TM10007: Machine Learning Assignment

Mitchel Molenaar and Siri van der Meijden

March-April 2020

1 Introduction

Head and neck cancer is an worldwide problem with increasing incidence and health costs. Currently, the incidence and mortality rate of head and neck cancer is 650.000 with high mortality rates [1]. The diagnosis and treatment of these patients is putting pressure on the healthcare system, which emphasize to need to optimize the clinical workflow.

The clinical symptoms, treatment strategy and cure-rate depend on the anatomical location and the stage of the disease. The diagnosis is determined by a combination of modalities. Computed tomography with contrast or magnetic resonance imaging are mostly applied to evaluate the location and size of the tumor. The definitive diagnosis and tumor staging (stage of disease) is determined by the molecular profile of a biopsy [2]. However, a biopsy is invasive, painful and can result in metastasis [3, 4]. Previous studies have suggested that artificial intelligence algorithms applied on features extracted from computed tomography (CT) scans could predict the tumor stage in patients with head and neck cancer [5]. Therefore, the aim of this article was to find the optimal classifier to predict the tumor stage in patients with head and neck cancer based on features extracted from CT scans.

2 Methods

2.1 Description of the dataset

The dataset consists of 160 features extracted from 113 head and neck cancer CT scans. All features in the dataset are numerical. There are patients in the dataset with two tumor stages: T12 (low stage) and T34 (high stage). The dataset is balanced in terms of outcome labels. 58 patients were classified as T12 and 55 patients as T34.

2.2 Preprocessing

First, the data was split into a 80% train and 20% testing dataset.The data was split in a stratified manner to ensure that the train and test set have approximately the same percentage of samples of both classes (T12 and T34). The train set was used for model selection, hyper parameter tuning and model evaluation. The test set was used to assess final performance metrics on an unseen part of the dataset.

2.2.1 Missing data, scaling and feature selection

There was no a prior knowledge about the interdependence of features and relationships between the features and the classes. Three patients in the training data set had a substantial amount of missing features (14, 16 and 17) respectively. The other samples had a maximum of seven missing features. The number of samples is relative small in this dataset, therefore the choice was made to include all the patients to train the model. When assessing the missing values for each feature, six features had 40 or more missing values. Other features had zero, three or five missing values. Due to the relative large amount of features, the six features with 40 or more missing values were chosen to be excluded from the dataset. Besides from the six features that had a large amount of missing values, features with zero variance were excluded from the dataset. This was true for four of the features resulting in a total of 10 excluded features. A 2NN-imputation method was used for the remaining missing values. K-nearest neighbor (KNN) imputation was chosen because it can handle different types of missing data. The data was scaled using a standardscaler which removes the mean and scales the dataset with a unit variance. All feature extraction methods, scaling and the imputation strategy were first fitted on the training data and afterwards applied on the test data.

To avoid the curse of dimensionality, in which the number of features is larger than the number of samples ($N_{features} > N_{samples}$), we further selected features by two methods: (1) Principal component analysis (PCA) to reduce dimensionality of the data or (2) regularization by least absolute shrinkage and selection operator (LASSO).

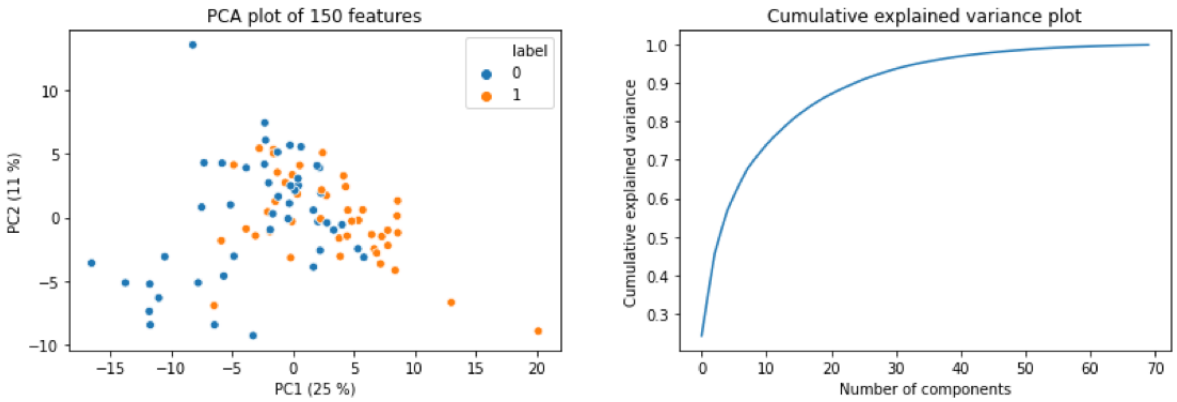

Figure 1: *Left: first two principal components, from which can be seen that the data cannot be linearly separated based on these two components. Right: Cumulative explained variance plot. The first 27 principal components explain approximately 90% of the cumulative variance, indicating complex data.*

PCA was performed to search for features with the largest variation between patients. From the cumulative variance plot we saw that circa 27 principal components were required to explain >90% of the cumulative variance (see figure 1). Therefore, The first 27 principal components were selected when using PCA as feature selection method. Besides feature selection, LASSO was applied to avoid overfitting on the trainingset. LASSO resulted in a useful set of 68 features.

