Dataset exploration

A case study for a computational biology framework

T.P.A. Beishuizen (0791613)
Biomedical Engineering - Computational Biology
Computer Science - Data Mining
Eindhoven, University of Technology
Email: t.p.a.beishuizen@student.tue.nl

October 11, 2018

Contents

1 Introduction

Many biomedical datasets have been created to use for expansion of biomedical knowledge and improvement of healthcare. Biomedical data is a generalizing term that describes multiple data types[1]. Examples of biomedical data are micro-array data[2], mass spectrometry data[3, 4] and nuclear magnetic resonance data[5], but also clinically derived data[6, 7] and survey data[8]. From a bio-informatics perspective these biomedical data types vary significantly[1] and therefore extracting information out of biomedical data is not a trivial task. A framework for biomedical data analysis can help guiding biomedical engineers in their process of information extraction from their biomedical datasets. The framework can provide different options in processing the data, taking into account common dataset issues[9, 10, 11] and approaches to reach a certain goal[12, 13]. Currently available frameworks however mainly focus on the integration of databases[14, 15], are made specifically for one research area[16, 17, 18] or are limited to one specific type of analysis[19]. A framework that combines database integration, multiple research areas and multiple types of data analysis would be very beneficial for biomedical engineers, guiding them through their biomedical data analysis projects.

Before actual analysis can be done, a course of action needs to be chosen and the dataset usually needs to be preprocessed. This preprocessing can vary from removing redundant values to normalizing data. The course of action can vary from a type of quality testing to starting with a certain analysis technique. To find the right types of preprocessing and data analyses, initial analysis can be done. The goal of this research is *to find initial analyses that help finding suitable preprocessing algorithms and show a direction for data analysis.*

A possible way of finding out possible suitable preprocessing techniques is by using meta-features. Meta-features are dataset specific values that have information on a certain aspect of the dataset[20] (Appendix A). With the use of the information available in these meta-features suitable preprocessing and analysis techniques can be found and help the scientist in its research.

2 Background

First two datasets are introduced. These datasets are different in multiple ways and several preprocessing and analysis techniques are known to be suitable for these datasets. Secondly, multiple well known preprocessing and analysis techniques are discussed that are used often in the initial phases of research.

2.1 Datasets

Two different datasets are used to show the use of the datasets. Each of the datasets has different aspects, which makes them useful for testing a wider variety of meta-features. These datasets are based on micro organisms and hepatitis:

- *Micro organisms mass spectrometry dataset*
  This dataset is created to back up a proposed method for routinely performing direct mass spectrometry based bacterial species identification[21]. It consists of 1300 features corresponding to different spectra of the mass spectrometry data and 20 test subject groups corresponding to Gram positive and negative bacterial species. Gram classification is a result of a Gram stain test[22]. The groups differ in size varying from 11 to 60 samples, making a total of 571 samples. This dataset is created from mass spectrometry data. Mass spectrometry data has peaks in its data, most values are 0 or close to and a few peaks have a much higher value. This indicates that feature distributions can be very different. Also, the number of features is quite high, and most likely a smaller feature selection can be made.

- *Hepatitis dataset*
  The mortality rate of hepatitis was tested using 19 features over 155 samples. Two class types are distinguished as "died" and "lived" with attributes orginating from both clinical

and survey samples. On top of that, missing values are known to be present. Two articles that used this dataset were written[23, 24]. This hepatitis dataset consists of two types of data and therefore potential differences in types and distribution are present. Also missing values are known to exist in this dataset, which also need to be detected.

2.2 Preprocessing and Analysis

Before a dataset is properly used in analysis, several possible issues must be addressed. All of these issues have techniques to overcome them. A selection of these issues is discussed, together with ways to find those issues and example methods to address them.

2.2.1 Feature Types

Features values can be of multiple types, for example numbers, text, dates and other possibilities. The usefulness of these types can be different for several analyses, showing that not all techniques are suitable for a dataset. Usually all of these types can be put in one of these three groups:

- *Categorical data*: This type of data splits the features into different categories, giving a finite number of options for the feature values. These categories are usually text-based and lack special relations between them, such as an ordering[25]. Analyses that are based on making choices between different categories prefer categorical data. Examples are Bayesan models[26] and decision trees[27]. These choices are quickly made, due to the already made divisions between the categories.

- *Numeric data*: Features that have numbers as values that originate from a predefined interval are of the numeric type. This type has ordering in them and numeric values usually have infinite possibilities on the interval. Therefore the cardinality (number of distinct feature values) usually is much higher for numeric data[28]. Analyses that use numbers to compute the output prefer numeric data. A good example for this is regression analysis[29] that creates functions for which a numeric value is input to directly compute the output. These analysis do not limit the possibilities in choices and therefore look to a bigger spectrum of possibilities.

- *Ordinal data*: Ordinal data specifics are in between categorical and numeric data. Ordinal features have multiple predefined categories that have some kind of ordering. This ordering however usually is subjective and therefore cannot be placed on an interval[30]. Ordinal data is harder to define in data types, due to the lack of ordering in text based values and the lack of possibilities in defining only a finite number of categories in numerical data. Therefore usually the most suitable of categorical data and numerical data is chosen and ordinal data is transformed in that type.

The difference in feature types usually can be seen by the data type of the values. If the data type consists of only numbers, the feature is most likely numeric. If feature values are text based however, the feature is more likely to be categorical.

A dataset can of course also consist of multiple data types. This makes analyses incompatible, because techniques usually only requiring categorical or numeric data. This incompatibility can be removed by transferring one type into the other. Categorical data can for example be changed into numeric data by hot encoding[31], creating one features for every category and giving every feature value a 0 if that is not the category and a 1 if it is. This can create many additional features, however does remove potential bias by adding a non-existent ordering. Numerical data can be changed into categorical data by cutting up the interval in bins and putting the numeric values in such bins[32]. Every interval then becomes a category, for which ordering is lost but closely related values are kept together.

2.2.2 Feature Importance Imbalance

A dataset contains measurements for multiple features. When multiple features are numeric these measurements can be very different in intensity due to multiple reasons, such as the choice in units or measurements being on different scales. This difference in intensity can create problems in measurements due to bigger values possible having bigger effects on the outcome. Sensitivity can cause problems as well, not being able to process smaller differences between values within one feature than within the other. This creates a measure of unbalance, that features become more important only due to distribution specifics[33, 34].

The detection of possible unbalanced features can be done by comparing distributions aspects of a datasets. If distribution specifics such as mean and variance differ greatly, the features are possibly unbalanced, creating bias in analyses.

The removal of this bias is usually by standardisation or normalisation. Standardisation changes distributions in such a way that the distribution specifics become equal. Usually that means that the mean, standard deviation or both are set to a standard value, for examples mean $\mu = 0$ and $\sigma = 1$. Normalisation similarly puts all values of a distribution on a predefined interval, for example interval $[0, 1]$ for which the minimum value becomes 0 and the maximum becomes 1[33, 34].

2.2.3 Output Imbalance

Every instance is labelled with an output value. These labels are distributed over different classes for classification data. If the classes do not occur a similar number of times, the analysis techniques may give biased results. These analysis techniques will focus more on classes that are overrepresented and modify the result accordingly In cases for which misclassification is not equally important for every output class, additional measures must be made to properly show that.

This output imbalance can be found before actually starting the analysis. by looking at the class probabilities. A proper representation would mean that all classes have roughly the same number of instances.

Addressing this imbalance in the analysis of the data can be done by choosing different quality measurements. This quality can be measured in multiple ways, several of them are explained. All of them are defined in terms of true and false positives (TP and FP respectively) and true and false negatives (TN and FN) and n being the size of the test set.

- **Accuracy**
  The simplest and most used approach to measure quality is by using the accuracy. This is usually the choice when no imbalance is present and there is no specific focus in results. Accuracy is simply the ratio of correctly classified samples and all classified samples (Equation 1). So for a test set of 30 samples, if 6 samples are incorrectly classified the accuracy would be $24/30 = 0.8$.

$$\text{accuracy} = (TP + TN)/(TP + FP + TN + FN) \tag{1}$$

- **F1 score** The accuracy assumes there is no difference in output categories. If the output is unbalanced and majority and minority classes exist, another approach can be used called the F1 score. The F1 score treats classes equally by computing the precision and recall and combining those. Precision shows the portion correctly classified out of all positively classified (Equation 2) and recall shows the portion correctly positively classified out of all positive values (Equation 3). After that both are balanced in the F1 score (Equation 4). Therefore F1 evaluates both FP and FN evenly. Precision and recall are also sometimes used if one of the false classifications FP and FN is more important than the other.

$$\text{precision} = TP/(TP + FP) \tag{2}$$

$$\text{recall} = TP/(TP + FN) \tag{3}$$

$$\text{F1} = 2\frac{\text{precision} \cdot \text{recall}}{\text{precision} + \text{recall}} \tag{4}$$

- **Cohen's kappa**
  Cohen's kappa κ is a more statistical approach for a quality label. κ shows the agreement between the result p_0 (for example the aforementioned accuracy) and compares this with the expected agreement by chance p_e (Equation 5) It is mainly used if the output can be two classes[35] and does not take into account random classification, therefore better gives an indication of the effectiveness.

$$\kappa = \frac{p_0 - p_e}{1 - p_e} = \frac{2(TP \cdot TN - FP \cdot FN)}{(TP + FP) \cdot (TN + FN) + (TP + TN) \cdot (FP + FN))} \tag{5}$$

2.2.4 Missing Values

Datasets are hardly ever perfect. Mistakes were possibly made during the design, or entries were incorrect due to errors by the designer. Another problem often occurring in dataset is the presence of missing values. Mistakes can be made somewhere in the creation leading into missing entries and most analysis techniques can not work with data including missing values.

