Project 1

02450 Introduction to Machine Learning and Data Mining

05-10-2021

Group 217:

Alex Abades | s212784

Tommy Vu | s163059

Table of Contents

Table of Figures

Table of Tables

Table of contribution:

Student/Section	Sec. 1	Sec. 2	Sec. 3	Sec. 4	Sec. 5	Sec. 6
S212784	40%	60%	40%	60%	40%	60%
s163059	60%	40%	60%	40%	60%	40%

All group members have contributed evenly to this report. Analysis, use of models and solutions have been discussed in plenum.

1 Description of dataset

The dataset consists of 336 E.coli proteins and is obtained from UCI Machine Learning (Insert reference). The dataset has 8 attributes (7 predictive, 1 name) and each instance belongs to one of 8 classes. The overall problem of interest is to classify the proteins into cellular localization sites. The 7 predictive attributes can be seen below:

- **meg:** A modification of McGeoch's (MeGeoch 1985) signal sequence detection parameter.
- **lip:** The presence or absence of the consensus sequence (yon Heijne 1989) for Signal Peptidase II.
- **gvh:** The output of a weight matrix method for detecting cleavable signal sequences (yon Heijne 1986).
- **alam:** The output of the ALOMpr ogram (Klein, Kanehisa,& DeLisi 1985) for identifying membrane spanning regions on the whole sequence.
- **aim2:** On the sequence excluding the region predicted to be a cleavable signal sequence by yon Heijne's method (yon IIeijnc 1986).
- **chg:** The presence of charge on the N-terminus of predicted mature lipoproteins.
- **aac:** The result of discriminant analysis on the amino acid content of outer membrane and periplasmic proteins.
-

All the attributes above are a measurement for whether or not the protein belongs to the cellular localization site.

The 8 classes aforementioned are seen below:

- **imL:** Inner membrane lipoproteins
- **omL:** Outer membrane lipoproteins
- **imS:** Inner membrane proteins with cleavable signal sequence
- **om:** Other outer membrane proteins
- **pp:** Periplasmic proteins
- **imU:** Inner brane proteins with an uncleavable signal sequence
- **im:** Inner membrane proteins without a signal sequence
- **cp:** Cytoplasmic proteins
-

These are the 8 classes it is desired to assign the different proteins into.

The overall problem of interest is to predict the cellular localization of the proteins based on the attributes from the dataset. Since it is desired to classify the different proteins into the cellular localization, the main focus will be classification. However, to accomplish the goal some data processing is required. One way of allowing classification of the different classes is by using regression models. This could for example be a multinomial logistic regression that is usually used to classify subjects based on values of a set of predictor variables. For now the data will not be transformed since all continuous attributes are in the same range.

2 Summary of previous analysis of the data

A Probabilistic Classification System for Predicting the Cellular Localization Sites of Proteins, Paul Horton & Kenta Nakai.

The aim of the analysis made by Paul H. and Kenta N. was to classify proteins into their various cellular localization sites based on their amino acid sequences. Previous analysis with the same aim have been applied to the data set, such analysis had a common problem, they weren't well suited for reasoning under uncertainty.
In this analysis, Paul H. and Kenta N. describe a model for classification that can be seen either as a decision trees or a restricted form of Bayesian Network to deal with the uncertainty. Thanks to that model they have obtained, from 336 E. coli proteins, 8 classes with an accuracy of 81%.
There are also two analysis (Nakai & Kanhesia 1991) and (Nakai & Kanhesia 1991) that define the localization of a protein within a cell given its amino acid sequences, in gram negative bacteria and in eukaryotic cells respectively. However, there is no direct probabilistic interpretation of their certainty factors and for optimal prediction accuracy they must be hand-turned for each dataset.

In this paper they builded a probabilistic reasoning classifier system, which classifies objects on the basis of an input vector of real valued feature variables for each object. This relationship is provided by a human expert.