2.3 Classifiers

Because most variance is not explained by the the first two principal components (see figure 1), it is not possible to use simple linear predictors. Therefore, we hypothesize that is necessary to look into more complex relations between features. To test this hypothesis, we applied a KNN classifier. In addition, a classifier is required that works well with multidimensional feature spaces and generalizes well. Therefore, Support vector machine (SVM) and random forests (RF) were also used as classifiers.

2.4 Experimental and evaluation set up

We divided the training set (training set 1) into a training set (training set 2) and validation set. The classifiers, with feature selection method and tested hyperparameters, are shown in Table 1. Not all hyperparameters were tuned; a limited range was chosen to avoid long computation times and prevent overfitting on the trainingset. All steps are shown in Figure 1. To choose the optimal hyperparameters for the classifiers, we applied a randomized search cross-validation on training set 2 and tested the best hyperparameters on the validation set. A Python pipeline was used including the imputation, scaling, feature selection method and randomized search for the three different models. The randomized search included 30 iterations for each of the different pipelines. Feature elimination as described in the previous sections was performed before the data entered the pipeline.

The three models (classifier, with best feature selection method and best hyperparameters) with high area under receiver operating characteristic curve (AUC) on training and validation set were selected. It was believed that both a high AUC on training set 2 and validation set means that the model generalized well. The three models were evaluated on training set 1 by 10-fold cross-validation to obtain the mean AUC with standard deviation (STD). The standard deviation shows the amount of variance in the performance. The model with the highest mean AUC and lowest STD was the final model of choice. However, all three models were evaluated on the test set.

Table 1: *Classifiers with tested hyperparameters. k = number of nearest neighbors, PC = principal components, RBF = radial basis function, C = value that controls amount of slack, Criterion = function to measure the quality of a split, Trees = the number of trees in the forest, Depth = the maximum depth of the tree, Split = the minimum number of samples required to split an internal node.*

Classifier	**Feature selection**	**k**	**PC**	**Kernel**	**C**	**Criterion**	**Trees**	**Depth**	**Split**
KNN	PCA	k = 5-40 (1)	27						
KNN	LASSO	k = 5-40 (1)							
SVM	PCA		27	Linear or RBF	1-10 (1)				
SVM	LASSO			Linear or RBF	1-10 (1)				
RF	PCA		27			gini or entropy	10 - 100 (10)	1-10 (1)	2-10 (1)
RF	LASSO					gini or entropy	10 - 100 (10)	1-10 (1)	2-10 (1)

2.5 Statistics

Performance of the three models was evaluated on the test set using recall, precision, f1-scores, AUC and accuracy. The metric AUC was chosen for model selection because it represents both the sensitivity and specificity, and therefore evaluates the discriminating properties of the model.