Missing values usually are easily spotted. Most of the time a specific value is given to instances, such as 'None', 'NaN' (not a number) or just an empty entry. Other times it is a default value, such as a 0 or a default text value. This detection usually is not a big issue. More important is the way to cope with these missing values.

Several techniques are available to cope with missing values (Report Missing Values). Choosing between these techniques may differ between the ratio of missing values present per feature and instance and mainly focuses on either deleting entire features and instances or imputation of a good representing value.

2.2.5 Feature relevance

The dataset consists of a certain number of instances and features. The scientist usually rather have a high number of instances, as then the results are more accurate. A higher number of features also means more information is available. The number of useful features usually is limited, however. Moreover, if the number of features far exceeds the number of instances, bias may be created due to analysis algorithms not being able to filter out the irrelevant features with the abundance of information.

In case of a feature abundance, feature selection can be useful to remove irrelevant features. Whether feature selection might be useful first can be seen by dividing the number of features by the number of instances, also known as the dimensionality. If the dimensionality is higher than one, feature selection can be used to reduce it. Aside from a high dimensionality, the presence of irrelevant features can be found in several ways:

- *Potential information:* Every value holds potential information that might contribute to the final product. The potential information can be measured by for example the entropy[25] (Appendix A). If the potential information is very low for one or multiple features, they are worth the consideration of being removed.

- *Predictive power:* Aside from the potential information present, this information will only be useful if relations to the output are present. Mutual information is an example that shows that, by not only looking at the information of the features, but also comparing it with the information of the output. If the information present in the feature can be used to explain the output, the mutual information will be higher[36] (Appendix A).

Several feature selection techniques are available (Report Feature Selection). These techniques show different possibilities that try to preserve as few features as possible and still show good results.

2.2.6 Multicollinearity

When looking at features separately, feature selection on feature relevance is effective. This usually does not take relations between features into account. Sometimes features are highly related between each other which may include overlap in the information present for the outcome. The presence of relations between features is called multicollinearity and reduction of multicollinearity will create more effective information per feature.

Detection of multicollinearity is in principle very simple. All features with each other can be compared to find out how high the relation between the features is. In practice this can be done efficiently for a low number of features, but combinatorial explosions can occur when the number becomes higher, as the number of combinations becomes n^2 with n being the number of features. Therefore this must be done with care for bigger datasets. An example of finding multicollinearity is correlation (Appendix A) that compares the difference between the features per instance with the mean difference. Another example is PCA (Appendix A) that computes how much variance can be combined in as few newly created features as possible, showing the relation in variances between features.

PCA is an effective way of removing multicollinearity. It does however use newly created features which makes linking input and output directly much harder to understand. Several feature selection techniques also take multicollinearity into account, such as wrapper methods (Report Feature Selection). Also clustering features can contribute to less multicollinearity, combining the cluster into one feature[37].

2.3 Meta-features Package

An implementation that focuses on extracting meta-features from dataset is the package *metalearn*. This package implements most of the aforementioned meta-features. It requires *Python 3.6* and eleven separate packages, most of them being used very often in data analysis (for example *numpy*, *scipy*, *scikit-learn* and *pandas*). All of these packages are available in *Anaconda*, which makes the usage of *metalearn* very easy.

Usage for this package is quite intuitive and for the user mainly consists of one class *Metafeatures*. After initialization of this class, it can compute the meta-features of a dataset with the method *compute()*. The input for this method should consists of a dataset and an output array. Additionally an indication for every feature of the dataset being categorical or numeric can be added if this is not clear by feature types, a subset of all meta-features can be added if only those are required and some implementation specific parameters can be added if needed.

The number of meta-features that can be extracted is very long. Themes which they can be put in are given, as well as textual explanation. The names of the meta-features are self-explanatory and therefore not given for a better explanation:

- **Basic meta-feature themes**

  1. *Data size*
     The number of instances, features and classes and the dimensionality (number of features divided by the number of instances).
  2. *Data types*
     The number and ratio of both the numeric and the categorical features.
  3. *Missing values*
     The number and ratio of the total number of missing values and the number and ratio of both instances and features with missing values.

4. *Output type*
   The average probability of each class, as well as its standard deviation, minimum and maximum and the size of the minority and majority class.
5. *Feature cardinalities*
   The mean, standard deviation, minimum and maximum cardinality of both categorical and numeric features.

- **Statistical Meta-feature themes**

1. *Feature distributions*
   The mean, standard deviation, skewness and kurtosis were all tested for numeric features. The values of all of these features are shown by the mean, standard deviation, minimum, maximum and the first-, second- and third quartile after combining them.
2. *Principal component analysis*
   The ratio of variability explained by the first three principal components and the size of the first three eigenvalues.

- **Information-theoretic meta-feature themes**

1. *Entropy*
   The attribute and joint entropy are computed for both categorical and numeric features. Of these the mean, standard deviation, minimum, maximum and the first-, second- and third quartile values are recorded.
2. *Mutual information*
   For both categorical and numeric features the mutual information is tested. From this the mean, standard deviation, minimum, maximum and the first-, second- and third quartile values are recorded.
3. *Entropy and mutual information*
   The equivalent number of categorical and numeric features is given to explain the class entropy. Also the signal to noise ratio for the dataset is given for both categorical and numeric features.

- **Landmarking meta-feature themes**
  The error rate (1 - accuracy) and the Cohen's kappa κ are computed for a selection of machine learning algorithms. This selection consists of Naive Bayes, k-nearest neighours with $k = 1$, decision stump (decision tree with one node), random tree (decision tree with random splits and a depth of 1 to three nodes) and linear discriminant analysis.

Advantages of this package is the ease of use. When using an Anaconda environment, the use of *metalearn* is very simple. The variety in the number of meta-features is quite big and much information can be retrieved from them. Since all of these meta-features are a numeric value themselves, they can be easily used in a follow-up algorithm that can make use of these meta-features.

The number of meta-features is a disadvantage for a user, when using it directly. It is hard to find the relevant information from these values as there are so many of them. Also the conclusions that can be taken from these meta-features are sometimes a bit hard to grasp. Additional plots of distributions as well as several examples that indicate this would make it easier to understand the meta-features.

3 Hypotheses

The main goal of this research is to find out which preprocessing and analysis techniques should be used for datasets. To find those, several meta-features are provided, as well as the package

metalearn that computes several of those meta-features. For this goal, two hypotheses are made. The first hypothesis is the following:

H1: All mentioned issues, except for multicollinearity, are properly addressed in the meta-features form the package metalearn. Multicollinearity is only partly addressed.

Most of the issues are expected to be addressed with the proposed meta-features (). The feature types, feature importance imbalance, output imbalance, missing values and feature relevance all have known meta-features that can indicate these possible issues. Multicollinearity is only partly addressed with PCA, which is expected to not give enough insight in multicollinearity.

The second hypothesis is based on follow up after finding the meta-features:

H2: The understanding of the outcome of the meta-features is limited and further information would improve this.

The links between the issues and meta-features are given. The understanding of these links are not though and figures and examples to explain these links should be useful to add to these meta-features to further solidify the choice of using a preprocessing or analysis technique.

4 Methods

To test both hypotheses, two different analyses were created. The first analysis focuses on the results when only using the *metalearn* package, without any additions to it. The second analysis focuses on adding several additional results to support the meta-features and create better understanding.

Due to previous experiments with the two datasets, the issues for both datasets are known and should be represented in the meta-feature analyses:

- *Hepatitis:* This dataset consists of a mixture of two data types. The features and classes are imbalanced and missing values are present.
- *Micro organisms:* This dataset has a highly imbalanced feature importance, as well as imbalances in the multiple classes present. A big variation in feature relevance is present, as well as multicollinearity.

4.1 Existing Meta-features Evaluation

The first analysis is done to investigate the dataset issues with the *metalearn* package (Section 2.3) and all available meta-features in that package. The two datasets used are based on micro organisms and hepatitis (Section 2.1) and both contain known issues. The meta-features are then compared with those issues to find out if those can be extracted from them.

4.2 Additional Exploration Evaluation

For the second analysis several additional computations are done. These additional computations focus on correlation, low cardinalities, outlier examples and at last visual plots.

The designer of *metalearn* already initialized finding correlations between the values. The meta-features for this correlation were added for datasets with fewer than 1000 features, to prevent combinatorial explosion problems. The meta-features are the mean and the standard deviation of this correlation, as well as the three correlation combinations with the lowest and the highest value for this correlation.

Sometimes features only have one distinct value as output having a cardinality of one. This feature cannot contribute to the analysis, as there is no information present. To find a list for all

features with only one distinct value an additional method in the package is made that counts the number of features with the lowest cardinality. Aside from this count, also the features having this cardinality are given to potentially remove them.

Outlier examples are shown, to give an initial insight in the data. These examples show which features at least require more attention during analysis or even should be removed. These outliers are given for the following examples:

1. The instances and features with the most missing values.
2. The output classes that have the lowest or highest probability.
3. The categorical and numeric features with the lowest or highest cardinality.
4. The categorical and numeric features that have the lowest or highest attribute entropy.
5. The categorical and numeric features that have the lowest or highest joint entropy.
6. The categorical and numeric features that have the lowest or highest mutual information.

In multiple instances, the number of values can much better be explained by using some kind of plot. For the numeric distribution values for example, a boxplot for the means, standard deviations, skewnesses and kurtoses combines all of those meta-features. For the class counts a histogram shows these meta-features visually as well and at last the outlier distributions are better understandable when shown with a boxplot or histogram. Therefore multiple visual plots are made to show this.

5 Results

The results are also split into two different sections, corresponding to the methods.

