on the other hand seems to be negatively correlated to z order and thickness. When looking at plot C there doesn't seem to be any obvious form of clustering, if any, for that matter.

All of the results from the data exploration led to the following, refined, research question: "Is it possible to predict membrane composition based on membrane characteristics, with a high degree of accuracy, using machine learning?" With a focus on the sterol type, aliphatic tails and sterol consecration components.

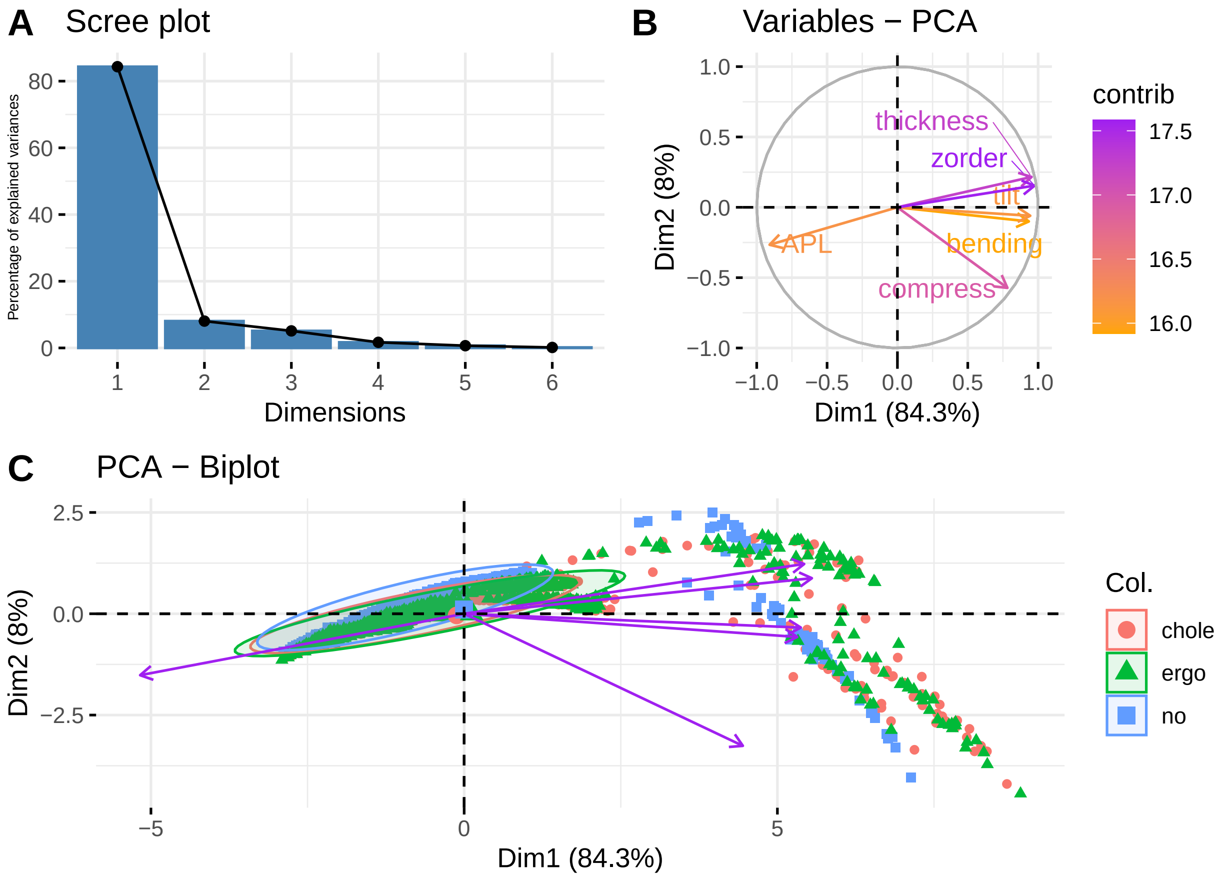

Figure 6: Clustering of samples using PCA.

4.2 Data cleaning

To make the dataset usable for machine learning it had to be cleaned up using various of methods. The difference between the cleaned dataset and the original dataset can be seen when comparing table 3 and table 4. Unused columns like temperature and ethanol concentration where removed. All of the data in the columns was scaled using min max normalization. This is visible in table 4, seeing as the values now range from zero to one. NA's where also omitted. It was decided not to remove any variables based on a high degree of correlation. Table 4 doesn't have all of the variables that table 3 has. This is because the variables that where removed would be seen as labels from a machine learning point of view. Seeing as this research is interested in only one of the labels, sterol type, the others where removed.

Table 3: Top of original data

temperature	sterol.type	sterol.conc	other.phosph	tails	satur.index	PC.conc	ethanol.conc	APL	thickness	bending	tilt	zorder	compress
298	chole	20	PE	PI	1	33	20	0.736911	3.628973	10.08020	14.7578	0.175499	24.18218
298	chole	20	PE	PI	1	50	20	0.746226	3.596991	10.14030	14.3075	0.172084	24.32495
298	chole	20	PE	PI	1	25	20	0.731945	3.652637	10.25470	14.9777	0.178494	23.76607
298	chole	20	PE	PI	1	100	20	0.769630	3.496052	10.02320	13.2029	0.163349	22.96077
298	chole	20	PE	PI	1	0	20	0.719509	3.710018	10.46540	15.7530	0.182191	23.10503
298	chole	20	PE	PI	1	75	20	0.760091	3.538908	9.86163	13.7148	0.166807	23.86050

Table 4: Top of cleaned data

APL	thickness	bending	tilt	zorder	compress	sterol.type
0.6474103	0.2983812	0.0789085	0.0361181	0.1640910	0.0426240	chole
0.6685354	0.2831408	0.0793886	0.0340809	0.1603082	0.0428898	chole
0.6361482	0.3096582	0.0803025	0.0371130	0.1674086	0.0418494	chole
0.7216122	0.2350388	0.0784531	0.0290836	0.1506323	0.0403502	chole
0.6079452	0.3370028	0.0819857	0.0406205	0.1715037	0.0406188	chole
0.6999791	0.2554618	0.0771624	0.0313995	0.1544628	0.0420252	chole

In table 5, the five number summary of the cleaned dataset can be found. It shows that all of the data is now on a common scale of zero to one, and that all of the NA's from the original dataset where omitted. This can be deduced from the fact that there is no line present in the table with the number of NA's, seeing as when there are no NA's present, this line won't be returned.