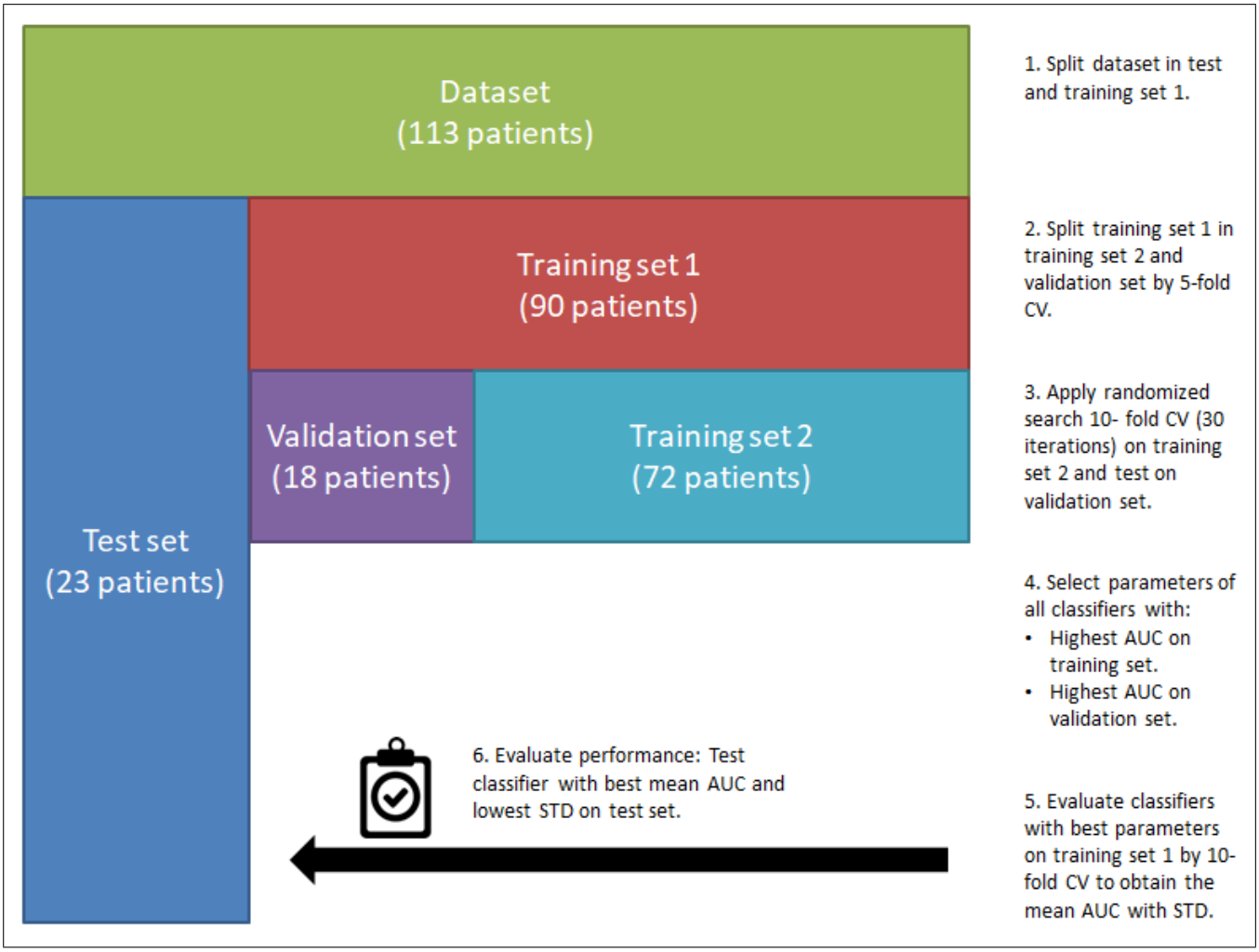

Figure 2: *Experimental and evaluation set up. CV = cross-validation. AUC = area under receiver operating characteristic curve, STD = standard deviation.*

3 Results

3.1 Chosen classifiers using Randomized search

Based on the randomized search set-up for hyper-parameter optimization, for each of the three classifiers (KNN, RF and SVM), a 'best' model was chosen. This best model showed both a good training and validation AUC. The chosen classifiers, hyper-parameter settings, training AUC and validation AUC are shown in Table 2. Afterwards, the three classifiers were trained and evaluated over the whole training dataset using 10-fold cross-validation. The SVM classifier with PCA, a radial basis function kernel and slack parameter (C) with value 6 was chosen to be the model of choice, having the highest mean AUC (0.88) and lowest standard deviation (0.12). Mean AUC (STD) was 0.82 (0.12) for the RF and 0.81 (0.12) for the KNN classifier. The AUC showed a high variance over the different folds in the training set for all three classifiers.

3.2 Final evaluation of classifiers on independent test set

Performance metrics were evaluated on the test set to evaluate the performance on unseen data. SVM was the model of choice from the previous steps. However, the SVM algorithm showed lower performance compared to KNN and RF on the independent test set. The performance metrics of the three classifiers are shown in Table 3. The

KNN-classifier showed to have the best overall performance on the test set. When assessing the confusion matrices for all three classifiers (see Figure 3), all three models showed to have a high recall. This means that high stage patients were likely to be classified correctly. However, there was a larger amount of false positives for all three classifiers, indicating low specificity.