5.1 Existing Meta-features Evaluation

All meta-feature values are computed and given for the hepatitis (Appendix B) and micro-organisms (Appendix C) datasets. The results are also split into two sections, each dataset evaluated in one subsection. As can be seen, the number of meta-features is very high and therefore difficult to find the relevant ones. Therefore a selection is made for the visible meta-features, based on the (subjective) estimation that a conclusion can be derived from those.

5.1.1 Hepatitis Dataset

First the important meta-features of the hepatitis dataset are given:

- *Data size* Number of instances: 155
  Number of features: 19
  Number of classes: 2

  The number of instances is somewhat limited, but is not too low to hurt analysis. The number of features is low enough to really need feature selection and the number of classes shows a boolean result.

- *Data types*
  Number of numeric features: 6
  Number of categorical features: 13

  Both numeric and categorical features are present. This means the type of analysis will matter and possibly hot encoding or binning needs to be used for categorical features for further analysis.

- *Missing values*
  Number of missing values: 167
  Number of instances with missing values: 75
  Ratio of instances with missing values: 0.48
  Number of features with missing values: 15
  Ratio of features with missing values: 0.79

  There are missing values present and these missing values are spread out significantly over the instances (ratio of 0.48) and features (ratio of 0.79) and therefore something more than deletion would be advised to do.

- *Output type*
  Minimum of class probability: 0.21
  Maximum of class probability: 0.79

  The two classes deviate significantly in class probability (0.21 to 0.79), so the dataset is not balanced. This can create possible bias when using plain accuracy in quality measurements.

- *Feature cardinalities*
  Minimum categorical feature cardinality: 2
  Maximum categorical feature cardinality: 2

  The categorical feature cardinality shows that all categorical features are boolean, which limits the possible information present in these categorical features.

- *Feature distributions*
  Mean of means numeric features: 49.92
  Standard deviation of means numeric features: 42.57
  Mean of standard deviations numeric features: 29.74
  Standard deviation of standard deviations numeric features: 29.73
  Means of skewnesses numeric features: 1.28
  Means of kurtoses numeric features: 4.49

  Both the means and the standard deviations of the numeric features differ significantly from each other. Therefore scaling on both means and standard deviation per feature would be balance out feature importances. Also the skewness and kurtosis are both much higher than expected, showing that these values do not follow normal distributions.

- *Principal component analysis*
  Explained variation component 1: 0.69
  Explained variation component 2: 0.26

- *Entropy and mutual information*
  Mean mutual information categorical features: 0.04
  Mean mutual information numeric features: 0.08
  Categorical noise to signal ratio: 13.60
  Numeric nise to signal ratio: 11.90

  The mutual information of the categorical features is low, averaging at a mutual information of 0.04. The numeric features contain about twice as much information with an average mutual information of 0.08. Also relatively a lot of noise is present, compared with the signal.

- *Landmarking*
  Naive bayes error rate: 0.22
  Naive bayes kappa: 0.48
  Decision stump error rate: 0.21
  Linear discriminant error rate: 0.17
  Linear discriminant kappa: 0.46

  The error rates can show possible bias as, indicated from the deviation in class probabilities. Of al algorithms, the linear discriminant and naive bayes already show an error rate of around 0.2 and a kappa of 0.5, showing potential for a good machine learning model. The decision stump and random tree with depth one shows that only using one feature can already give a quality of 0.8, being quite high.

The meta-features confirm the presence of missing values and show that these missing values can cause a problem, due to their spreading. The feature types (categorical and numeric) both are found and specific issues that need focus for the categorical values (low cardinality) and the numeric features (distribution differences) are found, too, as well as the information they provide. Additional useful information is found in PCA, showing that the 19 features share much variation and that using simple analysis techniques show promising results. A better focus on locations of multicollinearity and mutual information is missing however, as of these 19 features no specific information on the features is present.

5.1.2 Micro-organisms Dataset

Secondly the most important meta-features are given for the micro organisms dataset:

- *Data size*
  Number of instances: 571
  Number of features: 1300
  Number of classes: 20
  Dimensionality: 2.28

  The number of features for this dataset is very high in comparison with the number of instances. The dimensionality is preferably much lower, which already indicates that feature selection would be a good approach for this dataset. The number of classes shows that is high in comparison with the number of instances as well, so attention to the class probabilities is useful.

- *Data types*
  Number of categorical features: 0

  The number of categorical features is 0, so only numeric features are present. This indicates that we do not need to look at categorical information in the data.

- *Output type*
  Minimum of class probability: 0.02
  Maximum of class probability: 0.1

  The classes are not balanced. This indicates that using the accuracy might give bias.

- *Feature Cardinality*
  Mean cardinality numeric features: 45.29
  Minimum cardinality numeric features: 1

The mean cardinality is lower than expected for numeric values with 571 instances. It also shows that at least one feature has a cardinality of 1 and can therefore be removed from the data, not having any meaning.

- *Feature distributions*
  Mean of means numeric features: 67437.29
  Standard deviation of means numeric features: 191740.68
  Mean of standard deviations numeric features: 343176.13
  Standard deviation of standard deviations numeric features: 700237.17
  Mean of skewnesses numeric features: 10.67
  Mean of kurtoses numeric features: 177.27

  The mean and standard deviation of both the means and standard devations of the numeric features, show that scaling is needed to balance the feature importances. Also, the mean skewness and kurtosis show that the distribution is highly irregular and cannot be seen as a normal distribution.

- *Entropy and mutual information*
  Maximum mutual information numeric features: 0.27
  Third quartile mutual information numeric features: 0.06
  Equivalent number of numeric features: 70.27
  Numeric noise to signal ratio: 1.64

  The maximum mutual information is relatively higher than the third quartile mutual information. This can indicate that several outlier features are significantly more useful in predicting the output. The equivalent number of numeric features is quite high and indicates on average a high number of features is needed to predict the output. At last there seems to be more noise than signal, so noise reduction might be useful to add.

- *Landmarking*
  Naive bayes error rate: 0.20
  Naive bayes kappa: 0.79
  Decision stump error rate: 0.83

  The naive bayes error rate and kappa are significantly better than for the other algorithms. So analysis in the area of naive bayes seems like a good start. Also the decision stump error rate shows a very high error rate. This high error rate is due to the splitting method in decision stumps, only being able to split the instances in two groups, even though 20 different groups are present.

The high number of features was known beforehand and the meta-features confirm that this issue needs to be handled. At start it seems that several already can be removed because if the low cardinality of 1 and the very low mutual information. Also, due to the data being gathered by mass spectrometry the distributions are very different, which indicated normalisation to be useful. The number of features needed to create a good prediction seems to be high, knowing the equivalent number of numeric features. At last basic naive bayes already shows a good result in this, therefore more in depth analysis in that area might be useful.

5.2 Additional Exploration Evaluation

All additional computation results are given for the hepatitis and the micro-organisms dataset. The results are also split into two sections, each dataset evaluated in one subsection.

5.2.1 Hepatitis Dataset

The additions are shown and discussed per different addition theme. First the correlation is discussed, secondly the minimum cardinality, followed by the outliers and at last the plots.

- *Correlation*
  Mean correlation: 0.08
  Standard deviation correlation: 0.20

  The mean correlation of the dataset is not very high and indicates that multicollinearity should not create big problems. The relatively high standard deviation does indicate that there might be some features that have a very high correlation with each other. This indicates further investigation.

- *Minimum cardinality*
  Minimum cardinality feature count: 13

  Whereas the minimum cardinality being 2, the minimum cardinality count could already be deduced to 13, being all 13 categorical values being boolean.

- *Outliers*

  All outliers are computed and the lowest and highest three outliers are shown (Table 1). Some of these outliers seem to need care:

  1. The feature *STEROID* misses 43% of its values, does contain the most information in both attribute and joint entropy, but not having the most mutual information. The ratio of missing values is much more in this feature than in the others. Also, considering the number of instances not being high (155), the removal of this feature might be a good idea.
  2. The feature *LIVER_FIRM* has the highest correlation with the features *SPIDERS, ANOREXIA, SEX* and *AGE*, but no mutual information with the output. To reduce multicollinearity without losing much information, this feature could easily be removed.
  3. The features *AGE* and *ALK_PHOSPHATE* both show having a very high potential information, but not having a high mutual information.
  4. The features *PROTIME*, *ALBUMIN*, *BILIRUBIN*, *ASCITES* and *SPIDERS* all show a high mutual information and of these *BILIRUBIN*, *ALBUMIN* and *ASCITES* even show a low potential information, so most likely not having a high amount of noise. These five most likely are very useful in predicting the output.

- *Plots*

  The values for the plots are computed and visualized. Whereas these plots do not show any additional information, they do show earlier mentioned problems:

  1. *Class distribution* (Figure 1)
     The class distribution clearly visualizes the imbalance between the output classes and makes the user think about approach to deal with this.