Table 5: Five number summary of the cleaned dataset

	APL	thickness	bending	tilt	zorder
Minimum	0.0000	0.0000	0.00000	0.00000	0.0000
Q1	0.4163	0.2954	0.07817	0.03477	0.1758
Median	0.5412	0.3865	0.09643	0.05161	0.2561
Mean	0.5211	0.4453	0.17643	0.13251	0.3382
Q3	0.6560	0.5915	0.20684	0.13454	0.4743
Maximum	1.0000	1.0000	1.00000	1.00000	1.0000

4.3 Machine learning

The dataset for every machine learning experiment was manipulated in such a way that the column containing the label was put as the last column of the dataset. The first few columns where always: area per lipit (APL), thickness, bending regidity, tilt angle, zorder and compressibility. In that order. All of the experiments where done with a p-value of 0.05.

4.3.1 Sterol type

Table 6: Sterol prediction accuracy

Dataset	(1)	(2)	(3)	(4)	(5)	(6)	(7)	(8)	(9)
R-data-frame	42.94	56.65 ○	59.51 ○	47.83 ○	57.30 ○	60.20 ○	57.07 ○	58.03 ○	70.03 ○

○, ● statistically significant improvement or degradation

Table 7: Sterol prediction accuracy (Key)

(1)	rules.ZeroR " 48055541465867954
(2)	rules.OneR '-B 50' -3459427003147861443
(3)	lazy.IBk '-K 100 -W 0 -A \"weka.core.neighboursearch.LinearNNSearch -A \\\"weka.core.EuclideanDistance -R first-last\\\"\"' -3080186098777067172
(4)	bayes.NaiveBayes " 5995231201785697655
(5)	trees.J48 '-C 0.25 -M 200' -217733168393644444
(6)	trees.RandomForest '-P 100 -I 100 -num-slots 1 -K 0 -M 1.0 -V 0.001 -S 1' 1116839470751428698
(7)	functions.SMO '-C 1.0 -L 0.001 -P 1.0E-12 -N 0 -V -1 -W 1 -K \"functions.supportVector.PolyKernel -E 1.0 -C 250007\" -calibrator \"functions.Logistic -R 1.0E-8 -M -1 -num-decimal-places 4\"' -6585883636378691736
(8)	functions.SimpleLogistic '-I 0 -M 500 -H 50 -W 0.0' 7397710626304705059
(9)	trees.LMT '-I -1 -M 15 -W 0.0' -1113212459618104943

Table 6 shows the results of some basic machine learning models in predicting the sterol type of a give membrane, the settings for every algorithm can be found in table 7. The highest accuracy that is achieved is equal to 70.03%. Table 8 shows a more in depth view of the accuracy per label for the RandomForest and the SimpleLogitc model, in the form of a confusion matrix. This table shows something quite interesting, it seems like the the reason for the low accuracy is the distinction between ergosterol and cholesterol. This leads to the idea that machine learning algorithms are capable of predicing whether or not a sterol is present in the given membrane, but aren't able to discern between ergosterol and cholesterol. This coincides with the previous findings in the Data exploration section.

Table 8: Confusion matrix from RandomForest and SimpleLogistics, sterol type

RandomForest			
a	b	c	<– classified as
608	535	0	a = chole
261	893	1	b = ergo
1	4	387	c = no

SimpleLogistics			
a	b	c	<– classified as
624	511	8	a = chole
585	564	6	b = ergo
18	2	372	c = no

Table 9: Recoded sterol type prediction accuracy

Dataset	(6)	(1)	(2)	(3)	(4)	(5)	(7)	(8)	(9)
Sterol-Present	99.75	85.43 ●	93.36 ●	92.93 ●	86.91 ●	92.78 ●	90.35 ●	98.99 ●	99.94
Removed-No	53.30	50.26 ●	55.67	55.06	53.88	57.31 ○	51.43	52.90	57.54 ○

○, ● statistically significant improvement or degradation

After recoding the sterol type column in the dataset, and running the same experiment, the accuracy went up to 99.75% in the dataset where ergosterol and cholesterol where grouped together (Sterol_Present dataset).

It does seem like machine learning isn’t able to discern between ergosterol and cholesterol. Seeing as in the dataset where the no sterol present instances where removed (Removed-no dataset), the highest achieved accuracy is a measly 57.54%. The results of these experiments are depicted in table 9. The settings for this experiment are the same as the ones shows in table 7

From the gathered results it seems like RandomForest is the most suitable algorithm to use for predicting whether or not there is a sterol present in a given membrane. The gathered results also seem to support the theory that there is no significant difference between the effect of ergosterol and cholesterol on membrance characteristics. Or at least, not one that can be found by the used machine learning algorithms.

4.3.2 Aliphatic tails

Table 10: Tails prediction, RandomForest as base of t test

Dataset	(6)	(1)	(2)	(3)	(4)	(5)	(7)	(8)	(9)
R-data-frame	99.47	37.81 ●	82.28 ●	82.39 ●	77.83 ●	81.03 ●	79.71 ●	99.42	99.94 ○

○, ● statistically significant improvement or degradation

Table 11: Tails prediction, RandomForest base (Key)

(1)	rules.ZeroR ” 48055541465867954
(2)	rules.OneR ’-B 50’ -3459427003147861443
(3)	lazy.IBk ’-K 100 -W 0 -A \"weka.core.neighboursearch.LinearNNSearch -A \\\"weka.core.EuclideanDistance -R first-last\\\"\"’ -3080186098777067172
(4)	bayes.NaiveBayes ” 5995231201785697655
(5)	trees.J48 ’-C 0.25 -M 200’ -217733168393644444
(6)	trees.RandomForest ’-P 100 -I 100 -num-slots 1 -K 0 -M 1.0 -V 0.001 -S 1’ 1116839470751428698
(7)	functions.SMO ’-C 1.0 -L 0.001 -P 1.0E-12 -N 0 -V -1 -W 1 -K \"functions.supportVector.PolyKernel -E 1.0 -C 250007\" -calibrator \"functions.Logistic -R 1.0E-8 -M -1 -num-decimal-places 4\"’ -6585883636378691736
(8)	functions.SimpleLogistic ’-I 0 -M 500 -H 50 -W 0.0’ 7397710626304705059
(9)	trees.LMT ’-I -1 -M 15 -W 0.0’ -1113212459618104943