The model consists of a rooted binary tree of classification variables (x) and featured variables (y). Each leaf of the tree represents possible classes of the protein. The nodes represent the classes associated within the leaves that are branches from that specific node, which in addition, has associated a probability which specifies the probability that a given label of a given protein is in that specific node. That probability follows the sum rule. Therefore, if the probability that that node contains the label of a specific protein is true, the sum of all the probability leaves must be 1.
Furthermore, the model also follows the conditional independence, which allows the model to calculate the probability of each node given a set of values of the feature variables.

They have followed three different approaches, each one with defined continuous features. The three approaches have one thing in common, the first step. In this first step they normalize the data $X \in [0,1]$.
Afterwards, methods 1 and 2 used discretization by dividing the range of X into intervals, and each interval treated as a scalar value. The problem was that they didn't have a good criteria for choosing the number of intervals. Consequently, they randomly chose the intervals to be equal as either the log to the base 2 of the number of data points, or the square root of the number of examples that belong to that node class that was being discretized.

As opposed to the other two, the third method didn't discretize the data points, instead they applied a conditional probability function that learned a sigmoid function. Gradient descent was used to minimize the values of alpha and beta (variables from the sigmoid function) given that the sigmoid function doesn't have local minimums where gradient descent could get stuck. Specify that the variables alpha and beta were different for each variable.

After creating the model, this one detected that the **lip** and **chg** features have only two values on the data set used, therefore the program treated them as discrete variables.

The classes of the classification model with its number of examples and corresponding percentage of sequences for which the correct class matched the class with the highest probability using the 3th method, sigmoid function.

- **imL:** Inner membrane lipoproteins:
  - 2 Examples.
  - 50.0% matching.
- **omL:** Outer membrane lipoproteins.
  - 5 Examples.
  - 100.0% matching.
- **imS:** Inner membrane proteins with cleavable signal sequence.
  - 2 Examples.
  - 0.0% matching.
- **om:** Other outer membrane proteins.
  - 20 Examples.
  - 65.0% matching.
- **pp:** Periplasmic proteins.
  - 52 Examples.
  - 78.9% matching.
- **imU:** Inner brane proteins with an uncleavable signal sequence.
  - 35 Examples.
  - 71.4% matching.
- **im:** Inner membrane proteins without a signal sequence.
  - 77 examples.
  - 77.9% matching.
- **cp:** Cytoplasmic proteins.
  - 143 Examples.
  - 96.5% matching.

Below, the accuracy of classifying the E. coli protein for each method is shown. The sigmoid function (3th method) is the most accurate of all three. The accuracy is computed with all data.

- Log to base 2 method (1st):
  - Accuracy of 79.8%
- Square root (2nd):
  - Accuracy of 82.7%
- Sigmoid function (3rd):
  - Accuracy of 84.2%

To be able to contrast the accuracy results from the classifier, they performed a cross validation test on the data. They randomly split the data into 10 different equally sized subsets and trained on the remaining data. Point Out that the X-validation accuracy is the mean of all 10 subsets).

- Log to base 2 method (1st):
  - X validation test: 80.6%

- Square root (2nd):
    - X validation test: 79.1%
- Sigmoid function (3th):
    - X validation test: 81.1%

Overall, it can be said to be a successful project with an accuracy of 81%. The paper showed different methods and ways of processing the e.coli dataset.

3 Description of attributes in dataset

Besides the name, all other attributes in the dataset are some sort of score explaining how likely it is that the protein belongs to the cellular localization site. The attributes appear to be normalized, since they are all within the interval 0-1. To get an even better overview of the dataset, some basic statistic operations have been done on the dataset. The result of these is seen on figure 1:

sumary	mgc	gvh	lip	chg	aac	alm1	alm2
"count"	336	336	336	336	336	336	336
"mu"	0.50006	0.5	0.49548	0.50149	0.50003	0.50018	0.49973
"median"	0.5	0.47	0.48	0.5	0.495	0.455	0.43
"std_dev"	0.19463	0.14816	0.088495	0.027277	0.12238	0.21575	0.20941
"max"	0.89	1	1	1	0.88	1	0.99
"min"	0	0.16	0.48	0.5	0	0.03	0
"25%"	0.34	0.4	0.48	0.5	0.42	0.33	0.35
"50%"	0.5	0.47	0.48	0.5	0.495	0.455	0.43
"75%"	0.665	0.57	0.48	0.5	0.57	0.71	0.71