Table 1: The outliers for the hepatitis dataset

Outlier Type	Lower bound			Upper bound		
	First Outlier (name)	Second Outlier (name)	Third Outlier (name)	Third Outlier (name)	Second Outlier (name)	First Outlier (name)
Ratio of instances with missing values	–	–	–	0.37 (80)	0.37 (10)	0.74 (49)
Ratio of features with missing values	–	–	–	0.10 (ALBUMIN)	0.19 (HISTOLOGY)	0.43 (STEROID)
Class probabilities	0.21 (1)	–	–	–	–	0.79 (0)
Categorical cardinality	2 (SEX)	2 (ANTIVIRALS)	2 (HISTOLOGY)	2 (SPIDERS)	2 (ASCITES)	2 (VARICES)
Numeric cardinality	30 (ALBUMIN)	35 (BILIRUBIN)	45 (PROTIME)	49 (AGE)	84 (ALK_PHOSPHATE)	85 (SGOT)
Correlation combinations	-0.55 (LIVER_FIRM, SPIDERS)	-0.40 (ANOREXIA, LIVER_FIRM)	-0.38 (MALAISE, SPIDERS)	0.47 (ANTIVIRALS, SPLEEN_PALPABLE)	0.60 (LIVER_FIRM, SEX	0.60 (LIVER_FIRM, AGE)
Categorical attribute entropy	0.33 (SEX)	0.37 (VARICES)	0.39 (ASCITES)	0.68 (LIVER_FIRM)	0.69 (HISTOLOGY)	0.69 (STEROID)
Numeric attribute entropy	0.60 (SGOT)	0.64 (BILIRUBIN)	1.12 (ALBUMIN)	1.19 (PROTIME)	1.27 (ALK_PHOSPHATE)	1.32 (AGE)
Categorical joint entropy	0.81 (ASCITES)	0.82 (SEX)	0.82 (VARICES)	1.14 (HISTOLOGY)	1.16 (LIVER_FIRM)	1.19 (STEROID)
Numeric joint entropy	1.05 (BILIRUBIN)	1.09 (SGOT)	1.47 (ALBUMIN)	1.52 (PROTIME)	1.72 (ALK_PHOSPHATE)	1.80 (AGE)
Categorical mutual information	0.00 (LIVER_FIRM)	0.00 (LIVER_BIG)	0.01 (ANOREXIA)	0.06 (HISTOLOGY	0.08 (SPIDERS)	0.09 (ASCITES)
Numeric mutual information	0.01 (SGOT)	0.03 (AGE)	0.03 (ALK_PHOSPHATE)	0.09 (BILIRUBIN)	0.14 (ALBUMIN)	0.17 (PROTIME)

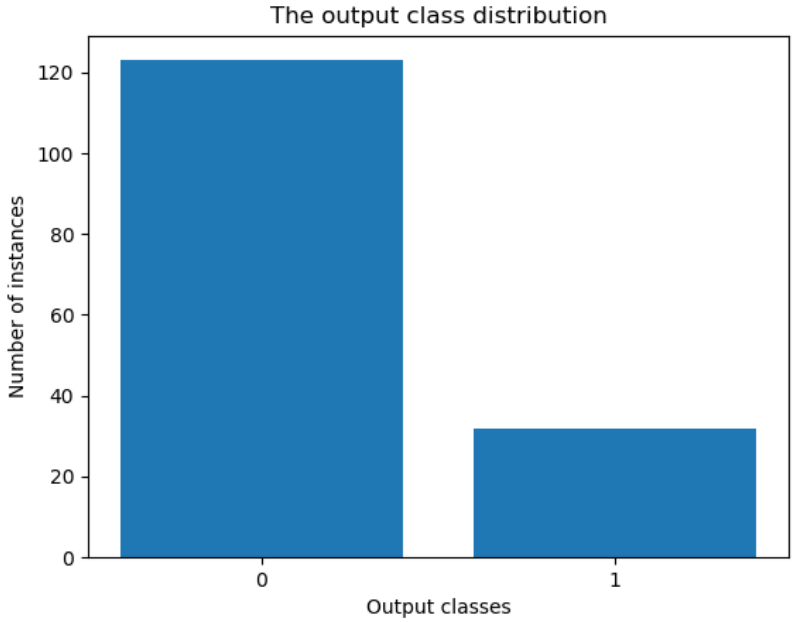

Figure 1: The class distribution of the output of the hepatitis dataset.

2. *Numeric feature distributions* (Figure 2)
   These distributions show very clearly how the feature values are distributed and that normalisation and standardisation should be considered to remove the feature value imbalance.

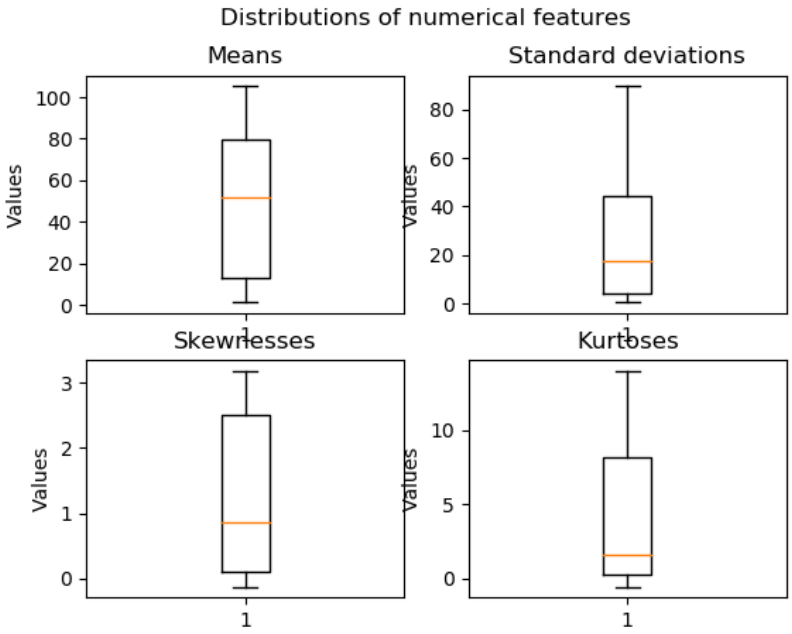

Figure 2: The boxplots of the means, standard deviations, skewnesses and kurtoses from the features of the hepatitis dataset.

3. *Categorical and numeric cardinality* (Figures 3 and 4)
   The distributions for these cardinality outliers are shown, but not very relevant in this case. No additional information can be found in this case.

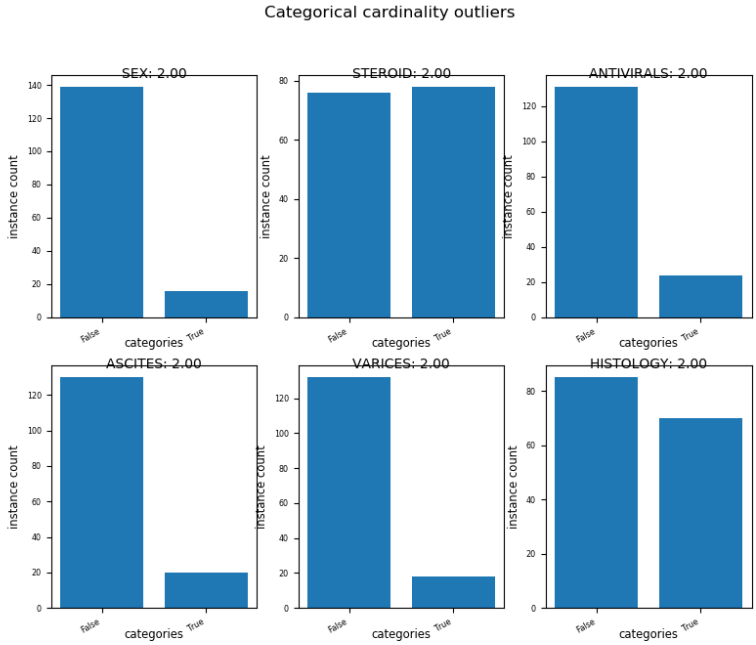

Figure 3: Histograms of the categorical features with either the highest or the lowest cardinality in the hepatitis dataset.

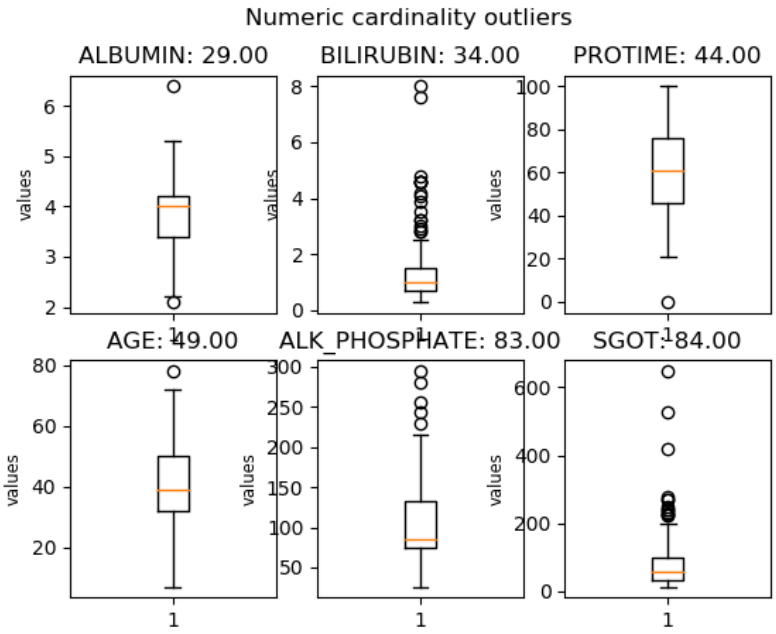

Figure 4: Boxplots of the numeric features with either the highest or the lowest cardinality in the hepatitis dataset.

4. *Attribute entropy outliers* (Figures 5 and 6)
   Especially the figure for categorical distributions show very well how features differ in entropy. The features *SEX*, *VARICES* and *ASCITES* show a strong bias towards the value FALSE.The other three maximum outliers are far more balanced.

