AI prediction model for Knee Arthroplasty

Computer Vision and Machine Learning

Source Code: github

Kaggle: mortenrosenquist

Morten Lyng Rosenquist
Faculty of Technical Sciences
Aarhus University
Aarhus, Denmark
201706031

April 19, 2022

Write Abstract

***Abstract*—**

***Index Terms*—**

Write keywords

Write Introduction

I. Introduction

This thesis will analyze the development of an AI prediction model in a clinical setting regarding knee arthroplasty. Based on a patients demographics, lifestyle and clinical metrics, the model is to determine a patient's risk group. Knowing a patient is at risk can improve the chance of the knee arthroplasty surviving by tailoring an individual treatment plan. The thesis is done as a project in the course Computer Vision and Machine Learning at Aarhus University. The dataset is given together with a explanation of the features and description of the problem.

cite

binary classification problem

Support Vector Machine

The model that will be used is a Support Vector Machine(SVM). It produces nonlinear boundaries by creating a linear boundary in a tranformed version of the feature space. With this functionality SVMs can be used for classification, regression and outlier detection. We will use it for classification, this is also called Support Vector Classification(SVC). In its simplest form the SVC creates an linear optimal seperating hyperplane between two seperated classes. Where classes can not be seperated by a linear boundary other kernels(polynomial, Radial Basis Function) can be utilized. There are hyper-parameters to be tuned of the SVC. The most important is the regularization parameter that determines the amount of punishment for misclassifications. Utilizing a polynomial kernel the degree of the kernel function is also to be tuned.

cite

Include support vectors

A. Imbalanced Data

Building a model with imbalanced data can lead to trouble. As the model primarily sees the majority class it might not learn enough from the minority class. This can lead to a model predicting all new observations as the majority class. This will lead to a high accuracy and recall for the majority class but a recall of 0 on the minority class. This is basically an useless model, that wont predict any of the minority class observations. In the clinical setting this means that we wont place any patients in the risk group. There are several ways to mitigate the issues with imbalanced data. The majority class can be downsampled or the minority class can be oversampled. With regards to SVC we can add class weights to penalize or reward the classification of a class.

B. Feature Selection

The dataset contains a range of features regarding the patients. Having data with a high dimensionality can lead to problems such as long training time, a complex model and overfitting. Therefore, it will prefered if we can reduce the dimensionality while still having good performance of the model. This can be done by selecting the features that contribute the most to the target variable. This is typically supervised and will keep features intact. Principal Component Analysis can also be used to reduce the amount of features. This will however transform the features based on variance. In regards to a clinical settings it would be interesting to see which features are the most valuable in terms of determining wether an patient is in the risk group.

C. Metrics

II. Methods & Results

Write method

The experiments performed is covered in the *jupyter* notebook in the source code.

Præsenter afsnit

A. Dataset

The dataset is in advance split in training, validation and test sets. The training and validation observations are labeled. The idea is to train the model on the training set and evaluate the performance on the out-of-sample observations from the validation sets.

Source code

Præsenter sklearn

B. Preprocesssing

The dataset contains two types of features - boolean (yes/no) and continous values (numeric). The dataset is not complete, meaning some observations are missing values in certain features. To overcome this the mean value is used for the numeric features and the most popular for the boolean features. As the features are in different units it is important to scale the data. Without scaling certain features might dominate the other features. This is specially the case for SVC, because it tries to maximize the distance between the support vectors and the hyperplane. *sklearn* have different options for scaling; StandardScaler, MinMaxScaler and RobustScaler. The MinMaxScaler with default settings will suffice with a scaling of each feature individually between 0 and 1 .

C. Model training

Tuning the model to perform as desired is not a trivial task and it is highly dependent on the use case. Initially, there is several questions to be answered such as: How do we train the model? Do we use the raw training data? Are we to use sampling methods? How do we test different model parameters? Which metrics are used to train/evaluate the model? Answering these differently will lead to very diverse models. Due to lack of experience, this section will cover a trial and error approach. Firstly, a model is trained with default parameters. Then we look into tuning the model in various ways. The different models are concurrently evaluated.

sklearn's SVC implementation is used to build the model. Initially we try the model with the default parameters:

- Regularization parameter (C): 1
- Kernel: Radial Basis Function (rbf)

The model is trained and evaluated. The evaluation on the validation set can be seen on Figure 1.

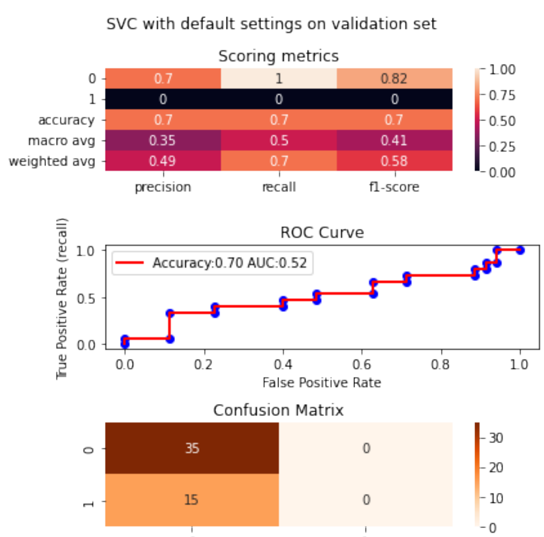

Fig. 1: Performance of SVC model with default parameters

It is seen that the model scores well in terms of precision, recall and f1-score on class *0*. However, as seen in the confusion matrix this is due to the model classifying all observations to class *0*. This is due to the concerns introduced earlier regarding imbalanced data. As a measure, we balance the weights of each class. This is done by settings the model's *class_weight* parameter to *balanced*. *class_weights* can be defined explicitly or have *sklearn* calculate them by the proportions. The validation performance can be seen on Figure 2.

