University of California, Berkeley

IEOR 242
Fall 2021 Project

Predicting Osteoarthritis with Machine Learning

Aishwarya Acharya, Antoine Reinaud, Oriane Cavrois
Vijitha Cheekala, William Mpondo

Contents

1 Introduction

Osteoarthritis (OA) is the most common form of arthritis of the knee that affects individuals of age 50+. OA causes the cartilage in the knee joint to gradually wears away, resulting in pain to carry out every day activities like walking. Osteoarthritis is one of the most quickly expanding chronic conditions, thanks to global ageing and the growing obesity pandemic [1]. OA, which most commonly affects knees and hips, can increase the risk of cardiovascular disease by limiting activity due to significant joint pain. It also has a detrimental impact on quality of life [2].
The damage is irreversible and treatments include physiotherapy, weight loss, painkillers, anti-inflammatory steroids, and knee replacement surgery. While risk factors for both frequent knee pain and knee OA may include old age, and overweight/obesity, not all people with these characteristics develop the condition.
Therefore early identification is critical to control the progression of the wear, which can be done by identifying the significant risk factors in data.

2 Objective

The aim of our model is to **predict if a patient will contract osteoarthritis in the next 5 years**, given physiological observations (age, body mass, knee pain, etc.) and diverse information on the patient's medical history (previous surgery, etc.).

3 Data Processing and Feature Engineering

We get data sets on the NIMH data archive website. This association conducted a ten-year long study, called the OA initiative, with almost 5,000 patients, diagnosed with OA or not. The patient's condition was observed and analyzed along the study. The aim was to better understand and prevent knee arthritis.
We had several data sets with information on each patient: general information, biomarker evolutions, serum analysis, medication information and X-ray images. For each type of information, there was one data set per year.

The data pre-processing part required several steps: Building of the outcome: At the beginning, we had one data set per year with one outcome in each data set, which was if the patient had been diagnosed with knee OA in the past year or not. Our goal was to predict, for the patients who were not sick at the beginning of the study, whether or not he/she has been diagnosed within 5 years after the beginning of the study. As this outcome was not directly available, we needed to build it with the following steps:

- Extraction of the outcome for each data set
- Creation of a new column (our final outcome) to combine the outcomes:
  - If the patient has been diagnosed at some point during the past 5 years, the final outcome is equal to 1
  - Otherwise, it is equal to 0
- Removal of the rows (patients) already sick at the beginning of the study

The next step was to see the large number of features, analyse them and figure out which ones will be the most useful. The following steps were taken before the data was set to be used in the models :

- The file had to be first converted from a `.txt` file to a `.csv` file
- The intial dimensions of the data frame was 4796 rows × 1187 columns which means there are 1187 featured. We divided the 1187 features amongst the 5 of us to manually find out which features make more sense for the analysis.
- We came down to 139 features after the manual analysis and amongst these 139 features we removed the features which had more than 25% of missing data.
- Once these are removed we came down to 121 columns and then we replaced the missing data with the most common value of the particular feature.
- Once this is done the data frame was good to go for being used in the models.
- At the end of building models, we also assigned feature importance score to all the features to understand the best 25 features and built models using just the 25 most important features to get more robust models.

4 Model Analysis

4.1 Baseline Model

Before fitting any model to the data, it is necessary to know the prediction of the baseline model. The metric used is accuracy, as the objective is to know the ratio of correctly predicted outcomes. The performance of the baseline model will be used as a reference value to compare with the other models.

The most prevalent outcome in the training set is 0 (i.e., the patient will not have OA). The performance of this prediction on the test set are as follows:

Baseline Model		
Accuracy	**TPR**	**FPR**
0.5554	0.0000	0.8000

4.2 Logistic Regression

Predicting if a patient will get OA or not is a classification problem. Therefore, one of the first models to apply is a logistic regression. To do so, we use dummy encoding. The best Logistic Regression model that we obtained is summarized in the following table.

Logistic Regression		
Accuracy	**TPR**	**FPR**
0.5743	0.5695	0.4250

Note: At first, we used a logistic regression from `statsmodels` and we developed an automated algorithm to remove one-by-one the highest p-values, calculate between each removal the new accuracy, and eventually return the best one. However, at some point, we face an '`Singular matrix`' error from `statsmodels` that we were unable to solve. Therefore, we chose to switch to `sklearn` library.

4.3 Clustering with Logistic Regression

We wanted to see if we could improve the performance of our model by using clustering. As the observations are patients, it makes sense to try to make clusters of patients with similar characteristics (age, sex, etc.) which could influence the likelihood to have osteoarthritis or not.