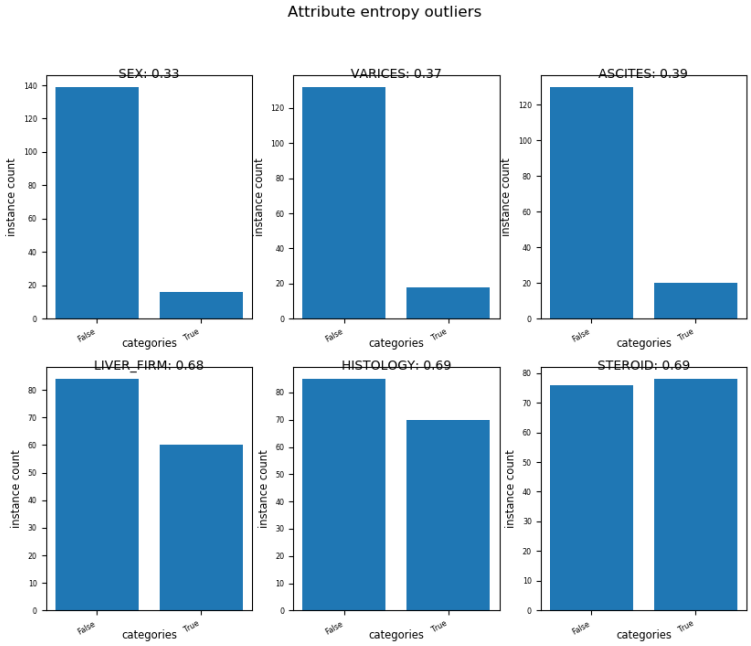

Figure 5: Histograms of the categorical features with either the highest or the lowest attribute entropy in the hepatitis dataset.

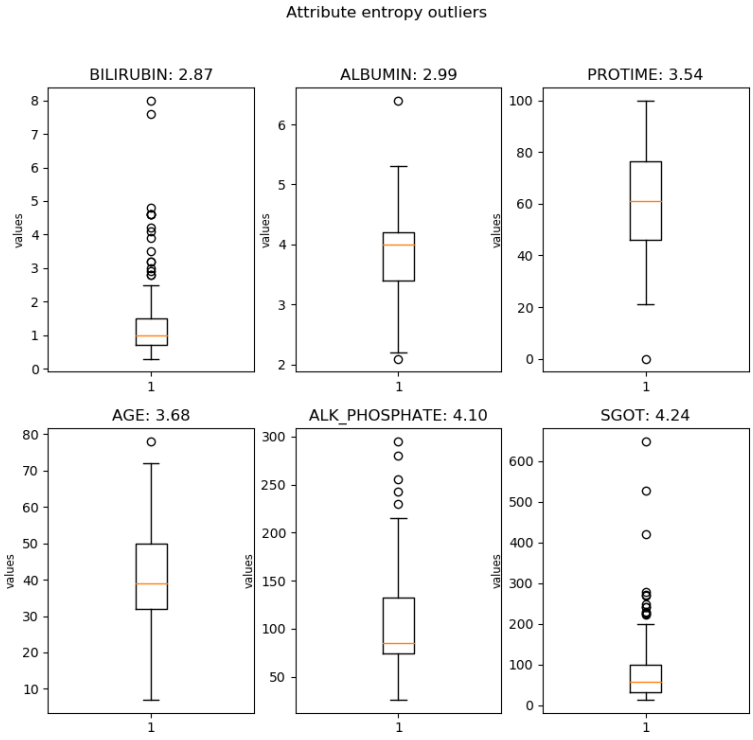

Figure 6: Boxplots of the numeric features with either the highest or the lowest attribute entropy in the hepatitis dataset.

Looking at the overall additional results, several aspects are improved. More insight in correlation is given and several features are known to most likely be irrelevant or to the contrary very relevant. Also the understanding of the issues is better with the histograms and the boxplots.

5.2.2 Micro-organisms Dataset

The additions are shown and discussed per different theme. Also, some of the additions are not discussed, due to lack of relevance. These can still be read (Appendix **??**). First the correlation is discussed, followed by the minimum cardinality. Thirdly the outliers are discussed and lastly the plots.

- *Correlation:*

  Due to the imposed restriction of combinatorial explosion, correlation will not be checked for more than 1000 features, and this dataset has 1300 features.

- *Minimum cardinality*
  Minimum cardinality feature count: 218

  A total of 218 features have a cardinality of one and therefore do not contain any information. Aside from this, also the names of the 218 features are returned, but not shown due to its size.

- *Outliers:*
  (Table 2)

All outliers are computed and the lowest and highest three outliers are shown (Table 2). Some results need further discussion:

1. The 20 possible class probabilities show no extreme outliers. Imbalance is still present, but there is no majority problem.
2. The feature *V1261* has the most potential information and second to most mutual information and therefore is a feature to keep an eye on.
3. Again the features with one distinct value are found in the lowest outliers.
4. The closeness in entropy and mutual information for the top three features, shows that far more features are interesting to look at and that these features would be useful.

Table 2: The outliers for the micro-organisms dataset

	Lower bound			**Upper bound**		
Outlier Type	**First Outlier (name)**	**Second Outlier (name)**	**Third Outlier (name)**	**Third Outlier (name)**	**Second Outlier (name)**	**First Outlier (name)**
Ratio of instances with missing values	–	–	–	–	–	–
Ratio of features with missing values	–	–	–	–	–	–
Class probabilities	0.02 (7)	0.02 (6)	0.03 (9)	0.09 (19)	0.09 (3)	0.11 (11)
Categorical cardinality	–	–	–	–	–	–
Numeric cardinality	1 (V1)	1 (V447)	1 (V1090)	259 (V1261)	294 (V719)	296 (V838)
Correlation combinations	–	–	–	–	–	–
Categorical attribute entropy	–	–	–	–	–	–
Numeric attribute entropy	0.0 (V1)	0.0 (V1099)	0.0 (V212)	0.60 (V1163)	0.62 (V1002)	0.83 (V1261)
Categorical joint entropy	–	–	–	–	–	–
Numeric joint entropy	2.91 (V1)	2.91 (V1071)	2.91 (V782)	3.34 (V1163)	3.35 (V1002)	3.49 (V1261)
Categorical mutual information	–	–	–	–	–	–
Numeric mutual information	0.00 (V1)	0.00 (V251)	0.00 (V248)	0.25 (V787)	0.25 (V1261)	0.27 (V1008)

- *Plots*

The values for the plots are computed and visualized. Whereas these plots do not show any additional information, they do show earlier mentioned problems:

Class count histogram: Figure 7

The class count histogram show that most of the classes are around 25 to 30 instances. There are however seven classes that have significant more or less instances.

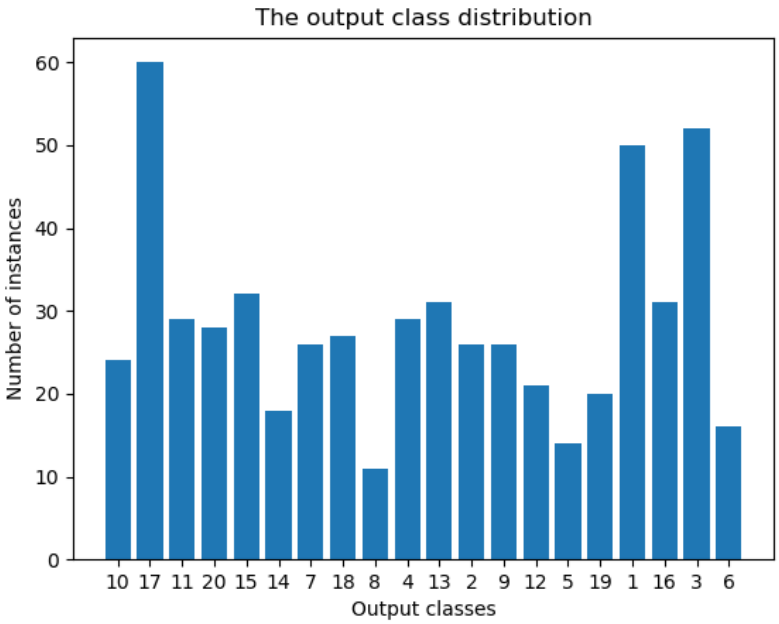

Figure 7: The class distribution of the output of the micro-organisms dataset.

Distribution boxplots: Figure 8

These boxplots show very well the difference in values for every feature. The creation of a boxplot of the means and standard deviation and the high number of outliers, show that normalisation or standardisation seems a good preprocessing action.

Numeric cardinality outlier boxplots: Figure 9

The numeric cardinality boxplots show very well how the feature values are distributed. Due to this dataset orginating from mass spectrometry, most values are 0. All values that are not 0 can become significantly high, with outliers having a factor 10^7. Again the outliers with the lowest cardinality have just one value, 0.

Numeric attribute entropy boxplots: Figure 10

The attribute entropy outliers cannot directly be seen in the box plot. This plot does not seem very useful in this case.

At first finding out that 218 features in the dataset that contain no information is very useful in initial understanding of the dataset. The outliers mainly show that most likely many more features are useful in finding a good result. The histograms and boxplots are more useful as they give a better view in the class distribution. They also visualize the distributions of both the means, standard deviations and also the feature distributions, giving more understanding and insights.

