Assignment 7: Supervised Learning

Danny Rogaar (s2393344)
Daan Opheikens (s3038416)
Panagiotis Giagkoulas (s3423883)
Saim Eser Comak (s3432548)
Carlos Huerta (s3743071)
Emile Muller (s3787915)

Group 14

October 22, 2018

1 Introduction: Descriptive and Exploratory Analysis - 25P

The task at hand is a classification of patients into two classes, "healthy" or "AML patient". In order to perform this task, it would be proper to first get a proper view on our data. Therefore we perform an initial descriptive and exploratory analysis on our dataset hereafter. The paper aims to classify the patients as best as possible and to that end, an ensemble of K-Nearest Neighbour (KNN) classification and Decision Trees (DT) is constructed. For both classification methods, parameters are justified by search and experimentation. A final classification is performed by the KNN/DT ensemble and is compared to a Random Forest.

1.1 A - Investigate the features - 5P

After applying eigen value decomposition, we can get the contribution of each principal component to the explanation of the variance. Based on fig.2 we can see that if we intend to explain 80% of the variance or more, we need at least 15 principal components. As for the ANOVA, using a 0.01 cut-point for statistical significance, we can see on fig.1 that we can have up to 60 significant features to solve the problem.

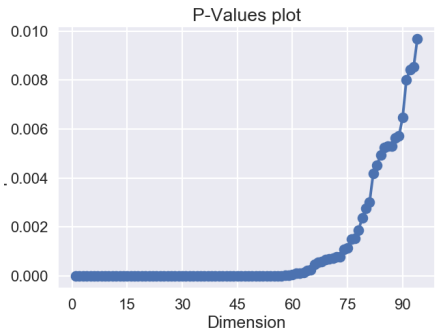

Figure 1: P-values plot

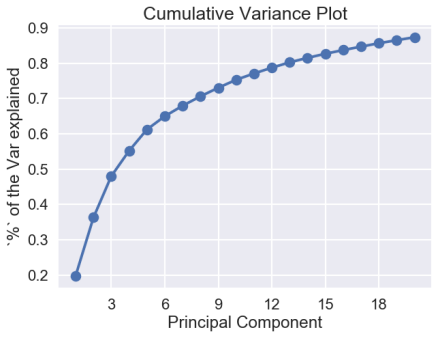

Figure 2: Variance explained with regards to principal components

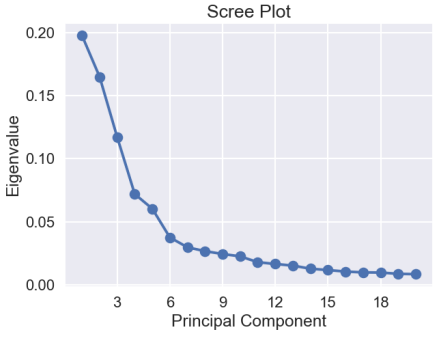

Figure 3: Eigen values of principal components

Fig.3 shows that there are a few components that seem considerably more significant than others, based on their eigen value. These principal components/features can be used to explain a significant amount of the data variance. However we should take into account that less significant features have to be considered as well. Otherwise, as previously explained, we won't be able to reach high levels of variance explanation.

In order to gain some insight into the distributions of some features, boxplots are shown in figure 4. The plots visualise the value distributions after scaling and includes the best three features, three worst but significant features (according to $p < 0.01$) and three random insignificant features. The figure shows no clear trend relating to feature performance. The distributions do not clearly in- or decrease, and outliers are present for all features. The feature distributions, however, do have different tails in their distributions. For example, the best feature, 39, has a distribution tending to higher values whereas feature 13 tends to the lower values.