RandomForest, with an accuracy of 99.47%, seems to be the one of the most accurate algorithm for predicting aliphatic tails when looking at table 10. The only algorithm that preforms significantly better is the LMT algorithm with a 99.94% accuracy. The fact that RandomForest is one of the more accurate models is not unexpected seeing as RandomForest is technically an ensemble learner. The keys/settings for the experiment from table 10 can be found in table 11.

Table 12 depicts the accuracy of multiple ensemble learners compared to the RandomForest algorithm. The keys for this experiment can be found in 13. None of the ensemble learners are significantly better at predicting aliphatic tails than RandomForest. The MultiClassClassifier algorithm is the only algorithm that preforms significantly worse.

Table 12: Ensemble learning tails prediction accuracy

Dataset	(1)	(2)	(3)	(4)	(5)	(6)
R-data-frame	99.47	99.41	99.69	94.41 ●	99.29	99.48

○, ● statistically significant improvement or degradation

Table 14 shows that attribute selection doesn’t make a significant difference in results, the keys for this experiment can be found in table 15. This would lead to the conclusion that it is probably best to run the model with less attributes, seeing as less input is usually better. However, for and application that needs to predict multiple components of a membrane, for the purpose of creating a membrane with specific characteristics, it won’t actually be that useful. This is because every component in a membrane leads to differences in different characteristics. This means that to predicts every component of the membrane, all of the wanted characteristics are probably necessary, seeing as different attributes will be needed for different models to predict different components.

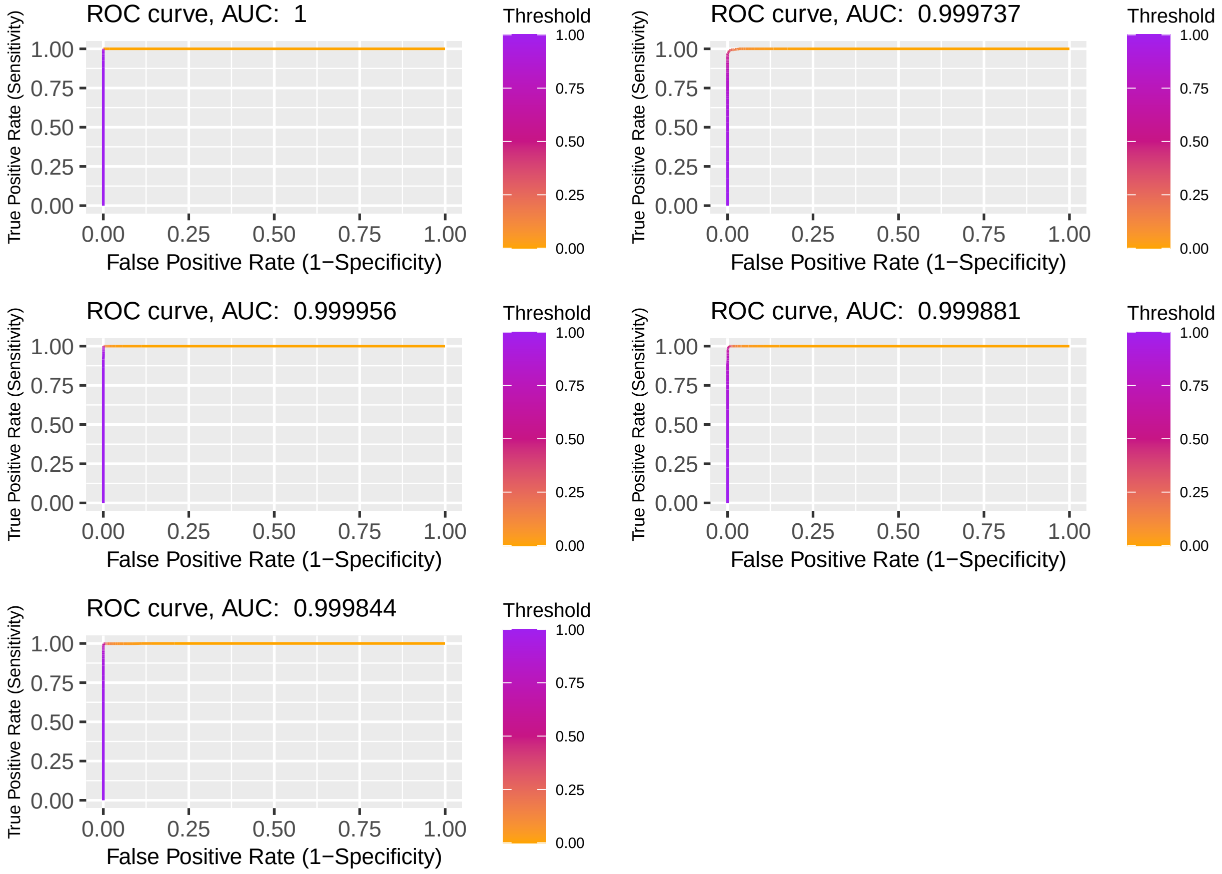

Figure 7: Performence mesurement of the RandomForest algorithm using an ROC curve shows excellent results. The labels for the plots are the following in order from left to right: DI, DO, DP, PI, PO.