6 Discussion

There are numerous ways biomedical datasets are gathered and therefore numerous issues that can be found in the dataset. The six discussed issues are well known and therefore a good representation, however there are more issues. For example the issue of having too few instances

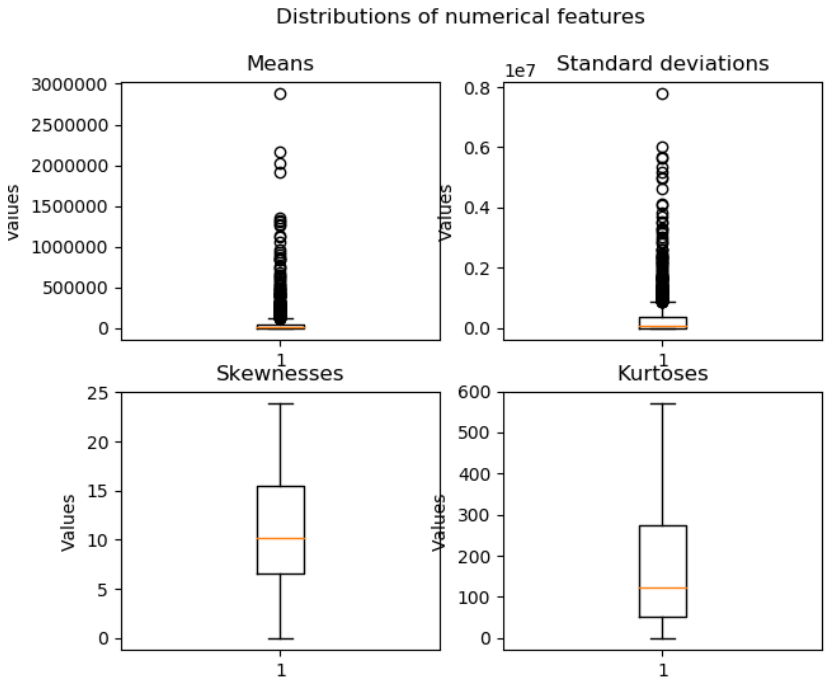

Figure 8: The boxplots of the means, standard deviations, skewnesses and kurtoses from the features of the micro-organisms dataset.

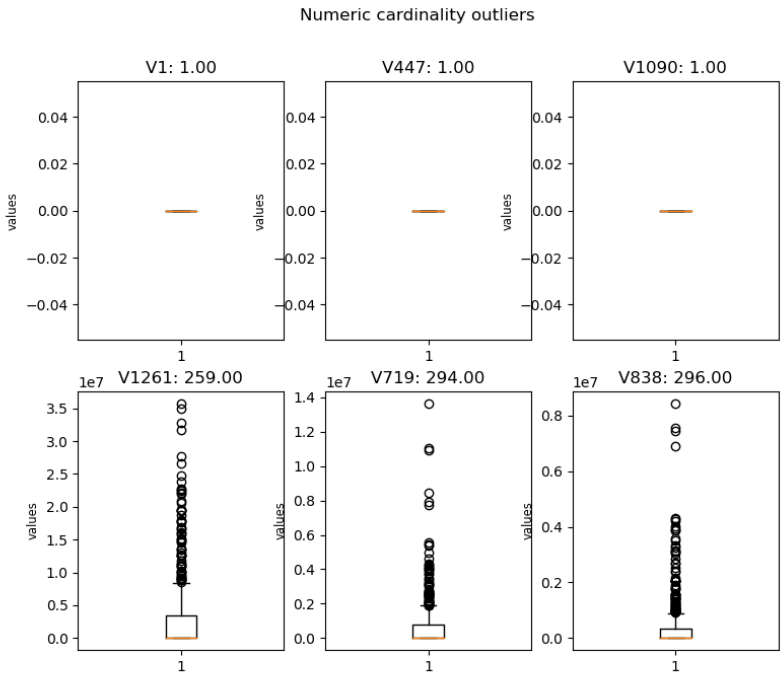

Figure 9: Boxplots of the numeric features with either the highest or the lowest cardinality in the Micro Organisms dataset.

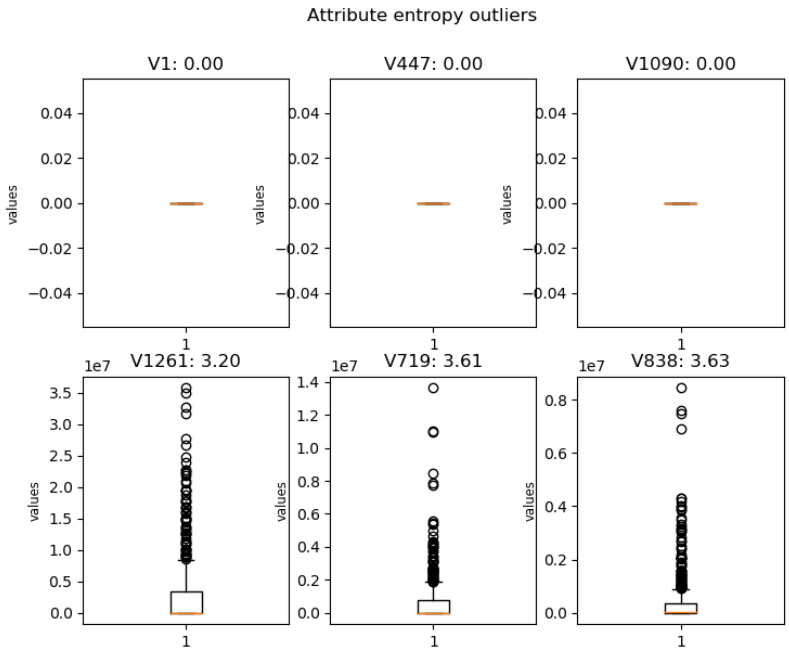

Figure 10: Boxplots of the numeric features with either the highest or the lowest attribute entropy in the Micro organisms dataset.

and trying to generate more is not discussed or the possibility of anomalies being present in the data. These two issues may be addressed in future researches.

7 Conclusion

The conclusions are based on hypotheses H1 and H2. The first hypothesis H1 is confirmed, in that all issues except for multicollinearity were found in the two datasets with the current meta-features. Multicollinearity would need more addressing besides these meta-features. The information from the meta-features is tough to understand, though. The sheer number of the meta-features makes it hard to find the issues and the lack of examples and visualization makes it hard to how these issues manifest in the dataset, therefore H2 is also confirmed.

The problem posed in the confirmed H2 and the lack of multicollinearity testing in H1 is partly removed by the addition of new meta-features, outliers and histograms and boxplots. These make a better understanding, show several features that need more attention and also show better if multicollinearity can be a problem for the dataset. Therefore the additions are useful for an addition in dataset exploration

References

[1] N. Gehlenborg, S. I. O'donoghue, N. S. Baliga, A. Goesmann, M. A. Hibbs, H. Kitano, O. Kohlbacher, H. Neuweger, R. Schneider, D. Tenenbaum, *et al.*, "Visualization of omics data for systems biology," *Nature methods*, vol. 7, no. 3s, p. S56, 2010.

[2] A. Brazma, P. Hingamp, J. Quackenbush, G. Sherlock, P. Spellman, C. Stoeckert, J. Aach, W. Ansorge, C. A. Ball, H. C. Causton, *et al.*, "Minimum information about a microarray

experiment (miame)—toward standards for microarray data," *Nature genetics*, vol. 29, no. 4, p. 365, 2001.

[3] J. S. Cottrell and U. London, "Probability-based protein identification by searching sequence databases using mass spectrometry data," *electrophoresis*, vol. 20, no. 18, pp. 3551–3567, 1999.

[4] K. Dettmer, P. A. Aronov, and B. D. Hammock, "Mass spectrometry-based metabolomics," *Mass spectrometry reviews*, vol. 26, no. 1, pp. 51–78, 2007.

[5] D. Capitani, A. P. Sobolev, and L. Mannina, "Nuclear magnetic resonance–metabolomics," *Food Authentication: Management, Analysis and Regulation*, p. 177, 2017.

[6] B. Liu, X. Zhou, Y. Wang, J. Hu, L. He, R. Zhang, S. Chen, and Y. Guo, "Data processing and analysis in real-world traditional chinese medicine clinical data: challenges and approaches," *Statistics in medicine*, vol. 31, no. 7, pp. 653–660, 2012.

[7] D. F. Sittig, A. Wright, J. A. Osheroff, B. Middleton, J. M. Teich, J. S. Ash, E. Campbell, and D. W. Bates, "Grand challenges in clinical decision support," *Journal of biomedical informatics*, vol. 41, no. 2, pp. 387–392, 2008.

[8] G. Magni, C. Caldieron, S. Rigatti-Luchini, and H. Merskey, "Chronic musculoskeletal pain and depressive symptoms in the general population. an analysis of the 1st national health and nutrition examination survey data," *Pain*, vol. 43, no. 3, pp. 299–307, 1990.

[9] P. Bertolazzi, G. Felici, P. Festa, and G. Lancia, "Logic classification and feature selection for biomedical data," *Computers & Mathematics with Applications*, vol. 55, no. 5, pp. 889–899, 2008.

[10] G. Piatetsky-Shapiro and P. Tamayo, "Microarray data mining: facing the challenges," *ACM SIGKDD Explorations Newsletter*, vol. 5, no. 2, pp. 1–5, 2003.

[11] A. Lommen, "Metalign: interface-driven, versatile metabolomics tool for hyphenated full-scan mass spectrometry data preprocessing," *Analytical chemistry*, vol. 81, no. 8, pp. 3079–3086, 2009.

[12] A. Holzinger, M. Dehmer, and I. Jurisica, "Knowledge discovery and interactive data mining in bioinformatics-state-of-the-art, future challenges and research directions," *BMC bioinformatics*, vol. 15, no. 6, p. I1, 2014.

[13] M. Wilkins, "Proteomics data mining," *Expert review of proteomics*, vol. 6, no. 6, pp. 599–603, 2009.

[14] D. Teodoro, R. Choquet, E. Pasche, J. Gobeill, C. Daniel, P. Ruch, and C. Lovis, "Biomedical data management: a proposal framework.," in *MIE*, pp. 175–179, Citeseer, 2009.

[15] M. Y. Galperin, "The molecular biology database collection: 2008 update," *Nucleic Acids Research*, vol. 36, no. suppl1, pp. D2–D4, 2008.

[16] A. Sturn, J. Quackenbush, and Z. Trajanoski, "Genesis: cluster analysis of microarray data," *Bioinformatics*, vol. 18, no. 1, pp. 207–208, 2002.