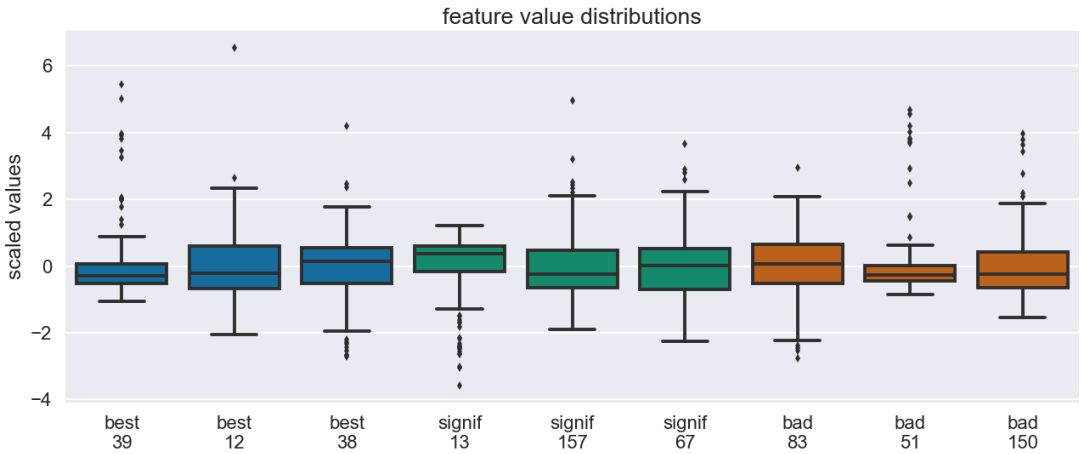

Figure 4: Boxplot showing the feature value distributions for the top 3 features in blue (as measured using ANOVA), 3 features with $p > 0.01$ in green and 3 randomly chosen insignificant features (brown).

1.2 B - Get a holistic view on the data - 5P

By applying PCA on our dataset and selecting the top 2 PCAs, we get the representation of fig.5. As we can see the "AML" patients (marked yellow) are spread out but are mainly focused on one area of the graph. We could actually linearly split the graph into two areas, separating the "healthy" from the "AML" patients. However there are multiple "AML" patients deep within the spread of the "healthy" class and a couple the other way around. The fact that PCA doesn't respect the different classes when different features are combined, might be an explanation for that.
Moving to t-SNE in fig.6 , we can see a clearer separation of the two classes, with the "healthy" being primarily on the center and the "AML" in clusters in the perimeter. This give us indications that the relationships between our data and the class label 'cancer' are not linear and therefore the t-SNE provides a more representative view of their distribution. Additionally we can see a couple "AML" objects within the "healthy" ones. Because t-SNE is more respectful towards the classes, it is safer to say that these points are possible outliers.

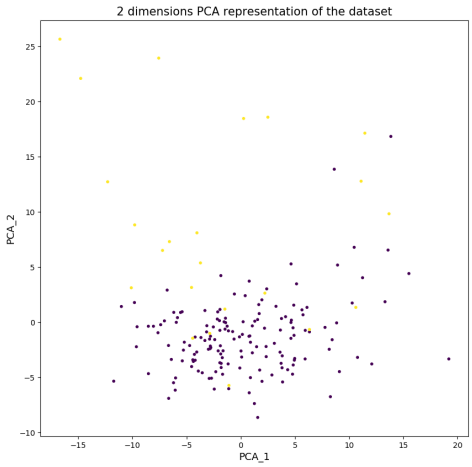

Figure 5: PCA in two dimensions

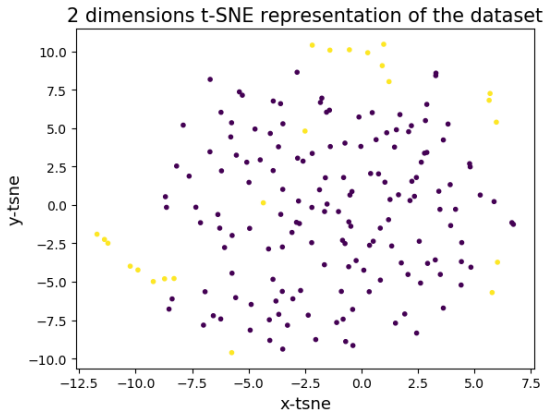

Figure 6: t-SNE plot

1.3 C - General impact of preprocessing - 15P

We applied four different methods of pre-processing on the data, namely standard and min-max scaling, PCA and ANOVA. The four figures 8-11 in the appendix display the resulting t-SNE embedding representation after when each method was used.
First and foremost, it is obvious that applying pre-processing improves the separation of the two classes on the tSNE graphs. Evidence is the comparison of fig.6 with the figs.8-11, where the "AML" class (yellow points) are more clearly separated from the "healthy" class (blue points).
We can also observe differences between these four methods. Between the PCA and ANOVA grahps, fig.10 and 11 respectively, no differences can be observed. The "AML" class is clearly separated and only 4 objects are within the "healthy" class, which could be considered as outliers. Next, we can observe considerable differences between the two

scaling methods. The standard scaling led to a clear separation of the 2 classes with the "AML" on the left and the "healthy" on the right, as seen in fig.9. However we can observe 6 "AML" points mixed in the "healthy" class.The min-max scaling also produced a clear separation but split the "AML" in two different sides (fig.8). However only two "AML" points are positioned within the "healthy" class, pointing to a better separation of the 2 classes. As a conclusion and also based on the previous analysis, the best choices would be the min-max scaling and the ANOVA pre-processing methods, as they seem to produce the most positive results.