Table 13: Ensemble learning tail prediction (Key)

(1)	trees.RandomForest '-P 100 -I 100 -num-slots 1 -K 0 -M 1.0 -V 0.001 -S 1' 1116839470751428698
(2)	meta.Vote '-S 1 -B \"trees.RandomForest -P 100 -I 100 -num-slots 1 -K 0 -M 1.0 -V 0.001 -S 1\" -B \"functions.SimpleLogistic -I 0 -M 500 -H 50 -W 0.0\" -B \"lazy.IBk -K 100 -W 0 -A \\\"weka.core.neighboursearch.LinearNNSearch -A \\\\\\\"weka.core.EuclideanDistance -R first-last\\\\\\\"\\\"\" -R AVG' -637891196294399624
(3)	meta.Stacking '-X 10 -M \"trees.J48 -C 0.25 -M 2\" -S 1 -num-slots 1 -B \"trees.RandomForest -P 100 -I 100 -num-slots 1 -K 0 -M 1.0 -V 0.001 -S 1\" -B \"functions.SimpleLogistic -I 0 -M 500 -H 50 -W 0.0\" -B \"lazy.IBk -K 100 -W 0 -A \\\"weka.core.neighboursearch.LinearNNSearch -A \\\\\\\"weka.core.EuclideanDistance -R first-last\\\\\\\"\\\"\"' 5134738557155845452
(4)	meta.MultiClassClassifier '-M 0 -R 2.0 -S 1 -W functions.Logistic – -R 1.0E-8 -M -1 -num-decimal-places 4' -3879602011542849141
(5)	meta.Bagging '-P 100 -S 1 -num-slots 1 -I 10 -W trees.RandomForest – -P 100 -I 100 -num-slots 1 -K 0 -M 1.0 -V 0.001 -S 1' -115879962237199703
(6)	meta.AdaBoostM1 '-P 100 -S 1 -I 10 -W trees.RandomForest – -P 100 -I 100 -num-slots 1 -K 0 -M 1.0 -V 0.001 -S 1' -1178107808933117974

Table 14: Attribute selection aliphatic tails accuracy

Dataset	(1)	(2)
R-data-frame	99.47	99.54

○, ● statistically significant improvement or degradation

Table 15: Attribute selection aliphatic tails accuracy (Key)

(1)	trees.RandomForest '-P 100 -I 100 -num-slots 1 -K 0 -M 1.0 -V 0.001 -S 1' 1116839470751428698
(2)	meta.AttributeSelectedClassifier '-E \"WrapperSubsetEval -B trees.RandomForest -F 5 -T 0.01 -R 1 -E DEFAULT – -P 100 -I 100 -num-slots 1 -K 0 -M 1.0 -V 0.001 -S 1\" -S \"BestFirst -D 1 -N 5\" -W trees.RandomForest – -P 100 -I 100 -num-slots 1 -K 0 -M 1.0 -V 0.001 -S 1' -1151805453487947577

Figure 7 shows the ROC curve for the RandomForest algorithm for every possible label in the aliphatic tails prediction. Looking at the ROC curves, the RandomForest algorithm seems to preform extremely well seeing as the ROC curves are all basically at right angles. Another metric to measure how well the algorithm preforms is the area under the curve (AUC). All of the ROC curves plotted in table 7 have an AUC of over 0.99, which is extremely good. From these results it seems like the RandomForest algorithm is suitable for the task of predicting the aliphatic tails present in a membrane when given the membrane's characteristics.

4.3.3 Sterol concentration

It seems like the LMT algorithm is the most accurate in predicting sterol concentration when looking at table 16. The keys for this experiment can be found in 17. The LMT algorithm has a very high accuracy of 99.94%, which is significantly higher than any of the other algorithms. It seems unlikely that ensemble learning will improve on the LMT algorithm.

Table 16: Sterol concentration prediction accuracy

Dataset	(9)	(1)	(2)	(3)	(4)	(5)	(6)	(7)	(8)
R-data-frame	99.94	37.81 ●	82.28 ●	82.39 ●	77.83 ●	81.03 ●	99.47 ●	79.71 ●	99.42 ●

○, ● statistically significant improvement or degradation

Table 18 shows that ensamble learning does not improve accuracy in predicting sterol concentration. All of the ensemble learners aren't significantly worse or better, except for adaboost. Adaboost is the only algorithm that preforms significanlty worse. The keys belonging to the experiment shown in table 18 can be found in table 19.

Table 17: Sterol concentration prediction accuracy (Key)

(1)	rules.ZeroR ” 48055541465867954
(2)	rules.OneR ’-B 50’ -3459427003147861443
(3)	lazy.IBk ’-K 100 -W 0 -A \"weka.core.neighboursearch.LinearNNSearch -A \\\"weka.core.EuclideanDistance -R first-last\\\"\"’ -3080186098777067172
(4)	bayes.NaiveBayes ” 5995231201785697655
(5)	trees.J48 ’-C 0.25 -M 200’ -217733168393644444
(6)	trees.RandomForest ’-P 100 -I 100 -num-slots 1 -K 0 -M 1.0 -V 0.001 -S 1’ 1116839470751428698
(7)	functions.SMO ’-C 1.0 -L 0.001 -P 1.0E-12 -N 0 -V -1 -W 1 -K \"functions.supportVector.PolyKernel -E 1.0 -C 250007\" -calibrator \"functions.Logistic -R 1.0E-8 -M -1 -num-decimal-places 4\"’ -6585883636378691736
(8)	functions.SimpleLogistic ’-I 0 -M 500 -H 50 -W 0.0’ 7397710626304705059
(9)	trees.LMT ’-I -1 -M 15 -W 0.0’ -1113212459618104943

Table 18: Ensemble learning sterol concentration prediction accuracy

Dataset	(1)	(2)	(3)	(4)	(5)	(6)
R-data-frame	99.73	99.64	99.80	99.70	99.72	36.69 ●

○, ● statistically significant improvement or degradation

Table 19: Ensemble learning sterol concentration prediction accuracy (Key)