Table 1. Basic statistics applied to Ecoli-dataset

On Table 1 the different attributes are analysed. An interesting thing to notice is that all the attributes have a mean that is close to 0.5, meaning that they might have been normalized. Furthermore, all values can be said to have an absolute zero. For those reasons they can be considered continuous ratio attributes. Another interesting thing to look at in figure 1 is the standard deviation (std_dev). Higher standard deviations indicate that the data is more spread out for the attributes. It can also be seen that the attributes *lip* and *chg* only consist of two different values (0.5/1 and 0.48/0.5) and it might be beneficial to change these attributes into binary attributes later on.

Class	imL	omL	imS	om	pp	imU	im	cp
Counts	2	5	2	20	52	35	77	143
Percentage (%)	0.595	1.488	0.595	5.952	15.476	10.417	22.917	42.560

Table 2. Category summary

From Table 2, it becomes clear that cp is the dominant class for this dataset with 143 samples. Another thing to notice about the dataset, is the lack of samples for some of the classes. For

example, there are only 2 samples of imL/imS and their necessity for the project will be considered later on.

The last basic statistics plot that was made, was a matrix plot, containing the histograms of the classes in the diagonal plots and the scatterplots in the other plots. This is seen on figure 2:

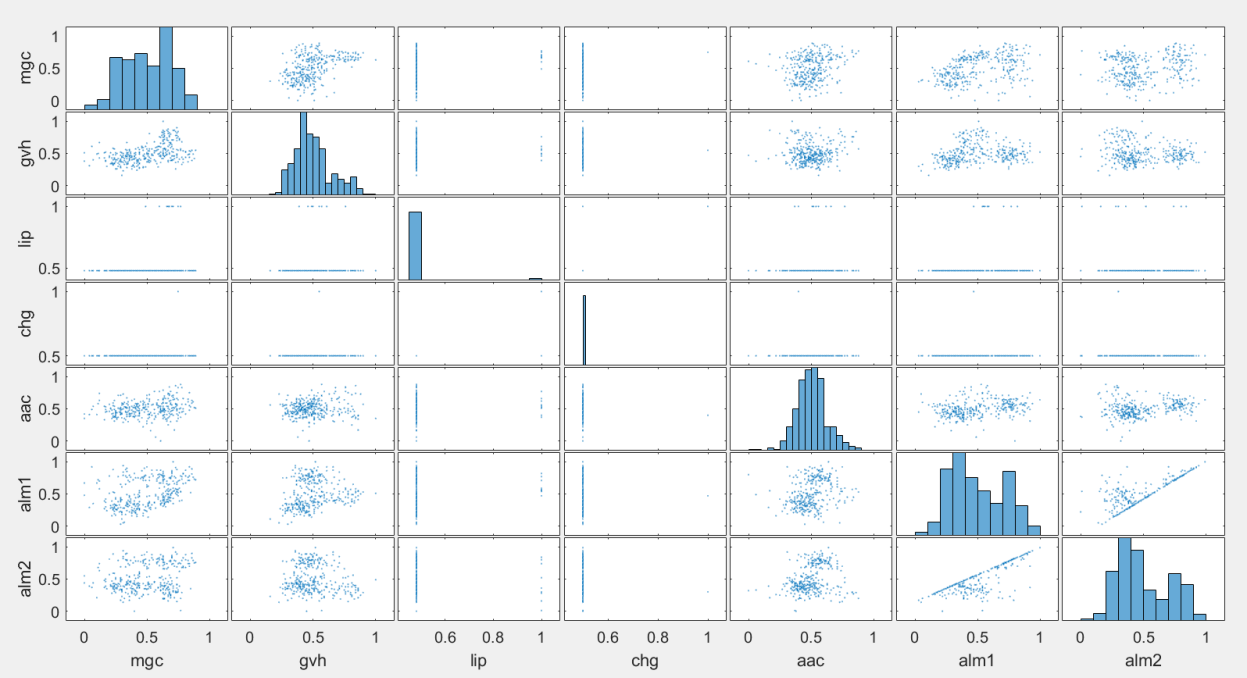

Figure 1. Plot matrix of attributes.