Methods

2 Best Prediction- 75P

2.1 A - Base-line experiments - 25P

2.1.1 K-NN

Different classifiers have different hyper parameters that affect their performance. The process of determining the optimal parameters can be computationally expensive and time consuming because the classifiers need to be trained multiple times. In this case we will use the N-fold cross validation method to determine the hyper parameters of two classifiers: K-Nearest Neighbors(K-NN) and Decision Trees.

We implement the following pipeline to determine the hyper parameters. First we do some pre-processing on our data to be able to be loaded into the python pandas library as a dataframe object from the raw csv files. Then we determined descriptive characteristics of our data in our previous experiments, ANOVA, t-SNE and PCA for a first evaluation of what could be a good scaling method, hyperparameter starting point, and some basic feature selection. We observed that a class imbalance was shown in our data: the "healthy" class in our data outnumbers the "AML patients", we need to apply class balancing on our test data, to train the classifier properly. We chose the Synthetic Minority Over-sampling Technique (SMOTE) approach to upsample the minority class. This is a method that can benefit the classification of high-dimensional data, especially in the case of K-NN if a feature selection process has been carried out first [1]. Then we split our data into 70% train and 30% test data, this is done to get a correct evaluation metrics after we are done calibrating our models. Then for the case of K-NN we applied scaling to our data using only information of the training set and finally we apply N-fold validation to determine the optimal K and then evaluate the resulting model by predicting our unlabeled dataset using our 'never seen' test-set.

The following tables are the confusion matrices generated by testing the K-NN model with different pre-processing techniques on the testing and the validation subsets. It is highly probable that using K-NN for $k = 1$ leads to highly overfitted results, to the point that any pre-process we applied leads to the same results. Applying on samples taken with different seeds might change that.

		Predicted	
		0	1
Actual	0	44	3
	1	0	47

Table 1: Confusion Matrix of standard scaled K-NN model

Precision	Recall	F-score
0.94	1	0.97

Table 2: Precision, Recall and F-Score of standard scaled K-NN model

		Predicted	
		0	1
Actual	0	44	3
	1	0	47

Table 3: Confusion Matrix of min-max scaled K-NN model

Precision	Recall	F-score
0.94	1	0.97

Table 4: Precision, Recall and F-Score of min-max scaled K-NN model

		Predicted	
		0	1
Actual	0	45	2
	1	0	47

Table 5: Confusion Matrix of standard scaled K-NN model and ANOVA

Precision	Recall	F-score
0.96	1	0.98

Table 6: Precision, Recall and F-Score of standard scaled scaled K-NN model and ANOVA

		Predicted	
		0	1
Actual	0	44	3
	1	0	47

Table 7: Confusion Matrix of min-max scaled K-NN model ANOVA

Precision	Recall	F-score
0.94	1	0.97

Table 8: Precision, Recall and F-Score of min-max scaled K-NN model ANOVA

Based on the analysis conducted and evaluation of the metrics, we determined that min-max scaling and ANOVA where the best performing methods. See also misclassification and accuracy of the models, as seen in figs.12-15 in the appendix. (Standard scaling plots can be found in the Appendix, figs.16-19)
First thing we can observe regarding the k-nn is through mean performance during the cross validation, $k = 1$ is favoured as the optimal hyper parameter (figs.12-13). However as it is mentioned in [2], choosing $k = 1$ reduces the bias significantly but increases the variance of the classifier. This is the case because the classification of a point is only dependent on its closest neighbour, making the model dependent on the current subset of points and overfitting to it. Should there be a shuffle of the points or a subset with a different distribution of points, the model would display high variance. Therefore for the next steps of our experiment we will choose as optimal hyper parameter the next lowest point as shown in fig.13 and 15, namely $k = 3$ or $k = 4$.