(1)	trees.LMT ’-I -1 -M 15 -W 0.0’ -1113212459618104943
(2)	trees.RandomForest ’-P 100 -I 100 -num-slots 1 -K 0 -M 1.0 -V 0.001 -S 1’ 1116839470751428698
(3)	meta.Vote ’-S 1 -B \"trees.RandomForest -P 100 -I 100 -num-slots 1 -K 0 -M 1.0 -V 0.001 -S 1\" -B \"trees.LMT -I -1 -M 15 -W 0.0\" -B \"functions.SMO -C 1.0 -L 0.001 -P 1.0E-12 -N 0 -V -1 -W 1 -K \\\"functions.supportVector.PolyKernel -E 1.0 -C 250007\\\" -calibrator \\\"functions.Logistic -R 1.0E-8 -M -1 -num-decimal-places 4\\\"\" -R AVG’ -637891196294399624
(4)	meta.Stacking ’-X 10 -M \"trees.J48 -C 0.25 -M 2\" -S 1 -num-slots 1 -B \"trees.RandomForest -P 100 -I 100 -num-slots 1 -K 0 -M 1.0 -V 0.001 -S 1\" -B \"trees.LMT -I -1 -M 15 -W 0.0\" -B \"functions.SMO -C 1.0 -L 0.001 -P 1.0E-12 -N 0 -V -1 -W 1 -K \\\"functions.supportVector.PolyKernel -E 1.0 -C 250007\\\" -calibrator \\\"functions.Logistic -R 1.0E-8 -M -1 -num-decimal-places 4\\\"\"’ 5134738557155845452
(5)	meta.Bagging ’-P 100 -S 1 -num-slots 1 -I 10 -W trees.LMT – -I -1 -M 15 -W 0.0’ -115879962237199703
(6)	meta.AdaBoostM1 ’-P 100 -S 1 -I 10 -W trees.DecisionStump’ -1178107808933117974

LMT seems to be the best algorithm for predicting sterol concentration. It also seems very capable of doing so. One of the most interesting observations of this algorithm is that the constructed tree isn’t very complex. The tree can be found in appendix C. The model build by the LMT algorithm could be interpreted as meaning that sterol concentration has a large and significant effect on the bending rigidity, thickness, comprehensibility and area per lipid (APL) of the membrane.

5 Discussion and Conclusion

5.1 Results summary

The goal of this research was to find out whether or not it was possible to use machine learning to predict the composition of a membrane when given the membrane characteristics. The first steps in trying to answer this question were to explore the gathered data to look for possible correlations between variables, and to clean the dataset so it could be used for machine learning experiments. During data exploration a few correlations where found that were later also found using the machine learning algorithms. We found that using machine learning we could predict certain components like the aliphatic tails and the sterol concentration with a high degree of accuracy. The average accuracy of both of these component was above 99% accuracy. We also found that it is possible to distinguish between whether or not a sterol is present in the membrane or not, but not between the specific sterols, when using machine learning. The highest accuracy in distinguishing between ergosterol and cholesterol that was achieved was 57.74% accuracy, which is quite low. This is compared to a 99.75% accuracy when only predicting whether or not a sterol is present in the membrane or not.

5.2 Discussion

5.2.1 Exploration

One of the most important things to note is that there was probably an subconscious bias in the exploration of the data. This is because most of the steps where taken while there was already a research question. Because of this the exploration was probably more focused on refining the research question then on looking for unexpected patterns.

The PCA plot was made before any of the data cleaning step. This means that the data wasn't transformed or normalized. The function used to plot the PCA plot did have the option to scale the data enabled, which should have taken care of any scaling issues. Except that this function might make use of z score scaling, in which case the results might not be totally accurate seeing as there is a slight skew in some of our variables. Z score scaling works better on data that is normally distributed, which is why it might have caused problems. In testing, even after taking this into account and trying to reduce the skew by using log and square transformations, we were still left with a slight skew. Even reducing the skew, the scree plot still showed around the same amount of variance explained by each dimension and PCA plot still showed around the same amount of correlation.

An important thing to address is the fact that outliers in the dataset aren't removed. This is for the sole reason that there is no justification to do so, there also wasn't much information given on how the data was gathered. Seeing as the data was most likely gathered from a controlled experiment, it should be assumed that these data points aren't incorrect inputs or caused by environmental errors, but actual correct measurement. This means that these outliers are actual possibilities that could be found in nature, which is why they shouldn't be removed. Some of the outliers could also be a result of a bias in the making of the samples. There were also no obvious erroneous instances that needed to be removed.

5.2.2 Data cleaning

The only data cleaning that was done was the removal of columns and min max normalization. The reason behind the removal of these columns was that these columns where technically other labels that could have been used in machine learning. Some of them could have been left in seeing as they could be interesting in future research, but this would just clutter up the dataset. Even though some of the variables seem to be highly correlated, it was decided not to remove some of them. This is mostly because they could still hold valuable information seeing as they aren't one hundred percent correlated. Another reasons not to remove variables with a high degree of correlation was because of the possible issues with the PCA plot. These variables could still be removed in the future to check if this further improves the machine learning algorithm. The biggest reason why they weren't removed was because the PCA plot is based on the assumption that the correlation between variables is linear. In some comparisons variables will indeed have a strong correlation, but this might change when another dimension like the sterol type is added. An example of this can be found in the results chapter in figure 3. When trying to fit a linear regression line through all of the data point, you would probably get a pretty strong correlation. But once colored on sterol type, it becomes very clear that the correlation isn't linear but inverse logarithmic.

In the cleaning of the data, min max normalization was used. This leads to a suppression of the effect of outliers. To combat this, z score normalization could have been used. The only problem with this is that z score normalization is based on a normal distribution, while most of the data in the dataset has a moderate to heavy skew. Something to combat this would be using transformation. It was however decided not to make use of this, seeing as, in testing, this lessens the skew in most data, but not by a big margin. The choice not to transform the data could lead to problems in the future if we where to use an algorithm that assumes the data has a Gaussian distribution. However, as previously stated, even transforming the data didn't reduce skewness by a large margin.

A few remarks concerning the data quality. The quality of the data seems to be good. There are enough instances, more is usually better, but in this case it probably isn't a necessity. Like mentioned before, the data does seem to be skewed, but this doesn't have to be a big problem. It is a bit worrying to see such high correlation between certain variables, but this may also be expected seeing as the subject of the data is a biological membrane. This is probably also the reason the dataset contains a few outliers.