[17] A. Karnovsky, T. Weymouth, T. Hull, V. G. Tarcea, G. Scardoni, C. Laudanna, M. A. Sartor, K. A. Stringer, H. Jagadish, C. Burant, *et al.*, "Metscape 2 bioinformatics tool for the analysis and visualization of metabolomics and gene expression data," *Bioinformatics*, vol. 28, no. 3, pp. 373–380, 2011.

[18] D. Tabas-Madrid, R. Nogales-Cadenas, and A. Pascual-Montano, "Genecodis3: a non-redundant and modular enrichment analysis tool for functional genomics," *Nucleic acids research*, vol. 40, no. W1, pp. W478–W483, 2012.

[19] F. Faul, E. Erdfelder, A.-G. Lang, and A. Buchner, "G* power 3: A flexible statistical power analysis program for the social, behavioral, and biomedical sciences," *Behavior research methods*, vol. 39, no. 2, pp. 175–191, 2007.

[20] P. Kluegl, M. Atzmueller, and F. Puppe, "Meta-level information extraction," in *Annual Conference on Artificial Intelligence*, pp. 233–240, Springer, 2009.

[21] P. Mahé, M. Arsac, S. Chatellier, V. Monnin, N. Perrot, S. Mailler, V. Girard, M. Ramjeet, J. Surre, B. Lacroix, A. van Belkum, and J.-B. Veyrieras, "Automatic identification of mixed bacterial species fingerprints in a maldi-tof mass-spectrum," *Bioinformatics*, vol. 30, no. 9, pp. 1280–1286, 2014.

[22] M. T. Madigan, J. M. Martinko, J. Parker, *et al.*, *Brock biology of microorganisms*, vol. 13. Pearson, 2017.

[23] P. Diaconis and B. Efron, "Computer-intensive methods in statistics," *Scientific American*, vol. 248, no. 5, pp. 116–131, 1983.

[24] B. Cestnik, "Kononenkoj., bratkoj.(1987): Assistant-86: A knowledge elicitation tool for sophisticated users," *Progress in machine learning*.

[25] A. Agresti, *Categorical data analysis*, vol. 482. John Wiley & Sons, 2003.

[26] P. Congdon, *Bayesian models for categorical data*. John Wiley & Sons, 2005.

[27] S. R. Safavian and D. Landgrebe, "A survey of decision tree classifier methodology," *IEEE transactions on systems, man, and cybernetics*, vol. 21, no. 3, pp. 660–674, 1991.

[28] A. Edwards, G. Elwyn, and A. Mulley, "Explaining risks: turning numerical data into meaningful pictures," *Bmj*, vol. 324, no. 7341, pp. 827–830, 2002.

[29] A. Gelman and J. Hill, *Data analysis using regression and multilevel/hierarchical models*. Cambridge university press, 2006.

[30] J. Han, J. Pei, and M. Kamber, *Data mining: concepts and techniques*. Elsevier, 2011.

[31] C. Guo and F. Berkhahn, "Entity embeddings of categorical variables," *arXiv preprint arXiv:1604.06737*, 2016.

[32] D. T. Larose and C. D. Larose, *Discovering knowledge in data: an introduction to data mining*. John Wiley & Sons, 2014.

[33] L. Al Shalabi and Z. Shaaban, "Normalization as a preprocessing engine for data mining and the approach of preference matrix," in *Dependability of Computer Systems, 2006. DepCoS-RELCOMEX'06. International Conference on*, pp. 207–214, IEEE, 2006.

[34] S. Patro and K. K. Sahu, "Normalization: A preprocessing stage," *arXiv preprint arXiv:1503.06462*, 2015.

[35] N. J.-M. Blackman and J. J. Koval, "Interval estimation for cohen's kappa as a measure of agreement," *Statistics in medicine*, vol. 19, no. 5, pp. 723–741, 2000.

[36] H. Peng, F. Long, and C. Ding, "Feature selection based on mutual information criteria of max-dependency, max-relevance, and min-redundancy," *IEEE Transactions on pattern analysis and machine intelligence*, vol. 27, no. 8, pp. 1226–1238, 2005.

[37] L. Rokach and O. Maimon, "Clustering methods," in *Data mining and knowledge discovery handbook*, pp. 321–352, Springer, 2005.

[38] C. Castiello, G. Castellano, and A. M. Fanelli, "Meta-data: Characterization of input features for meta-learning," in *International Conference on Modeling Decisions for Artificial Intelligence*, pp. 457–468, Springer, 2005.

[39] R. Vilalta and Y. Drissi, "A perspective view and survey of meta-learning," *Artificial Intelligence Review*, vol. 18, no. 2, pp. 77–95, 2002.

[40] N. Bourbakis, A. Esposito, and D. Kavraki, "Extracting and associating meta-features for understanding people's emotional behaviour: Face and speech," *Cognitive Computation*, vol. 3, no. 3, pp. 436–448, 2011.

[41] S. Wold, K. Esbensen, and P. Geladi, "Principal component analysis," *Chemometrics and intelligent laboratory systems*, vol. 2, no. 1-3, pp. 37–52, 1987.

[42] L. Yu and H. Liu, "Feature selection for high-dimensional data: A fast correlation-based filter solution," in *Proceedings of the 20th international conference on machine learning (ICML-03)*, pp. 856–863, 2003.

[43] M. A. Hall, "Correlation-based feature selection of discrete and numeric class machine learning," 2000.

[44] C. Chow and C. Liu, "Approximating discrete probability distributions with dependence trees," *IEEE transactions on Information Theory*, vol. 14, no. 3, pp. 462–467, 1968.

[45] B. Pfahringer, H. Bensusan, and C. G. Giraud-Carrier, "Meta-learning by landmarking various learning algorithms.," in *ICML*, pp. 743–750, 2000.

A Meta-features

Meta-features are values gathered out of the dataset. These values can be computed easily with counting aspects or can be found by doing a more extensive analysis. These meta-features show valuable information that can be used for better insights in the dataset and can be used further on in analyses.

A.1 Basic Meta-features

The available dataset at the start of a project can have several different important aspects that need to be taken into account. Examples are the size of the dataset and the distribution of the values in it. These dataset aspects are called meta-features[20]. These meta-features are used in multiple analysis techniques and are an important aspect in meta-learning[20, 38, 39, 40]. These meta-features are also usable without meta-learning however, as they can help by a global exploration of the data. With this exploration directions for preprocessing and analysis can be found, aiding the data analyst in his research. Five different basic meta-feature types are discussed:

- **Data size**
  Finding out the size of the data shows very quickly whether possible limitations or challenges are present. The data size is split into the number of features and the number of samples.

- **Feature types**
  There are two main types of features that can be used in a dataset, categorical features and numeric features. Further sub-categorization of the feature types is possible, but is harder to detect. An example of this would be ordinal. Ordinal is categorical data with an order in the categories. In practice, ordinal features either have categorical values with given ordering or numeric values losing the names and are treated as the type they are given.

- **Missing values**
  Missing values may be present in the dataset and need some type of processing. This processing can be done in multiple ways which depend on the ratio of missing values as well as the distribution of them over features and samples.

- **Output type**
  The output of the dataset can be both categorical and numeric. The type of output changes the choice between using classification and regression respectively. For categorical data, the distribution of the separate categorical classes and whether there are majority and minority classes.

- **Feature cardinalities**
  Cardinalities shows the number of unique options available, as there is a significant difference between features with only 2 and with as much as 100 different categories. The same holds for numeric data as with low cardinality the data may be less spread out.

A.2 Statistical Meta-features

The dataset can also be approached in a statistical manner. The feature values usually follow a certain distribution. Also similarities between features are worth being looked into. Three statistical meta-feature types are discussed:

- **Feature distributions**
  The distribution of numeric features is usually shown by using four values, the mean, standard deviation, skewness and kurtosis. The combination of these four indicate distributions that afterwards can be compared for finding difference between distributions

- **Principal component analysis**
  An effective way of reducing the number of features by changing them is principal component analysis (PCA). PCA collects the data of all possibly correlated features and creates new uncorrelated features called principal components. To reduce the number of features as much as possible, these principal components are created one by one, giving as much variability to a new principal component as much as possible. Therefore the ratio of total explained variability of the first few principal components can show the presence of correlation within the features and whether feature reduction can be done to lower the dimensionality. In research for which no direct link is needed between input and output, PCA is used to significantly reduce the number of features[41].

- **Correlation**
  Whereas PCA creates new uncorrelated features to test variability between them, also a direct look at the correlation can be done by comparing all features with each other[42] (Equation 6). For a high number of features, this can be computationally very exhaustive, however it is a fairly easy way to spot multicollinearity in a dataset[43].

$$r(x,y) = \frac{\sum_i (x_i - \bar{x}_i)(y_i - \bar{y}_i)}{\sum_i (x_i - \bar{x}_i) \sum_i (y_i - \bar{y}_i)} \tag{6}$$

A.3 Information-theoretic Meta-features

Information-theoretic meta-features are focused on the information present in the dataset, how useful this information is and if this information is relevant when linking it to the output. These meta-features are focused on the entropy and mutual information and on the combination of these:

- **Entropy**
  The entropy shows the potential information present in features and the output. Using entropy best works with categorical features, however binning numeric features sometimes

can show information gain, too. This information can be seen in the distribution of values over separate categories in categorical data and if the values are more spread out over the categories, the potential information in the feature is higher. This entropy is called attribute entropy is computed by using spreading ratios r_c for every category c (Equation 7). This spreading ratio r_c is the ratio r of instances that are category c.

$$\text{entropy} = -\sum_c r_c \log r_c \tag{7}$$

The higher the entropy, the more potential information is present. The use of the potential information gain can computed by combining the feature categories and the output classes into new categories and then compute the entropy. If the new so-called joint entropy is very close to the attribute entropy, the potential information can be used to predict output classes and therefore the feature is a good predictor for the output[25].