Table 2: *Chosen hyper parameters using randomized cross-validation for each of the three models. Abbrevations are shown in Table 1*

Classifier	**Feature selection**	**Hyper-parameters**	**Training AUC**	**Validation AUC**
SVM	*PCA*	*Kernel: 'RBF'* *C: 6*	*0.89*	*0.81*
RF	Lasso	Trees: 80 Criterion: 'gini' Split: 4 Depth: 5	0.86	0.78
KNN	Lasso	k: 29	0.84	0.61

Table 3: *Performance metrics on the independent test set for the three chosen classifiers*

Classifier	**AUC**	**Accuracy**	**F1-score**	**Precision**	**Recall**
SVM	0.61	0.65	*0.69*	*0.60*	*0.82*
RF	0.74	0.74	0.72	0.73	0.73
KNN	0.83	0.83	0.85	0.73	1.00

4 Discussion

The aim of this article was to find a classifier to predict the tumor stage in patients with head and neck cancer based on features extracted from CT scans. Three different machine learning architectures (KNN, RF and SVM) were assessed. After training and evaluation, the model of choice for classification was a SVM (using PCA with 27 components, a RBF kernel and a slack parameter value of 6) with a mean AUC (std) of 0.88 (0.12). Final evaluation on the test-set resulted in a low AUC of 0.61 for the SVM classifier. However, KNN and RF performed better with an AUC of respectively 0.83 and 0.74.

Strengths and limitations
The major strength of this study was that multiple models, including three different classifiers, two feature selection methods and a range of hyperparameters, were tested on the data set. This could be seen as a limitation as well, since a randomized search was performed over a limited amount of hyperparameters for each model.

Several other remarks should be made to the study. The amount of samples in the data set was small. A model build on small data sets is prone to overfitting, in which the model is not able to generalize on data that has never been presented to the model. Our best model (SVM) was also affected by overfitting, as seen in the high AUC standard deviation (0.12) when performing cross-validation on the entire trainingset. A relative large amount of samples was used for the final evaluation of the classifiers, which resulted in a limited amount of training samples for model selection and training. The performance of the SVM-classifier was surprisingly low on the test set compared to the KNN and RF classifier. This could be the result of a marginal generalization and/or a non-representative test set.

Another limitation of our study was that we assumed that all zero values could be interpreted as missing values. By removing features with a lot of features we potentially removed valuable information. More insight in the features is required to avoid deletion of valuable features. Furthermore, a 2NN imputation method was used, which is common for different sorts of missing data. However, a higher number of neighbours could result in less influence of noisy data.

Clinical relevance
The implementation of a non-invasive method to stage head and neck cancer has several advantages for both patients, clinicians as the society. The patient benefits by avoiding a painful biopsy that increases the chance on metastasis. The clinician does not have to perform a biopsy, which saves time, treatment and optimalizes the workflow. The society could benefit from reduced medical care costs.

From the results of our chosen SVM-classifier on the test set, it was clear that discriminating properties of the classifier was moderate (AUC 0.61). However, recall was 0.73, indicating that patients with high stage cancer were

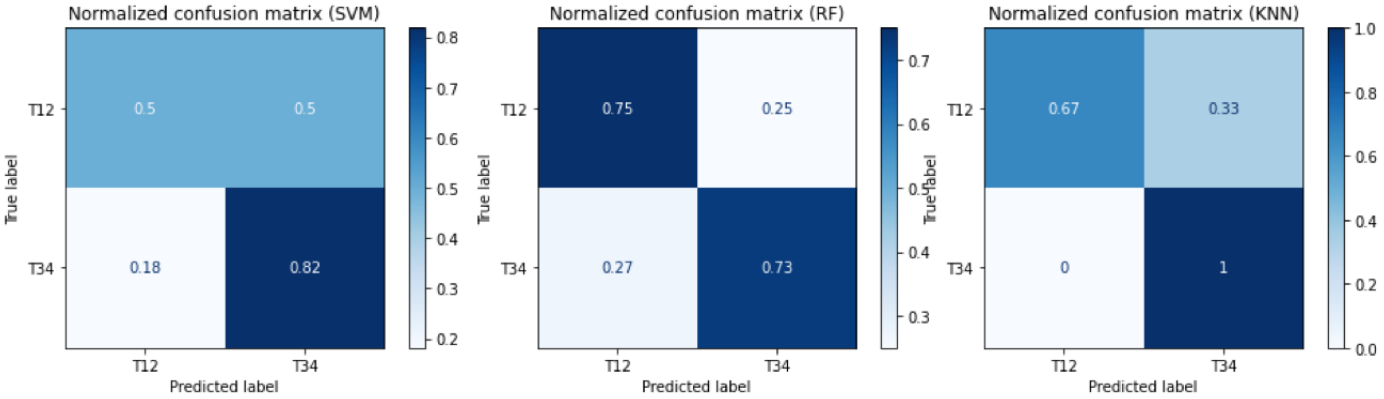

Figure 3: Confusion matrices for all three final classifiers.

likely to be detected by the algorithm. This is important in clinical practice. Since all patients are already known to have head neck cancer, a high recall is relevant to not under treat high stage patients. On the other hand, due to lower precision values, it might be that using our algorithm, low stage patients are treated to aggressively.