5.2.3 Machine learning

5.2.3.1 General discussion of machine learning The machine learning experiments could have been done a lot better. This is mostly because there wasn't any kind of parameter optimization that was done. This was mostly caused by problems with Weka. Nevertheless, this could still have lead to improvements in the models. Another major subject of discussion would be the attribute selection. Attribute selection was done but not to it's fullest extend, this was due to time restriction. This was also due to the fact that the results would only have mattered if the accuracy saw a significant improvement, which didn't seem likely with an accuracy of $> 99\%$. Another problem attribute selection would cause would be that implementing multiple models in java using Weka becomes a lot harder when attribute selection is used. This is due to the way the Weka dependency is set up. It also probably wouldn't have mattered seeing as due to the fact that the java wrapper uses three models, the chance that the program would still need all of the attributes as input would be quite high. This is due to the fact that when prediction multiple membrane components, you predict multiple components that have different effects on different characteristics of the membrane.

5.2.3.2 Sterol prediction Here our findings suggest that it quite hard for machine learning to discern between ergosterol and cholesterol. This would imply that there isn't a major difference between the effect of cholesterol compared to ergosterol on a membrane's properties. When trying to look for this subject in literature, there didn't seem to be many sources. This paper (Hung et al. 2016) states that "Our findings do not support the notion that different sterols have a universal behavior that differs only in degree." this is contradictory to the results found in the Sterol type section. This is because the results in this section support the theory that sterols have an almost universal effect on membrane characteristics. The subject of the paper mentioned is on the condensing effect of ergosterol and cholesterol. The paper also states that due to ergosterol's and cholesterol's condensing effect they are considered important regulators of membrane thickness. Another paper (Czub and Baginski 2006) also does a comparison of membranes containing cholesterol and ergosterol, but doesn't mention anything about the properties that we used for our research.

5.3 Conclusion & Future research

To conclude, based on the findings of the machine learning section, we can say that it is possible to predict membrane composition when given membrane characteristics, using machine leaning. This statement is only true to a certain extent. It is possible to predict certain component like the aliphatic tails and the sterol concentration with a high degree of accuracy. This can be said because we found that the accuracy in these predictions is over 99%. It is also possible to predict whether or not a sterol is present in the membrane, but not which sterol this might be. Future research should focus on trying to build a model that could possibly discern between different types of sterols. It might be possible to do something like this with a neural network. A neural network could also be used to try and improve the models that where made here for the aliphatic tails and the sterol concentration. Another thing future research could focus on would be trying to make the sterol concentration prediction a regression problem instead of a classification problem. This would mean you could predict the sterol concentration exactly instead of having an approximation.

6 Project proposal for minor

An inserting thing to do to expand upon the research done here would be to create a user friendly web application. This could be done in the Application Design (AD) minor. It would be especially important to make it usable both on mobile devices as on desktops. This means a web application would probably suffice. The fished product should be able to use the machine learning algorithms to predict components of a membrane with the specified characteristics. The input should be either a file or just simply all of the characteristics and their corresponding value. The output should be an easily readable and good looking display of the predicted components. It could also display the confidence that the predicted label is the actual correct label. The target audience for the application would be scientists/laborants. They could use it to try and make membranes with specific properties.

It could also be interesting to try and improve the current models with neural networks, especially the model for predicting the sterol. Maybe a neural network could actually discern between cholesterol and ergosterol whereas our current models can't. This could be an interesting project for the HighThroughput High Performance Biocomputing (HTHPB) minor.

7 Refereces

Czub, Jacek, and Maciej Baginski. 2006. “Comparative Molecular Dynamics Study of Lipid Membranes Containing Cholesterol and Ergosterol.” *Biophysical Journal* 90 (7): 2368–82.

Hung, Wei-Chin, Ming-Tao Lee, Hsien Chung, Yi-Ting Sun, Hsiung Chen, Nicholas E Charron, and Huey W Huang. 2016. “Comparative Study of the Condensing Effects of Ergosterol and Cholesterol.” *Biophysical Journal* 110 (9): 2026–33.

Landwehr, Niels, Mark Hall, and Eibe Frank. 2005. “Logistic Model Trees.” *Machine Learning* 59 (1-2): 161–205.

R Core Team. 2020. *R: A Language and Environment for Statistical Computing.* Vienna, Austria: R Foundation for Statistical Computing. https://www.R-project.org/.

Waikato, University of. 2020. *Weka 3: Machine Learning Software in Java.* Hamilton, New Zealand: University of Waikato. https://www.cs.waikato.ac.nz/ml/weka/.