- **Mutual information**
  Mutual information is another way of matching features with the results. The relevance of using a feature to predict the output is used in the equation to compute mutual information (Equation 8)[44].

$$MI(x,y) = \sum_x \sum_y p(x,y) \log \frac{p(x,y)}{p(x)p(y)} dxdy \tag{8}$$

In this equation the probability density function p is used to find the mutual information MI between variables x and y[36]. A higher mutual information means the feature has a higher predictive power for the output. This mutual information can also be seen as an intersection between the attribute and class entropies.

- **Entropy and mutual information**
  The outcome values of entropy and mutual information can be combined for measuring ratios between signal and noise. One way to do that is by dividing the class entropy by the averaged mutual information (Equation 9). The class entropy needs a multitude of features with a certain mutual information for perfect prediction. Since mutual information is lower than the class entropy an estimation can be done for the number of features needed to explain the complete class entropy.

$$\text{equivalent number features} = \frac{\text{class entropy}}{\frac{1}{n}\sum_1^m \text{MI(x)}} \tag{9}$$

A second way of combining attribute entropy and mutual information is by finding out the noise to signal ratio. The attribute entropy of a feature is only partly interesting as not all entropy can be used to predict the output. Mutual information is the part of entropy that can be used in prediction. To find out how much the ratio between noise and information is present, these two can be combined (Equation 10).

$$\text{noise to signal ratio} = \frac{\text{attribute entropy} - \text{MI}}{\text{MI}} \tag{10}$$

A.4 Landmarking Meta-features

The quality of analysis can also be tested by performing small analyses. These small analyses can show how well it would perform on the dataset and some small conclusions can be made on them. This is called landmarking, creating very preliminary results in order to reach an outcome[45].

Useful landmarks are based around well known machine learning algorithms. Decision trees are an example of such an algorithm, as it is a well known and easy to understand approach

in machine learning. Other examples are naive bayes, k-nearest neighbour, linear discriminant analysis and logistic regression. Quick analyses can show results that can be used as outcome.

These results and therefore the landmarks are based on the quality of the machine learning algorithm. This quality can be measured in multiple ways, such as the accuracy, F1-score and Cohen's kappa.

B Meta-feature Values Hepatitis

All values for the first analysis for the dataset hepatitis are given here.

- **Basic meta-features**

  1. *Data size*
     Number of instances: 155
     Number of features: 19
     Number of classes: 2
     Dimensionality: 0.12

  2. *Data types*
     Number of numeric features: 6
     Number of categorical features: 13
     Ratio of numeric features: 0.32
     Ratio of categorical features: 0.68

  3. *Missing values*
     Number of missing values: 167
     Ratio of missing values: 0.06
     Number of instances with missing values: 75
     Ratio of instances with missing values: 0.48
     Number of features with missing values: 15
     Ratio of features with missing values: 0.79

  4. *Output type*
     Mean of class probability: 0.5
     Standard deviation of class probability: 0.42
     Minimum of class probability: 0.21
     Maximum of class probability: 0.79
     Minority class size: 32
     Majority class size: 123

  5. *Feature cardinalities*
     (Table 3)

- **Statistical Meta-features**

  1. *Feature distributions*
     (Table 4)
  2. *Principal component analysis*
     Explained variation component 1: 0.69
     Explained variation component 2: 0.26
     Explained variation component 3: 0.04

Table 3: The cardinality meta-features for the Hepatitis dataset

Cardinality	**Categorical features**	**Numeric features**
Mean	2	54.67
Standard deviation	0	24.09
Minimum	2	30
Maximum	2	85

Table 4: The numeric feature distribution of the hepatitis dataset

Value distributions	**Means**	**Standard deviations**	**Skewnesses**	**Kurtoses**
Mean	49.92	29.74	1.28	4.49
Standard deviation	42.57	29.73	1.46	6.11
Minimum	1.43	0.652	-0.12	-0.53
Maximum	105.33	89.65	3.18	14.02
First quartile	13.16	89.65	0.11	0.25
Second quartile	51.53	17.72	0.86	1.66
Third quartile	79.88	44.35	2.51	8.18

Eigenvalue component 1: 7919.97
Eigenvalue component 2: 2924.60
Eigenvalue component 3: 412.14
Determinant value: 3.48

- **Information-theoretic meta-features**

1. *Entropy*
   Class entropy: 0.51
   (Table 5)
2. *Mutual information*
   (Table 5)

Table 5: The entropies and mutual information values of the hepatitis dataset

Distributions	**Categorical features**			**Numeric features**		
	Attribute entropy	**Joint entropy**	**Mutual information**	**Attribute entropy**	**Joint entropy**	**Mutual information**
Mean	0.54	1.01	0.04	1.02	1.44	0.08
Minimum	0.33	0.81	0.00	0.60	1.05	0.01
Maximum	0.69	1.19	0.09	1.32	1.80	0.17
First quartile	0.43	0.93	0.01	0.76	1.19	0.03
Second quartile	0.51	1.01	0.03	1.16	1.50	0.6
Third quartile	0.67	1.12	0.06	1.25	1.68	0.13

3. *Entropy and mutual information*
   Equivalent number of categorical features: 13.8
   Equivalent number of numeric features: 6.42
   Categorical noise to signal ratio: 13.60

Numeric noise to signal ratio: 11.90

- **Landmarking meta-features**
  (Table 6)

Table 6: The landmarking meta-features of the hepatitis dataset

Machine learning algorithm	**Error rate**	**Cohen's kappa**
Naive bayes	0.22	0.48
k-Nearest neighbour (k = 1)	0.32	0.15
Decision stump	0.21	0.13
Random tree (depth = 1)	0.19	0.21
Random tree (depth = 2)	0.23	0.15
Random tree (depth = 3)	0.27	0.21
Linear Discriminant	0.17	0.46

C Meta-feature Values Micro-Organisms

All values for the first analysis for the micro organisms are given here.

- **Basic meta-features**

1. *Data size*
   Number of instances: 571
   Number of features: 1300
   Number of classes: 20
   Dimensionality: 2.28

2. *Data types*
   Number of numeric features: 1300
   Number of categorical features: 0
   Ratio of numeric features: 1
   Ratio of categorical features: 0.0

3. *Missing values*
   Number of missing values: 0
   Ratio of missing values: 0.0
   Number of instances with missing values: 0
   Ratio of instances with missing values: 0.0
   Number of features with missing values: 0
   Ratio of features with missing values: 0.0

4. *Output type*
   Mean of class probability: 0.05
   Standard deviation of class probability: 0.02
   Minimum of class probability: 0.02
   Maximum of class probability: 0.1
   Minority class size: 11
   Majority class size: 60

5. *Feature cardinalities*
   (Table 7)

Table 7: The cardinality meta-features for the micro organisms dataset

Cardinality	**Categorical features**	**Numeric features**
Mean	NaN	45.29
Standard deviation	NaN	44.62
Minimum	NaN	1.0
Maximum	NaN	296

- **Statistical Meta-features**

1. *Feature distributions*
   (Table 8)

Table 8: The numeric feature distribution of the micro organisms dataset

Value distributions	**Means**	**Standard deviations**	**Skewnesses**	**Kurtoses**
Mean	67437.29	343176.13	10.67	177.27
Standard deviation	191740.68	700237.17	6.95	168.76
Minimum	0.0	0.0	0.0	0.0
Maximum	2882892.51	7792765.65	23.90	571.00
First quartile	696.92	9641.40	6.58	50.64
Second quartile	13117.02	94709.23	10.15	122.68
Third quartile	50446.57	356651.90	15.51	274.29

2. *Principal component analysis*
   Explained variation component 1: 0.14
   Explained variation component 2: 0.10
   Explained variation component 3: 0.08
   Eigenvalue component 1: 112301000896299.64
   Eigenvalue component 2: 83271983089506.69
   Eigenvalue component 3: 63192806579339.62
   Determinant value: ∞

- **Information-theoretic meta-features**

1. *Entropy*
   Class entropy: 2.91
   (Table 9)
2. *Mutual information*
   (Table 9)

Table 9: The entropies and mutual information values of the micro organisms dataset

Distributions	**Categorical features**			**Numeric features**		
	Attribute entropy	**Joint entropy**	**Mutual information**	**Attribute entropy**	**Joint entropy**	**Mutual information**
Mean	NaN	NaN	NaN	0.11	2.98	0.04
Minimum	NaN	NaN	NaN	0.0	2.91	0.00
Maximum	NaN	NaN	NaN	0.83	3.49	0.27
First quartile	NaN	NaN	NaN	0.03	2.93	0.01
Second quartile	NaN	NaN	NaN	0.08	2.96	0.03
Third quartile	NaN	NaN	NaN	0.16	3.01	0.06

3. *Entropy and mutual information*
   Equivalent number of categorical features: NaN
   Equivalent number of numeric features: 70.27
   Categorical noise to signal ratio: NaN
   Numeric noise to signal ratio: 1.64

- **Landmarking meta-features**
  (Table 10)

Table 10: The landmarking meta-features of the micro organisms dataset

Machine learning algorithm	**Error rate**	**Cohen's kappa**
Naive bayes	0.20	0.79
k-Nearest neighbour (k = 1)	0.34	0.64
Decision stump	0.83	0.08
Random tree (depth = 1)	0.87	0.05
Random tree (depth = 2)	0.78	0.14
Random tree (depth = 3)	0.71	0.22
Linear Discriminant	0.31	0.67