The histograms in Figure 1 show that the classes aac and gvh follow a Gaussian distribution. There is no valuable information to gain from the histograms of lip and chg. Looking at the last classes (mgc, alm1, alm2) it seems like they follow a bimodal distribution.

4 Data visualization / PCA

Before applying PCA we computed a heatmap after normalizing the data due to the low correlation we've seen between some attributes on the matrix plot. We've found 1 strong and 2 middle correlations, the strongest is between the attributes alm1 and alm2, with a score of 0.8093. The two remaining are between the two attributes gvh and mgc and alm1 and mgc with scores of 0.4548 and 0.397 respectively.
With these correlations, we can expect that PCA at least captures in three different eigenvectors the variance of these combined attributes.
After applying Principal component analysis by the singular value decomposition, we found that the first principal component explains almost 52% of the data while the second component explains 24% of the data. The other components are below 10% of variance.

PCA	Eigenvalues	Difference	Proportion	Cumulative
1	30.05	15.833	0.51617	0.51617
2	14.217	9.3152	0.2442	0.76037
3	4.9019	0.58588	0.084199	0.84457
4	4.316	1.4573	0.074135	0.91871
5	2.8588	1.2054	0.049104	0.96781
6	1.6533	1.4326	0.028399	0.99621
7	0.22074	0	0.0037916	1

Table 3. Eigen values of the correlation matrix.

To see a better representation of the variance explained by PCA, we've plotted the proportion of variance and the cumulative variance (columns 4 and 5 from Table 3), setting a threshold of 0.9.

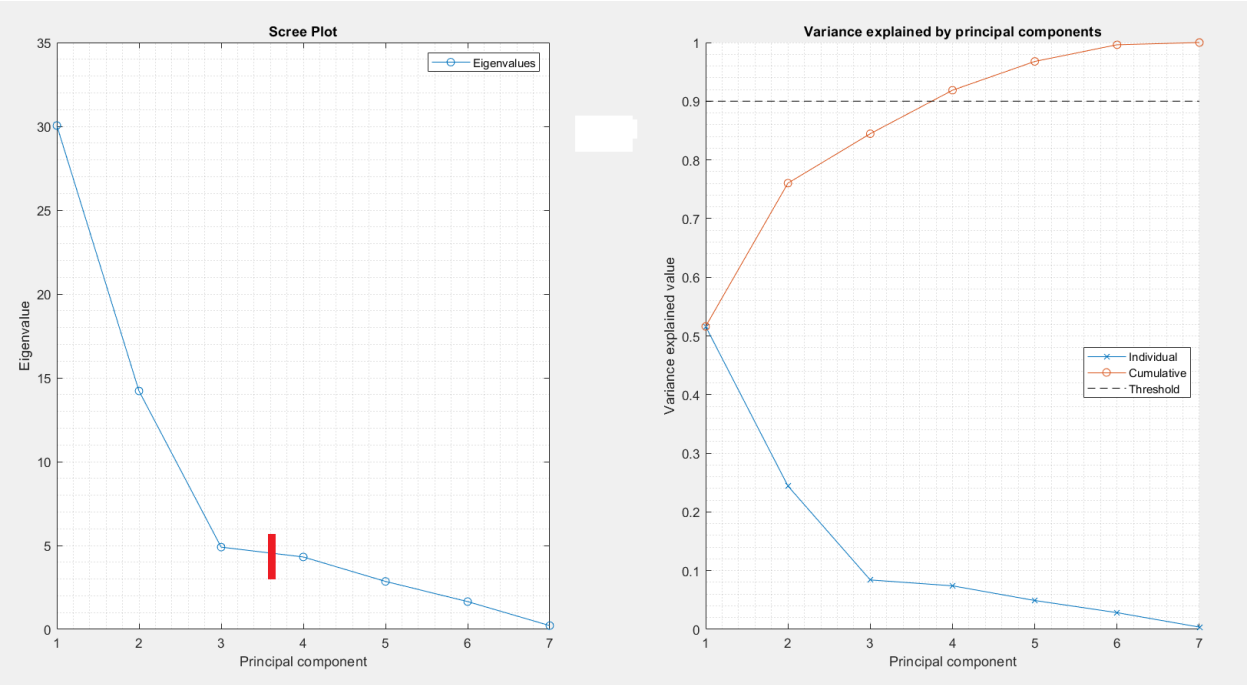

Figure 2. The Scree and Variance Explained Plots

We have discussed three different methods to select the number of principal components:

- Scree slope → First 3 PCA.
- Eigenvalues bigger 1 → First 6 PCA.
- Threshold on variance explained → First 4 PCA.

As each method gives us a different number of PCA, thus we've decided to select the number based on the scree plot, when the slope gets flat, in this way we only have 3 dimensions, and the further data visualization will be easier.

Once we have selected the number of PCA, we must interpret what is being explained.

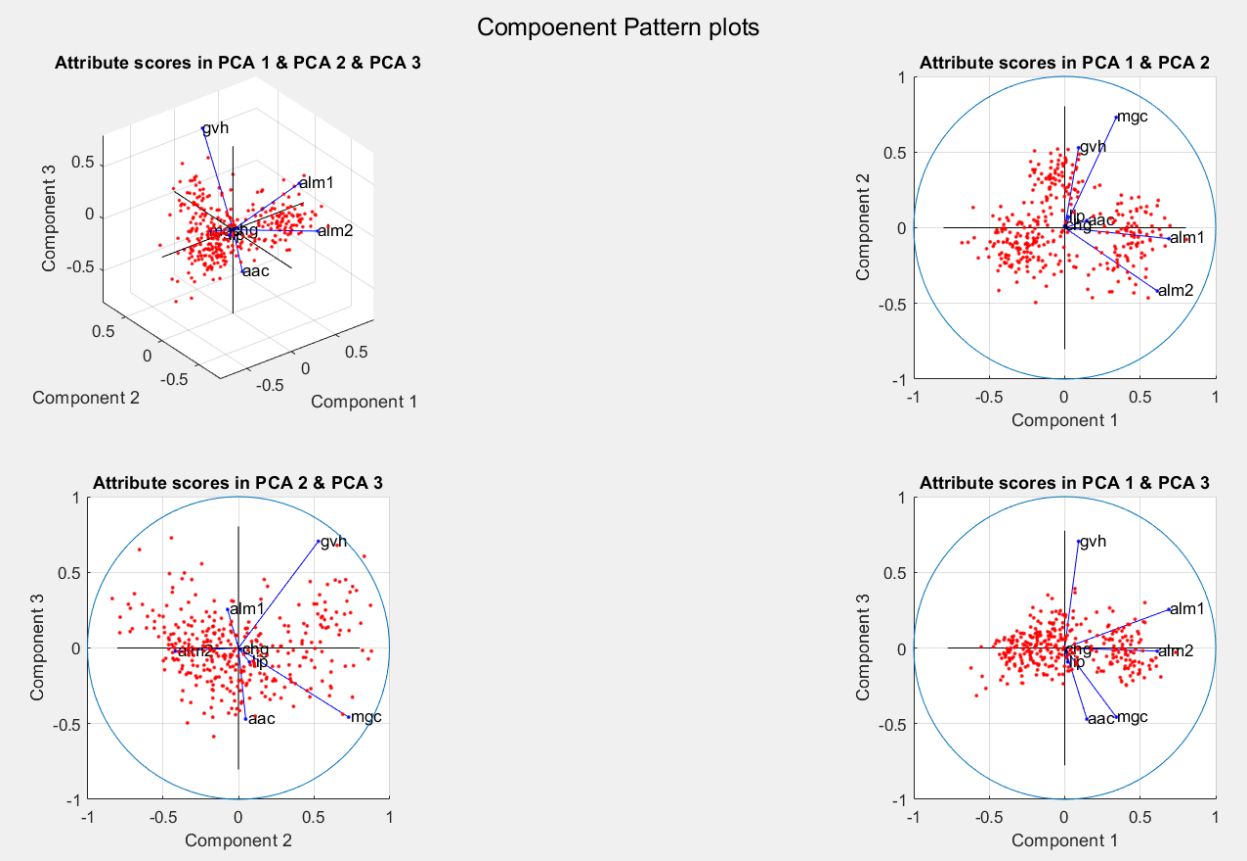

Figure 3. Component pattern plots of PCA 1, 2 & 3.