8 Apendix

8.1 A, The code

```
# Libraries
library("ggplot2")
library("kableExtra")
library("factoextra")
library("cowplot")
library("gridExtra")
library("dplyr")
library("scales")
library(ggpubr)
library(ggfortify)
library(tidyr)
library(purrr)
library(ggcorrplot)
library(reshape2)
library(e1071)
library(Hmisc)
library(forcats)

# Read the data and transform it to a tibble
data <- read.csv("data/dataFrame_all_sims.csv",
                 sep = ",", header=TRUE,
                 stringsAsFactors=FALSE)
data <- as_tibble(data)

# Remove NA's
nona_data <- na.omit(data)
# Scaling
scale_min_max <- function(x) (x - min(x)) / (max(x) - min(x))
nona_copy <- nona_data[c("APL", "thickness", "bending",
                                                "tilt", "zorder","compress")]

# Min max normalized data
scaled_data <- as.data.frame(lapply(nona_copy, FUN=scale_min_max))
scaled_data$sterol.type <- nona_data$sterol.type

# Dataframe used for PCA plot
correlation <- nona_data[,9:14]

codebook <- read.csv("data/codebook.csv", sep = ",", header=TRUE)
# Get the description by the abbreviation from the codebook
get_des_by_ab <- function(df, abbreviation){
  description <- df[df$Abbreviation == abbreviation,]$Desciption
  return(description)
}

# Get the five number summary of the selected columns
sum <- summary(data[c("temperature","sterol.conc","satur.index",
                      "PC.conc","ethanol.conc","APL", "thickness", "bending",
                      "tilt", "zorder","compress")])
```

```
sum <- sub(".*:", "", sum)
sum[is.na(sum)] <- 0
rownames(sum) <- c("Minimum", "Q1", "Median", "Mean", "Q3", "Maximum", "Number of NA's")

# Print five number summary using kable
kable(sum, caption="Five number summary of the original dataset") %>%
  kable_styling(latex_options = c("scale_down", "hold_position"))

# Calculate the skew on every column
results <- data.frame("Skew" = apply(nona_data[c("temperature","sterol.conc","satur.index",
                                                 "PC.conc","ethanol.conc","APL", "thickness", "bending"
                                                 "tilt", "zorder","compress")], 2, skewness, na.rm =TRUE

# Print results using kable
kable(t(results), caption="Skew in numeric columns") %>%
  kable_styling(latex_options = c("scale_down", "hold_position"))

# Make a boxplot of the numeric columns
plot1 <- nona_data %>%
  select("temperature","sterol.conc",
                      "PC.conc","ethanol.conc","bending",
                      "tilt","compress") %>%
  pivot_longer(., cols = c("temperature","sterol.conc",
                      "PC.conc","ethanol.conc","bending",
                      "tilt","compress"), names_to = "Variable", values_to = "Value") %>%
  ggplot(aes(x = Variable, y = Value)) +
  geom_boxplot() + theme(axis.text.x = element_text(angle = 90, vjust = 0.5, hjust=1))

plot2 <- nona_data %>%
  select("satur.index","APL","thickness","zorder") %>%
  pivot_longer(., cols = c("satur.index","APL","thickness","zorder"), names_to = "Variable", values_to =
  ggplot(aes(x = Variable, y = Value)) +
  geom_boxplot() + theme(axis.text.x = element_text(angle = 90, vjust = 0.5, hjust=1))

plot3 <- nona_data %>%
  select("APL") %>%
  pivot_longer(., cols = c("APL"), names_to = "Variable", values_to = "Value") %>%
  ggplot(aes(x = Variable, y = Value)) +
  geom_boxplot() + theme(axis.text.x = element_text(angle = 90, vjust = 0.5, hjust=1))

ggarrange(plot1, plot2, plot3,
          labels = c("A", "B", "C"),
          ncol = 3, nrow = 1)

# Create density plots
p <- ggplot(nona_data, aes(x=bending, colour = factor(sterol.type))) +
  geom_density() +
  xlab(get_des_by_ab(codebook, "bending")) +
  ylab("Density") +
  ggtitle("Bending rigidity density plot") +
  labs(colour = get_des_by_ab(codebook, "sterol.type"))

p2 <- ggplot(nona_data, aes(x=APL, colour = factor(sterol.type))) +
  geom_density() +
  xlab(get_des_by_ab(codebook, "APL")) +
  ylab("Density") +
  ggtitle("Area per lipid density plot") +
  labs(colour = get_des_by_ab(codebook, "sterol.type"))

p3 <- ggplot(nona_data, aes(x=compress, colour = factor(sterol.type))) +
  geom_density() +
  xlab(get_des_by_ab(codebook, "compress")) +
  ylab("Density") +
  ggtitle("Compressibility density plot") +
  labs(colour = get_des_by_ab(codebook, "sterol.type"))

p4 <- ggplot(nona_data, aes(x=APL, colour = factor(tails))) +
  geom_density() +
  xlab(get_des_by_ab(codebook, "APL")) +
  ylab("Density") +
  ggtitle("APL density") +
  labs(colour = get_des_by_ab(codebook, "tails"))
# Plot density plots
plots <- ggarrange(p, p2, p3, p4,
                   labels = c("A", "B", "C", "D"),
                   ncol = 2, nrow = 2)
plots

#Plot APL against bending rigidity
chart <- ggplot(nona_data, aes(APL, bending, colour = factor(sterol.type) )) +
  geom_point(alpha=1/20, size=0.5) +
  xlab(get_des_by_ab(codebook, "APL")) +
  ylab(get_des_by_ab(codebook, "bending")) +
  theme(axis.title.y = element_text(size=8)) +
  ggtitle("Bending rigidity as a function of APL") +
  scale_color_manual(name=get_des_by_ab(codebook, "sterol.type"),
                     labels = c("Cholesterol",
                                "Ergosterol",
                                "No sterol"),
                     values = hue_pal()(3)) +
  guides(colour = guide_legend(override.aes = list(alpha = 1)))

plots <- ggarrange(chart,
                   chart +
                     geom_smooth(method = "loess") + ggtitle("Bending rigidity as a function of APL, inc
                   labels = c("A", "B"),
                   ncol = 1, nrow = 2)

plots

# Create plot
chart <- ggplot(nona_data, aes(compress, bending, colour = factor(sterol.type) )) +
  geom_point(alpha=2/10, size=0.5) +
  xlab(get_des_by_ab(codebook, "compress")) +
  ylab(get_des_by_ab(codebook, "bending")) +
  theme(axis.title.y = element_text(size=8)) +
  ggtitle("Compressibility against bending rigidity,
          coloured on sterol type") +
  scale_color_manual(name=get_des_by_ab(codebook, "sterol.type"),
                     labels = c("Cholesterol",
                                "Ergosterol",
                                "No sterol"),
                     values = hue_pal()(3)) +
  theme(plot.title = element_text(size=10)) +
  guides(colour = guide_legend(override.aes = list(alpha = 1)))

chart2 <- ggplot(nona_data, aes(compress, bending, colour = factor(sterol.type) )) +
  geom_point(alpha=1/10, size=0.5) +
  xlab(get_des_by_ab(codebook, "compress")) +
  ylab(get_des_by_ab(codebook, "bending")) +
  theme(axis.title.y = element_text(size=8)) +
  ggtitle("Compressibility against bending rigidity,
          coloured on sterol type, zoomed in") +
  scale_color_manual(name=get_des_by_ab(codebook, "sterol.type"),
                     labels = c("Cholesterol",
                                "Ergosterol",
                                "No sterol"),
                     values = hue_pal()(3)) +
  scale_x_continuous(limits=c(0,120)) +
  scale_y_continuous(limits=c(0, 45))  +
  theme(plot.title = element_text(size=10)) +
  guides(colour = guide_legend(override.aes = list(alpha = 1)))

# Arrange plots
plots <- ggarrange(chart,
                   chart2,
                   labels = c("A", "B"),
                   ncol = 1, nrow = 2)

plots

# Create plots
chart2 <- ggplot(nona_data, aes(APL, bending, colour = factor(satur.index), shape = factor(sterol.type)
  geom_jitter(alpha=0.1, size=0.5) +
  xlab(get_des_by_ab(codebook, "APL")) +
  ylab(get_des_by_ab(codebook, "bending"))  +
  ggtitle("APL against bending regidity,
          coloured on saturation index, shaped on sterol type") +
  theme(plot.title = element_text(size=10)) +
  guides(colour = guide_legend(override.aes = list(alpha = 1)))
chart <- chart2 + labs(colour = get_des_by_ab(codebook, "satur.index"), shape = get_des_by_ab(codebook,

# Arrange plots
plots <- ggarrange(chart,
                   labels = c("A"),
                   ncol = 1)

plots
```

```r
# Create PCA plot
res.pca <- prcomp(correlation, scale = TRUE)
plot1 <- fviz_eig(res.pca) +
  theme(axis.title.y = element_text(size=6))
plot2 <- fviz_pca_var(res.pca,
                      col.var = "contrib", # Colour by contributions to the PC
                      gradient.cols = c("orange", "purple"),
                      repel = TRUE     # Avoid text overlapping
                      )
plot3 <- fviz_pca_biplot(res.pca, repel = FALSE,
                         col.var = "purple", # Variables colour
                         col.ind = nona_data$sterol.type,  # Individuals colour
                         label = FALSE,
                         addEllipses =  TRUE,
                         ellipse.type = "t"
                         )
# Arrange plots
plot <- arrangeGrob(plot1,  plot2,
                    plot3,
                    ncol = 2, nrow = 2,
                    layout_matrix = rbind(c(1,2), c(3,3)))

plot <- as_ggplot(plot) +
  draw_plot_label(label = c("A", "B", "C"), size = 15,
                  x = c(0, 0.5, 0), y = c(1, 1, 0.5))

plot

kable(head(data), caption="Top of original data") %>%
  kable_styling(latex_options = c("scale_down", "hold_position"))

kable(head(scaled_data), caption="Top of cleaned data") %>%
  kable_styling(latex_options = c("hold_position"), full_width = TRUE)

# Get the five number summary of the selected columns
sum <- summary(scaled_data[1:5])

sum <- sub(".*:", "", sum)
sum[is.na(sum)] <- 0
rownames(sum) <- c("Minimum", "Q1", "Median", "Mean", "Q3", "Maximum")

# Print five number summary using kable
kable(sum, caption="Five number summary of the cleaned dataset") %>%
  kable_styling(latex_options = c("hold_position"), full_width = TRUE)

AUC_calculator <- function(TPR, FPR){
  dFPR <- c(diff(FPR), 0)
  dTPR <- c(diff(TPR), 0)
  abs(sum(TPR * dFPR) + sum(dTPR * dFPR)/2)
}

ROC_variable_maker <- function(filehandle){
  open_file <- RWeka::read.arff(filehandle)
  AUC <- with(open_file, AUC_calculator(`True Positive Rate`, `False Positive Rate`))
  plot <- ggplot(open_file)+
    geom_path(aes(x=`False Positive Rate`,
                  y=`True Positive Rate`,
                  color=Threshold)) +
    scale_color_gradient2(low="orange", mid="mediumvioletred",
                          high="purple", midpoint = 0.5) +
    labs(title= paste("ROC curve, AUC: ", round(AUC, 6)),
         x = "False Positive Rate (1-Specificity)",
         y = "True Positive Rate (Sensitivity)") +
    theme(plot.title = element_text(size=10),
          axis.title.x=element_text(size=9),
          axis.title.y=element_text(size=7),
          legend.title = element_text(size = 8),
          legend.text = element_text(size = 6))
  return(plot)

}

ROC_creator <- function (filefolder){
  files <- list.files(path=filefolder, pattern="*.arff", full.names=TRUE, recursive=FALSE)
  plots <- lapply(files, ROC_variable_maker)

  n <- length(plots)
  nCols <- floor(sqrt(n))
  do.call("grid.arrange", c(plots, ncol= nCols))
}

ROC_creator("./Experiment/Results_Explorer/ROC_Curve")
```

