Predicting Lung Cancer with Machine Learning

October 4, 2022

1 Introduction

Lung cancer is the most common form of cancer in the whole world. Over 200.000 new cases are being diagnosed every year which makes up to 13% of of all cancer diagnosis. Almost half of the patients with lungcancer die after a year after the diagnosis. As for all forms of cancer, an early diagnosis increases the survival rate drasticaly. Earlier stages of lung cancer are easier to treat and can decrease the death rate by 14% to up to 20%. Applying machine learning appears to be reasonable use case for diagnosis of lung cancer, since it can achieve high effectivnes. In this paper I will try to determine whether a patient with certain characteristics has lung cancer or not.

1.1 Report structure

The report will continue with the section 2 "Problem Statement",

2 Problem Formulation

2.1 Data set

The data set consists of 309 entries in an Excel spreadsheet, with patients between 21 and 87 years old. Each data point describes a patient with different characteristics. The dataset is licensed under the *CCO: Public Domain* and was found on *Kaggle* [1]. For transparency, this dataset is used in another project found on *here* [2], but as this work was done independently, it has not been copied in any way. The dataset is complete and has no missing features. Each data point is mapped with 16 different attributes (gender, age, smoking, yellow fingers, chronic disease, *see appendics for more...*). Additionally, I was also able to find 33 duplicates, which I have eliminated. As you may have already noticed, the dataset is not too large, now containing only 276 entries. How we dealt with this problem will be further explained in the "Data Splitting" section

2.2 Type of data

The label for this prediction is whether a patient has lung cancer or not (yes/no), which means that this problem is a classification problem. I decided to change the dataset for the labels from *yes* and *no* to *-1* and *1* to make the prediction easier and because I will use the SVC hinge loss function in the second part. All attributes except gender and age are represented with *1* or *2*, which corresponds to *no* and *yes* respectively. Gender is

represented with *M* or *F*, which means male or female respectively. The age is the actual age in the form of a number. The histogram shows that the data is largely balanced, except for a few outliers such as *fatigue* and *lung cancer*. Since I will be using logistic regression, these outliers will not significantly affect our results [3][4].

3 Methods

3.1 Feature selection and engineering

For the features, I selected all attributes except for the lung cancer column, as this is used for the label. For the selected features, I have changed the data points from *1* and *2* to *0* and *1* for convention and ease of understanding. It is important to note here that with the exception of age, all characteristics can be represented in binary, meaning, a normalisation is not required. After looking at the correlation heat map of the characteristic, I could not identify any bad data sets. All data points appear to be in the range of -0.75 and 0.75, which is a common indicator of correlation. Also, all the identifiers, gender, age, smoking, yellow fingers, (. . .) seem intuitively important in determining whether a patient has lung cancer or not. After some small tests with *Variance Inflation Factor (VIF)* I could confirm that all 15 characteristics seem to improve the accuracy of the prediction.

3.2 Data Splitting

As mentioned earlier, the data set is not very large, so I decided to use the k-fold cross-validation method. This method is suitable for data sets with fewer entries. The algorithm works by randomly dividing the datasets into equal parts and using one set as the test set and the rest as the training set. I decided to split the data into 5 parts (k=5) to achieve an 80% to 20% ratio, which is very common.

Our goal is to classify whether a patient has lung cancer. This can be described as a binary classification problem.

3.3 Logistic Regression

For the first machine learning model, I chose logistic regression because it seemed reasonable for a simple binary classification problem. Logistic regression uses a linear hypothesis space and works by setting a limiter between the data points. In a 2D space, you would put a line between the given data points. In 3D space, the points would be separated by a plane, and in higher dimensions you would use a hyperplane to describe the separation.

3.3.1 Loss function for Logistic Regression

For the loss function, I chose the logistic loss function, as it is a very common and proven function for logistic regression and was easy to use as it is already implemented in the used library.

3.4 SVC

3.4.1 TODO Insert here

3.4.2 Loss function for SVC

1. **TODO** Insert here

4 Results

5 Conclusion

6 References

6.1 TODO Lung cancer facts

`https://www.lung.org/lung-health-diseases/lung-disease-lookup/lung-cancer/resource-library/lung-cancer-fact-sheet`

- [1] Data set from Kaggle: `https://www.kaggle.com/datasets/mysarahmadbhat/lung-cancer`
- [2] Other Kaggle project with same data set: `https://www.kaggle.com/code/gaganmaahi224/lung-cancer-5ml-models-full-analysis-plotly`
- [3] How to handle unbalanced sets tutorial : `https://www.kdnuggets.com/2017/06/7-techniques-handle-imbalanced-data.html`
- [4] Unbalanced data in Logistic Regression: `https://stats.stackexchange.com/questions/6067/does-an-unbalanced-sample-matter-when-doing-logistic-regression`
- [5] Heatmap Tutorial *Medium* an seaborn library: `https://medium.com/@szabo.bibor/how-to-\\create-a-seaborn-correlation-heatmap-in-python-834c0686b88e`

7 Code Appendics