On Figure 3 we can see that the lip and chg attributes are always close to zero, it's completely understandable since both are binary attributes and low variance is captured in these variables. The first principal component seems to capture the correlation between alm1 and alm2 and also explains a little of the variance captured from mgc, while the second component captures most of the gvh and mgc correlation in contrast with the variable alm2. Finally, the third component explains the variance of gvh in contrast with the mgc and aac variables, additionally some of the variance of alm1 is also captured in this 3rd component.

Once we know which attributes are describing the principal components, we can plot the scores (the new positions of our data in our new reference system) and colour each category to see if the 3 first components do a good job separating the data.

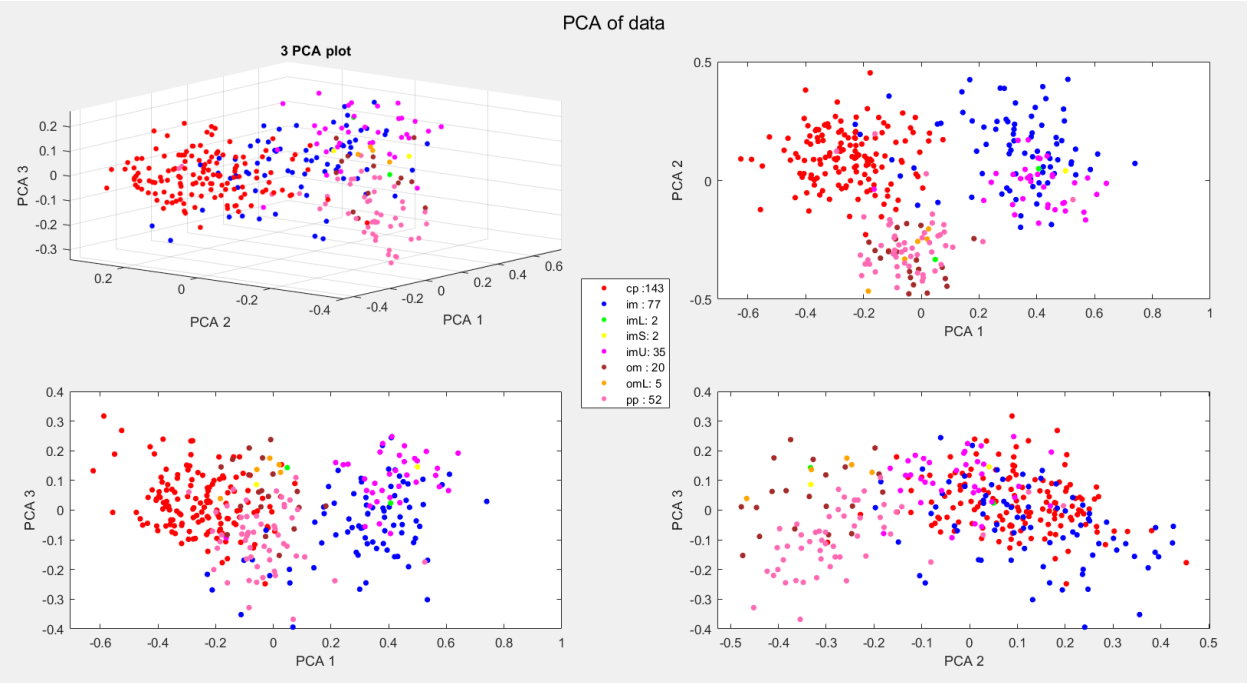

Figure 4. PCA of data.