Future research

Future studies that focus on a machine learning model to stage the tumor grade in patiens with head and neck cancer should include more patients. More effort could be done in tuning the model and performing generalization to a higher extent to avoid overfitting on the training data. The experimental set-up could be changed in order to get a different split in train and test set before going into hyperparameter tuning. Future studies should also include a priori knowledge about relationships between the features and the classes. Based on this knowledge important features could be selected in advance.

Conclusion

In summary, this study shows that three classifiers (KNN, RF and SVM) could be trained with a mean AUC > 0.75 for classification of tumor stages in patients with head and neck cancer, based on features extracted from CT scans. More samples are required to obtain a model with acceptable recall and specificity due to the complexity of the data. Using CT scans to stage tumors will provide clinicians a potent tool to optimize the clinical workflow in patients with head and neck cancer.

References

[1] F. Bray et al. “Global cancer statistics 2018: GLOBOCAN estimates of incidence and mortality worldwide for 36 cancers in 185 countries”. In: *CA Cancer J Clin* 68.6 (Nov. 2018), pp. 394–424.

[2] S. Marur and A. A. Forastiere. “Head and Neck Squamous Cell Carcinoma: Update on Epidemiology, Diagnosis, and Treatment”. In: *Mayo Clin. Proc.* 91.3 (Mar. 2016), pp. 386–396.

[3] K. Shyamala, H. C. Girish, and S. Murgod. “Risk of tumor cell seeding through biopsy and aspiration cytology”. In: *J Int Soc Prev Community Dent* 4.1 (Jan. 2014), pp. 5–11.

[4] S. R. Akkina et al. “The current practice of open neck mass biopsy in the diagnosis of head and neck cancer: A retrospective cohort study”. In: *Laryngoscope Investig Otolaryngol* 4.1 (Feb. 2019), pp. 57–61.

[5] R. Hermans. “Head and neck cancer: how imaging predicts treatment outcome”. In: *Cancer Imaging* 6 (Oct. 2006), S145–153.

5 Team evaluation

Planning and division of roles and tasks
First, the data was explored after which basic feature elimination was performed by Mitchel. Afterwards, different feature selection methods were assessed and tested by Siri. A pipeline was chosen for model selection and hyperparameter tuning, in which Mitchel was responsible for the SVM and Siri for the KNN and RF. Lastly, the chosen models were trained on the whole training set and tested on the independent test set. At the end, we were not completely happy with the final results of the model in terms of generalizability. Unfortunately, we had no time to improve our pipeline. More effort could have been made in choosing one or two modelling architectures, and performing a more extensive grid search for finding the optimal parameters. However, we chose to evaluate a broader range of model architectures for educational purposes. A rough outline of our planning is shown in Table 4.

Table 4: Planning of project

Planning	**Activities**
Week 1	Data exploration
Week 2	Feature selection
Week 3	Pipeline: Feature selection + classifiers
Week 4	Pipeline: Hyperparameter tuning
Week 5	Performance evaluation and writing report

Siri
From the start, Mitchel and I were dedicated to make the best out of the project. It was convenient that we both are doing our masterthesis at the moment, which made the collaboration to the project equivalent. Although we could not meet in real life, communicating with Mitchel was very efficient. We both were struggling with some concepts of the assignments, which resulted in some long and interesting discussions. Our main aim was to learn as much as possible about solving the machine learning problem and less in getting the 'best' result, and I think we succeeded in that learning purpose. In the starting phase of the project, Mitchel putted a lot of effort in data exploration and feature extraction. The pipeline for model selection and hyperparameter tuning was a joint effort and I made an effort in the evaluation of the model. For me personally, it was quite difficult to get into programming in Python after a year of Matlab programming during my internships. I was very happy with Mitchel helping me at some points in the programming process.

Mitchel
The collaboration between Siri and me was very pleasant. The short lines of communication were the key to our collaboration. Our long and interesting discussion on some topics made the project educational. We were able to balance teamwork with our own responsibilities in the project. Before finding the pipeline function by Siri, my specific individual contribution to the project was by working on data exploration and feature selection. In the last stage of the project I was working mainly on parts of the report, in which the results did not play a role, while Siri was obtaining the performance of the models. It was helpful that we both worked on the python code, understood the code both, and that we could ask each other about changes in it. A point to improve is to ask for help earlier, which I will illustrate with an example. On one afternoon we had a discussion about cross-validation that took us four hours. If we had asked for help earlier this would saved us time. However, as I mentioned, this was really educational.