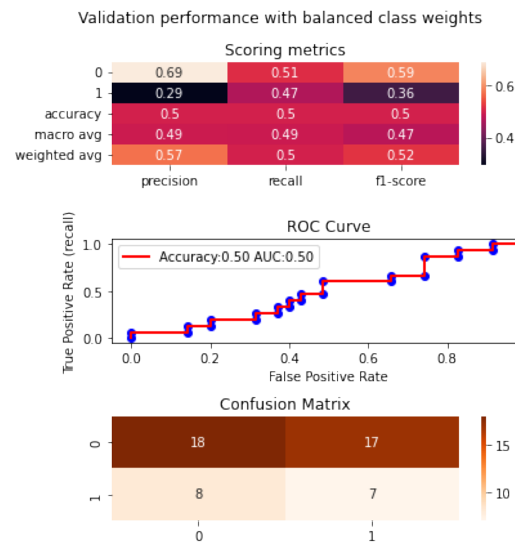

Fig. 2: Performance of SVC model with balanced class weights

The update in class weights clearly makes the model predict more observations to class 1. This leads to better scoring of class 1, but many false positives are introduced as a consequence. To mitigate these, we try to tune other parameters of the model. We do this with *sklearn GridSearchCV* implementation. Here we can define a grid of parameters and train models with each combination of the parameters. Additionally, we utilize stratified cross validation with 5 folds. Being stratified means that we have the same class ratio across all folds. We define a grid with different options for the regularization parameter, C. The performance of the models on each folds can be seen on Figure 3.

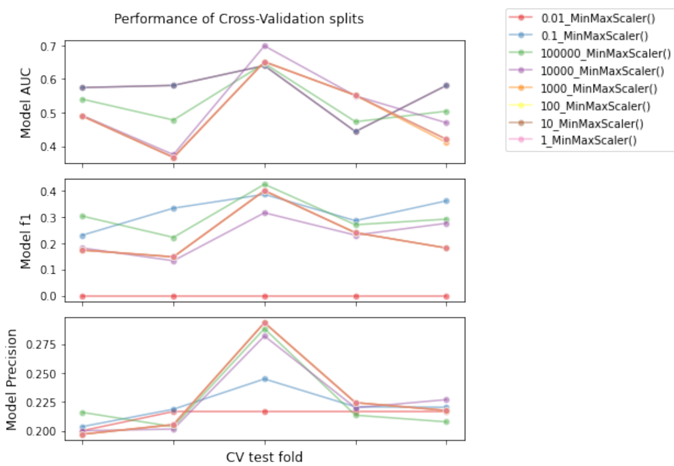

Fig. 3: Cross-validation performance of SVC model with different C values

There are eight different C values. However, less are showed on the plots. This is due to models performing similarly with nearby values. Looking at the different metrics; AUC, f1, precision, it can be seen that different C values affect the models. Additionally, models performing well at one metric might not succeed according to another metric. The best model is selected according to the average precision metric. The validation performance is evaluated. On figure 4 it shows, that we have less false positives and true positives.

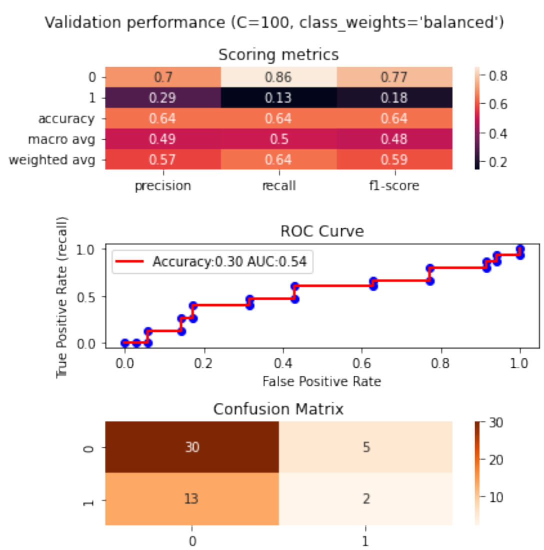

Fig. 4: Performance of best model looking at C

There are still many misclassified observations. The next step that we will is feature selection. We will utilize *SelectKBest* with *chi2* as selection method. The selection step is included in the grid search of the best set of parameters. We try with a range of k features. The best model according to precision ends up being, k=*21*, C=*100*. The performance of this model can be seen on Figure 5.

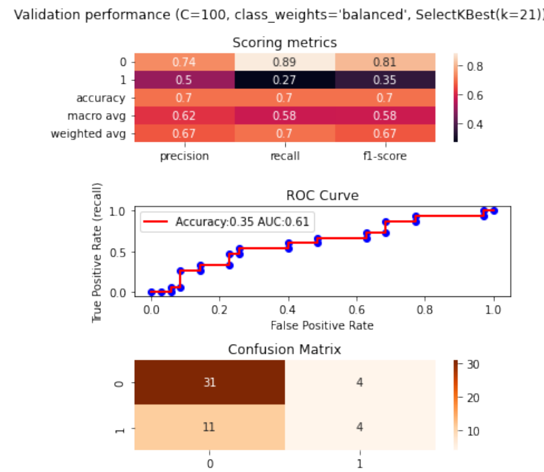

Fig. 5: Performance of best model looking at C and SelectKBest

The model seems to identify less false positives while classifying more true positives. Hence, selecting 21 features seems to be positive for the model. Next we will try different kernels of the SVC model. It showed that the default kernel that was utilized already, Radial Basis Functions, is the best performer according to precision.

III. Discussion

Write Discussion

IV. Conclusion

TP, TF, FP, FF

Not optimal model, not detecting the FP

Write Conclusion

Other model

conclude in regards to clinical setting