From the plots, it can be said that the first principal component does a really good job splitting the data into three categories, cp, im and om. The second component does a good job separating the data in two groups, pp-om-omL and cp-im-imU, the other two categories. The third one tells less about the data, we could say that separates two categories, omL and pp. These first 3 components of PCA don't do a good job capturing the variance among the imL and imS categories.
We could make a further analysis with these components if instead of plotting all the scores from all categories we only plotted two or three categories each time, in that way we could clearly see how well the components are splitting these specific categories.

In addition, PCA it's useful to reduce the dimensionality of the data i.e., when dealing with high dimensional data as images or other complex datasets (as when you have to apply binarization to some attributes). PCA tries to find the true underlying variables worth for the analysis and throw away the ones that could underrate our final algorithm. In this particular case, we don't have either a high dimensional dataset or a massive amount of data collected, so we don't need PCA for dimensionality reduction but for uncover worth variables.

5 Summary

From this analysis of the dataset, the primary objective of classifying the proteins based on the different attributes seems feasible. The dataset contains 8 attributes, with one of the attributes being the name and the rest a continuous score that can be used to locate the proteins into one of the 8 classes. By doing some basic statistics on the dataset, some valuable knowledge was gained for later on, when machine learning algorithms and further analysis will be conducted. By plotting the histograms, it was found out that most of the datasets followed either a Gaussian distribution or a bimodal distribution. Furthermore, some of the classes were underrepresented in the dataset with very few samples which gives the opportunity to leave out these classes later on. Lastly a PCA was applied to the dataset. It was found that the first principal component explained 52% of the data, while the second component explained 24% of the data. The rest of the components explained under 10%. The components were then plotted onto the dataset, to see how well it separated the data. The first component could split the data into three different categories (cp, im, om). The second component could split the data into two larger groups of categories (pp-om-omL, cp-im-imU). With all these minor methods on the dataset, it can be concluded that it is possible to categorise the different proteins into their cellular localization sites. Given that the correct methods and algorithms are used.

6 Exam problems for the project (Short explanation with answer)

1. Option A/B/C/D - The correct answer is D. X2-X7 is ratio. Leaving B out. Since x1 is an interval the only option left is D.

2. Option A - Option A is the correct answer. Looking at the formulas for p-norm distance from lecture 3 slides, it is simply done by trial error. Starting from A the p-norm infinite distance is given by:
   $$d_{inf}(x,y) = max\{|x_1 - y_1|, |x_2 - y_2|, \dots, |x_M - y_M|\}$$
   Giving the two values $d_{inf}(x14, x18) = max\{|7|, |2|\} = 7$

3. Option A - Is the correct answer. The explained variance for a single component is given by $Explained\ var. = \frac{{\sigma_i}^2}{\sum_{i=1}^{M} {\sigma_i}^2}$, with the diagonals in S begins sigma. Finding the total first
   $$total = {\sigma_1}^2 + {\sigma_2}^2 + {\sigma_3}^2 + {\sigma_4}^2 + {\sigma_5}^2 = 193.21 + 155.5009 + 131.7904 + 100.6009 + 89.3025 = 670.4047$$
   Finding the variance explained by each component
   $$var1 = \frac{193.21}{670.4047} = 28.82\%, var2 = \frac{155.5009}{670.4047} = 23.20\%, var3 = \frac{131.7904}{670.4047} = 19.66\%, var4 = \frac{131.7904}{670.4047} = 15.01\%$$
   $$var5 = \frac{89.3025}{670.4047} = 13.32\%$$
   Now checking option for first four components greater than 0.8 (80%)
   $$var1 + var2 + var3 + var4 = 28.82\% + 23.20\% + 19.66\% + 15.01\% = 86.68\% = 0.8668.$$
   Meaning A is true.

4. Option A/B/C/D

5. Option C - C is the correct answer. The Jaccard similarity is given by:

$J = \frac{f_{11}}{K - f_{00}}$, where f11 is the amount of words s1 and s2 have in common and f00 is the amount of words neither of them has. The K is the amount of samples which is given as 20000. Therefore the values are: f11 = 2, f00 =0, K = 20000. Inserting in formula: $J = \frac{2}{20000 - 0} = 0.0001$. Which is option C.

6. Option A/B/C/D