2.1.2 Decision Trees

In the case of building decision tree model, we followed similar strategy along the pipeline which was laid out at the start of the second paragraph in the K-NN subsection. However, the scaling step was applied to the whole data instead of just the training set as the scale of the predictor variables does not change the relativity of the measures. Meaning that the split will still be at the same point regardless of the scaling. For the feature selection ANOVA was used to select 134 features out of 186 total features. For the hyper parameter estimation and cross fold validation, we used RandomizedSearchCV function from scikit-learn library to estimate parameters for our decision tree model. The function creates specified number of folds(n=10 in our case) in which it tests a random integer provided by a distribution for each parameter for each iteration. The function was iterated for 100000 times as a greedy search algorithm. The estimated parameters were *max depth* which decides where to prune the decision tree, *min samples leaf* which determines the least amount of samples that can be in the leaf nodes, *max leaf nodes* which sets the value for maximum number of leaf nodes based on the best impurity decrease scenario, *min samples split* which decides the minimum number of samples required to split an internal node. As an impurity metric, we used gini impurity criterion.

Based on the search, the function returned optimized values of the following max depth=20, min samples split= 93, max leaf nodes=43, min samples leaf=2. The search algorithm first tests a tree with the max depth parameter and max leaf nodes and then it further branches out interior nodes into leaf nodes, comparing the impurity decrease between having an interior versus a single leaf node, thus it avoids over fitting. Using the provided parameter values, we then built the model. Over fitting was not observed and the test data prediction was 100% while the number of cancer patients detected was 22 when the model used to predict the labels of the unlabeled data. Confusion matrix can be found in table 9 and decision tree representation can be seen in figure 7

2.2 B - Ensemble -25P

For the ensemble method, we have chosen to use the majority vote approach. For the classifiers, we chose two; K-Nearest Neighbour and Decision Trees. Both these methods have been used with different parameters. First, we used the 'optimal' parameters which we had calculated in the previous exercise. Then, we used semi-random parameters. The 2 optimal and 2 sub-optimal Decision Tree and KNN classifiers were combined in a single ensemble. Finally, we created an optimal and sub-optimal Random Forest classifier to compare our ensemble with. The parameters are in table 10.

Table 9: Confusion matrix for the test data

		Predicted	
		0	1
Actual	0	49	0
	1	0	5

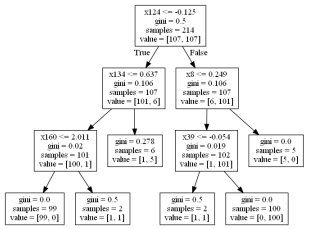

Figure 7: Decision tree representation based on the training data

Table 10: Parameters used for the KNN/Decision Tree, and the Random Forest ensembles.

Classifier	**Optimal parameters**	**Semi-random parameters**
K-Nearest Neighbour	K=1	K=2
Decision Trees	max_depth=20 min_samples_split= 93 max_leaf_nodes=43 min_samples_leaf=2 random_state=25	max_depth=10 min_samples_split= 50 max_leaf_nodes=30 min_samples_leaf=2 random_state=40
Random Forest	See optimal DT	See semi-random DT

These methods are then all trained and fitted to 70% of the labeled data. Finally, the decision trees and KNN classifiers are combined into an ensemble. The evaluation happens on the remaining 30% validation data. We then compared the accuracy of our ensemble classifier against the Random Forest classifier. The accuracy scores were as follows:

Classifier	**Accuracy score**
Ensemble Classifier	100%
Random Forest (optimal)	96.3%
Random Forest (semi-random)	100%

Achieving such high accuracy indicates that cross-validation might give more information about the performance of algorithms. That is, the high performance might be due to the dataset split and initialisation.

3 Discussion: Summarisation of your experiments - 25P

For a rough diagram of our experiments and methods, please see fig. 20 in the appendix. Focusing on features in the data, we found a number of significant features and were able to use a significance threshold of 0.01. Additionally embedding the data in 2D using both PCA and t-SNE with different scaling methods, allowed us to determine that a non-linear relationship with the data and the predictive class 'cancer' was more likely than a linear one.
Validation accuracy was calculated on a KNN classifier depending on the scaling method, determining min-max scaling to be most effective. First, however, we found the 'AML' class to be underrepresented in the data and upsampled this class in the training phase. These experiments also showed ANOVA to be most useful for reducing the amount of features. Running similar tests for different values of k, using only 1 neighbour is found to be the superior setup for KNN.
Given the higher number of parameters, we opted for an alternative to setting up experiments for them. Instead a randomized search was performed.
An ensemble using KNN and Decision trees with the best settings as well as a pair using different settings. The ensemble classifier scored as well as a Random Forest using the best settings for decision trees. The KNN/DT ensemble was used to classify the unlabeled data. The results are included outside of the report in *Team_14_prediction.csv*.