8.2 B, The codebook

```
codebook <- read.csv("data/codebook.csv", sep = ",", header=TRUE)
# Turn dataframe into a tibble and print
codebook <- as_tibble(codebook)
kable(codebook, caption = "Codebook") %>%
  kable_styling(latex_options = c("scale_down", "hold_position"))
```

8.3 C, LMT algorithm

```
=== Classifier model (full training set) ===

Logistic model tree

bending <= 0.054892: LM_1:116/232 (215)
bending > 0.054892
| thickness <= 0.894825
| | compress <= 0.089242
| | | APL <= 0.578692: LM_2:116/580 (894)
| | | APL > 0.578692: LM_3:116/580 (1029)
```

Table 20: Codebook

Abbreviation	Type	Class	Desciption	Units
temperature	integer	integer	Temperature (Kelvin)	Kelvin
sterol.type	character	character	Sterol type	
sterol.conc	integer	integer	Sterol concentration (%)	%
other.phosph	character	character	Other (phospho)lipids in membrane (headgroup)	
tails	character	character	Aliphatic tails	
satur.index	double	numeric	Saturation index (double bonds per tail)	
PC.conc	integer	integer	Phosphatidyl choline concentration (% of non-sterol lipids)	
ethanol.conc	integer	integer	Ethanol concentration (% of solvent)	
APL	double	numeric	Area per lipid (nm^2)	nm^2
thickness	double	numeric	Thickness (nm)	nm
bending	double	numeric	Bending rigidity (kB T)	kB T
tilt	double	numeric	Tilt angle (degrees)	degrees
zorder	double	numeric	Z-order	
compress	double	numeric	Compressibility (cN / m)	cN / m

```
| | compress > 0.089242: LM_4:116/464 (384)
| thickness > 0.894825: LM_5:116/348 (168)

Number of Leaves : 5

Size of the Tree : 9
```