Pre-processing: We first clean the data and transform some features into dummy variables. We also need to normalize the features, so each feature has the same weight when computing the clusters. Then, we split the data set into a training set and a testing set.
Clustering: Then, we computed clusters on the training set. To determine the number of clusters, we performed a cross-validation, looking for minimizing the WCSS. We selected $k = 8$ clusters, as it corresponds to a knee in the WCSS curve.
Training: We split the observations of the training set according to their cluster, to have one data frame for the observations of each cluster. We chose a Logistic Regression model. We used the `sklearn` library, as our matrix was not invertible with `statsmodels`, and `sklearn` was able to deal with them anyway. We trained separately one model on one cluster's data set. At the end, we had 8 different logistic regression models, one for each cluster.
Testing: We determined to which cluster belongs each observation of the testing data set. To do so, for each observation, we computed its distance to each cluster centroid, and we chose the cluster with the smallest distance (we used a classical distance measuring). Then, we created 8 new data sets, with data set i containing the observations of cluster i. We finally assessed our models, by computing the accuracy.
Features selection: As we have a big difference between the accuracy on the training and the testing data, we are going to perform features selection to reduce the overfitting of the training data. For each model, we removed the features with a low coefficient and a high p-value (we adapted the threshold to the cluster). We trained and tested our models again, but the performance only slightly improved from the models without features selection.
Results: We computed for each model the accuracy, TPR, and FPR. The performance metrics did not improve much overall, compared to the other models, with an accuracy between 0.5549 (cluster 4) and 0.7465 (cluster 7).

Clustering w/ Logistic Regression							
Accuracy for cluster #							
0	**1**	**2**	**3**	**4**	**5**	**6**	**7**
0.6218	0.6331	0.5793	0.6241	0.5549	0.5872	0.6571	0.7465

4.4 Random Forest

A Random Forest Classifier model can be used for our prediction problem. Using the accuracy as the performance metric, we perform a Random Forest with and without cross-validation. The results are summarized in the following table.

The 5 most important features are:

1. Body Mass Index 2. Daily Cholesterol Nutrients 3. Physical Activity Scale for the Elderly (PASE) 4. Daily Calcium Nutrients 5. Daily Vitamin D.

Random Forest Classifier					
RF w/o CV			**RF w/ CV**		
Accuracy	**TPR**	**FPR**	**Accuracy**	**TPR**	**FPR**
0.6070	0.302	0.149	0.6140	0.276	0.109

4.5 Gradient Boosting

We also used a boosting model, to obtain better results. Our final, and most accurate version of our Gradient Boosting model was obtained by only keeping the 25 most important features, as given by the `feature_importance` function of the Gradient Boosting model from `sklearn`.

The 5 most important features are:

1. Daily Cholesterol Nutrients 2. Daily Vitamin D 3. Daily Calcium Nutrients 4. Physical Activity Scale for the Elderly (PASE) 5. Body Mass Index

Gradient Boosting Classifier		
Accuracy	**TPR**	**FPR**
0.6337	0.4163	0.3163

4.6 Ridge

We use the `RidgeClassifier` from the `sklearn` library to try and predict the cumulative outcome. The value for the regularization parameter was found through a cross validation and the one standard error rule adapted to an accuracy scoring(see code in appendix), and allows us to improve on a classical logistic regression.

Ridge Classifier		
Accuracy	**TPR**	**FPR**
0.58596	0.35	0.18

5 Interpretation

As we can see, the models that we have were moderately performant, with accuracies ranging from 0.6 in the worst case to 0.74 in the best case (cluster and predict on cluster n°7). We can draw multiple conclusions from this:

- The ML models remain better than the baseline, but not by a great margin.
- The fact that observations (such as knee pain) are evaluated qualitatively, than encoded may lead to a loss of information that translates into poor model performance
- The fact that models only slightly improve when we select important features may be explained by the fact that some key information necessary to OA diagnosis may be missing
- The performance may also be explained by the great number of missing data, that doesn't allow us to exploit all information we may have had (90% of the features were removed in the data processing)

To improve on our prediction, we could use other models, such as neural networks, that may be able to figure out more complex patterns that the usual models we applied can not find.

Our models still allow us to infer that nutrition factors - namely calcium, cholesterol and vitamin D levels - as well as mass index and activity play a pivotal role in contracting OA.

Nevertheless, we must keep in mind that our model is destined to assist doctors in making their diagnosis, and raise awareness on a patient's chances to contract OA or not, in order to prescribe preventive treatment. The models should not be substituted to a doctor's diagnosis, but rather ran prior to one, to alert the doctor on the eventuality of OA for a patient.

6 Conclusion

This project led us to an effective applications discovered in the class. In spite of extensive data processing, feature engineering, and the implementation of multiple different models, we managed to get at best a 74% accuracy, which shows how difficult it is to do predictions in healthcare.