References

[1] Rok Blagus and Lara Lusa. Smote for high-dimensional class-imbalanced data. *BMC Bioinformatics*, 14(1):106, Mar 2013.

[2] Trevor Hastie, Robert Tibshirani, and Jerome Friedman. *The Elements of Statistical Learning*. Springer Series in Statistics. Springer New York Inc., New York, NY, USA, 2001.

4 Appendix

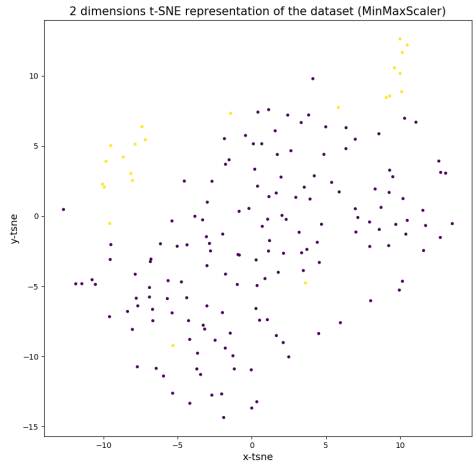

Figure 8: t-SNE after applying min-max scaling

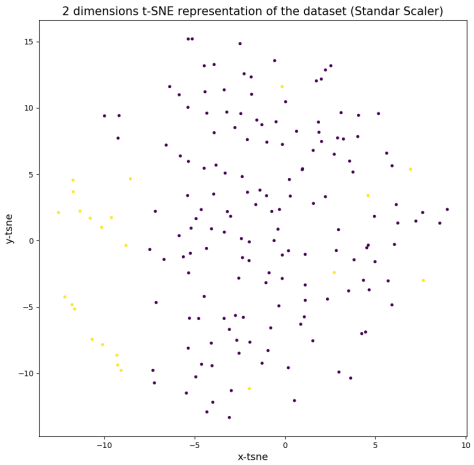

Figure 9: t-SNE after applying standard scaling

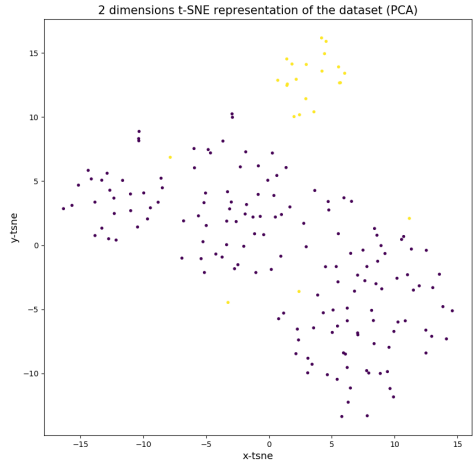

Figure 10: t-SNE after applying PCA

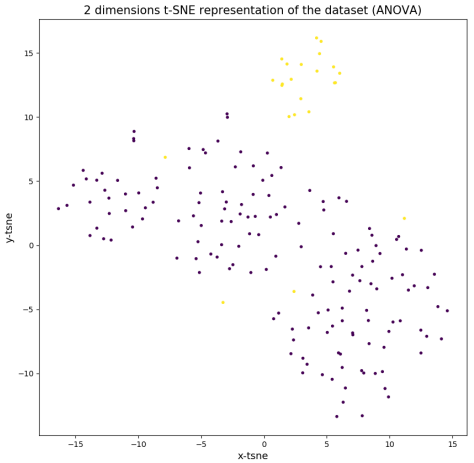

Figure 11: t-SNE after applying ANOVA

Figure 18: Accuracy with standard scaling

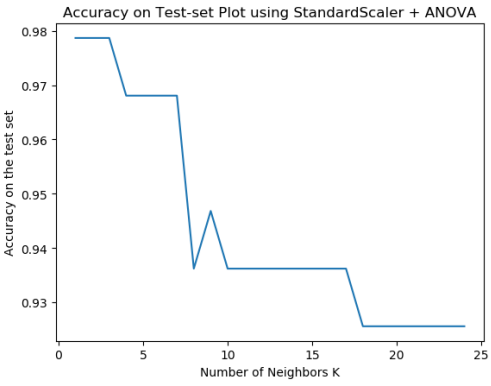

Figure 19: Accuracy with standard scaling and ANOVA

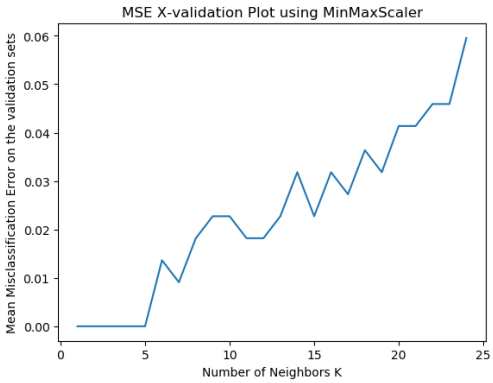

Figure 12: Misclassification error with min-max scaling

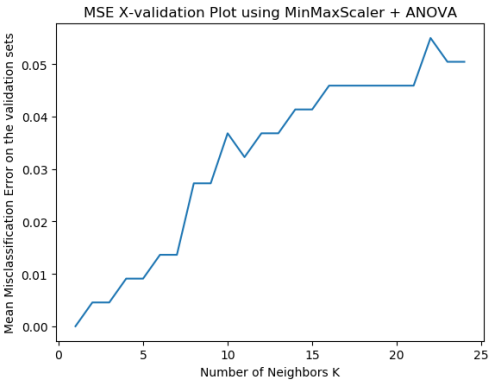

Figure 13: Misclassification error with min-max scaling and ANOVA

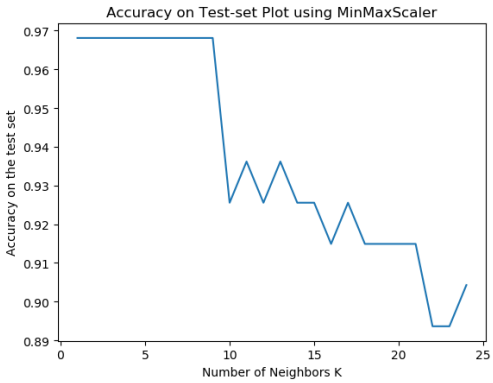

Figure 14: Accuracy with min-max scaling

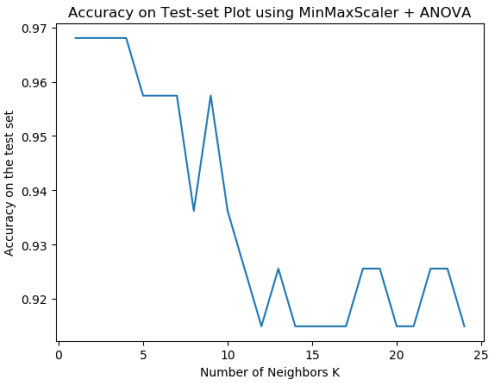

Figure 15: Accuracy with min-max scaling and ANOVA

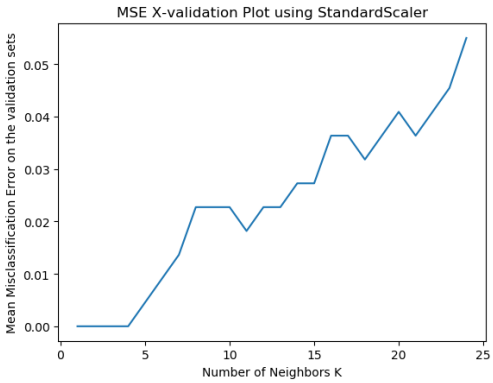

Figure 16: Misclassification error with standard scaling and ANOVA

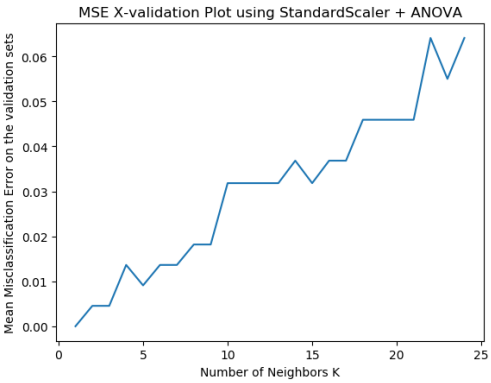

Figure 17: Misclassification error with standard scaling and ANOVA

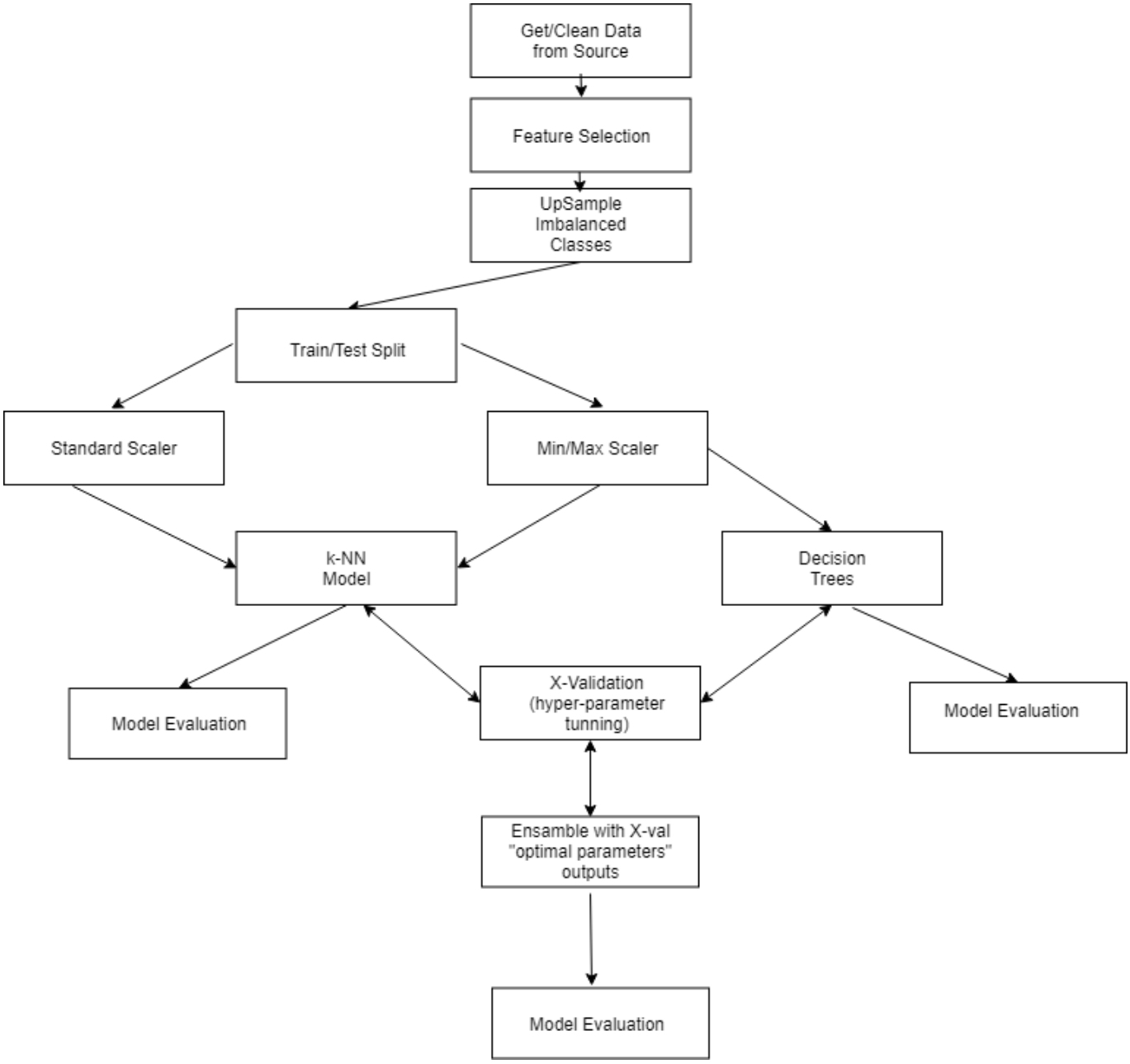

Figure 20: Modeling Diagram