Project 01 – Multi-label Classification in Differential Diagnosis

Disease Prediction Using Machine Learning

Summary: This project is an introduction to machine learning application for automating medical diagnosis

Contents

Chapter I. Preamble

Diagnostics in the medical sense involves a set of rules, methods, decisions that ultimately allow one to conclude about the presence or likelihood of a particular disease. Functional diagnostics is a specific and well-defined section of medical diagnostics. Assessing the prospects and development of functional diagnostics in recent years, one can note that there are both skeptics and optimists. Skeptics say that since the basis of functional diagnostics is clinical physiology, that is, analysis of regulatory processes occurring in the human body and indicating the presence of a disease, and clinical physiology itself has not undergone major changes in recent years, then functional diagnostics is currently undergoing a stabilization phase without waiting for revolutionary breakthroughs. Optimists think the exact opposite. They say that the accumulation of technological innovation and recent data is leading to qualitative and quantitative changes that open up new horizons in the field. There is no doubt that with the development of modern technologies, we are able to identify ever-earlier changes in regulation associated with the disease. Traditionally, the subject of study was the processes in the body, proceeding either in a state of relative rest, or with an artificially created substantial load, that is, in fact, it was a provocative functional diagnosis.

The diagnosis is a conclusion about the nature of the disease and the patient's condition, expressed in the accepted medical terminology. Its establishment is carried out based on data from clinical and additional examinations of the patient and involves a transition from an abstract theoretical assumption about the presence of a particular disease to a specific conclusion, taking into account the totality of anatomical, pathogenetic, etiological, social and symptomatic facts that take place in this case.

In the process of establishing a diagnosis and developing recommended treatment, the physician often faces a whole series of difficult decisions. In most cases, the doctor finds solutions heuristically, relying on his intuition. To develop the appropriate skills, he has to undergo a long course of study, which includes 10-12 years of classes in educational institutions, practice in a hospital and often several more years of special training. Unfortunately, over the years, the future specialist only rarely, if ever, has to deal with the logical foundations of diagnostics or with the very methods of choosing a solution, although both are an important part of his responsible activity. The doctor gradually masters the appropriate techniques purely practically. From publications in medical journals, extensive discussions and case histories, it is clear that the issues of the effectiveness of diagnostic methods are of great importance in medicine. The medical literature also reflects extremely difficult cases in which the doctor often has to make responsible decisions. This clearly implies the need to analytically formulate such problems that the doctor faces.

Chapter II. Introduction

Automation of medical diagnostics is a set of mathematical and technical techniques carried out in order to increase the reliability and effectiveness and accelerate medical diagnosis. This approach assumes a partial or complete transfer of the doctor's functions to devices and machines. The diagnosis consists of the following stages: 1) collection of information about the patient and the manifestations of the disease; 2) processing and evaluation of the collected data; 3) the actual establishment of the diagnosis. One can apply automation of medical diagnostics to each of the stages of diagnostics or the entire process. At the same time, it is possible to automate totally only those tasks of medical diagnostics for which there are algorithms and which, in principle, can be solved without the participation of medical personnel. For now, it is impossible to replace the doctor with some kind of AI, but it is advisable to develop a clinical decision support system to help the doctor in his work.

Clinical decision support system is a medical information system designed to assist doctors and other medical professionals in working with tasks related to clinical decision making. The development and implementation of such systems in practice belongs to the most important areas of development of artificial intelligence in medicine. Already, the topic of clinical decision support systems is one of the most popular in the media, blogosphere and social networks in the field of medicine. There are constant news about the creation by various companies, start-ups and research associations of new developments in the field of medical decision support, including the use of machine learning and other artificial intelligence technologies.

In this case, you have a dataset of patient outcomes, including symptoms that the patients experienced and the diagnoses that doctors gave them. The main task is to write a program to diagnose the patient by a set of symptoms. To do this, you need to analyze the available data, build a predictive model and evaluate the effectiveness of its work.

This problem is actually a multi-label classification problem, so you can use a wide range of classifiers to solve it, ranging from logistic regression to fully connected neural networks. We invite you to try many of these predictive models, compare their performance, and conclude which one is best for the task. We hope you enjoy it.

Chapter III. Goals

The goal of this project is to give you an example of using a machine learning approach to solve medical problems. You will try different algorithms to predict a diagnosis by a set of symptoms like a real doctor. You can actually use your program for the creation of a clinical decision support system.

Chapter IV. General Instructions

- This project will only be evaluated by humans. You are free to organize and name your files as you desire.
- Use Python as a programming language and any libraries and packages supported.
- Use Google Colab, Jupyter or PyCharm as a development environment.
- Write your program so that other people can understand it.
- Store the dataset in your Google Drive or locally to access it from your program.

Chapter V. Mandatory Part

a. Dataset

You will work with open dataset “Disease Prediction Using Machine Learning”:
https://www.kaggle.com/kaushil268/disease-prediction-using-machine-learning

Complete dataset consists of two CSV files: “Training.csv” is for training and “Testing.csv” is for testing your models. Each CSV file has 133 columns: 132 of these columns are symptoms that a person experiences and last column is the diagnosis. There are 41 diagnoses in total. You are required to train your model on training data and test it on testing data.

b. Task

1. Data Analysis
   - Read both training and testing datasets and check their shape. They should have the same number of columns. If they do not, solve this problem. How many rows are there in the training and testing samples?
   - Calculate how many ones are there in each column. What is the most popular symptom? What is the least popular symptom?
   - Visualize counts of each value in the columns of the most popular symptom and the least popular symptom as a bar plot.
   - For each disease, find the symptom that occurs most often and the symptoms that never occur at all.
   - Calculate numerical distribution characteristics for each column: mean, median, standard deviation, variance.
2. Data Preparation
   - Find the symptoms that never occur at all and exclude them from the dataframe.
   - Visualize the class occurrence ratio both in the training and in the testing sample. Is there class imbalance in the samples? Which disease occurs the most frequently in the testing sample?
   - Separate target column from other features and perform One-Hot Encoding of it. How many different diseases are there in the dataset?
3. Classification Models
   - Use Logistic Regression model for multi-label classification.
   - Try different kernels with multi-label SVM classification model: linear, polynomial, RBF, sigmoid. Which one is the best for this problem?
   - Try to build a decision tree to classify the disease.
   - Use common general purpose ensemble classifiers: Random Forest and Gradient Boosting.
4. Neural Networks

   - Train simple build-in Multi-layer Perceptron classifier.
   - Build your own fully connected neural network architecture for classification, consisting of several Dense layers and Dropout layers between them.
   - Try to modify your neural network changing the number of neurons. Build several versions of it, varying the size.
   - Visualize dependence of accuracy and loss function on the epoch number.
5. Model Evaluation
   - For each classification model mentioned above, train it using a training sample and evaluate the model performance using the given testing sample.
   - Calculate the classification accuracy.
   - For each class evaluate Precision, Recall and F1-score.
   - Show the confusion matrix.

c. Implementation

You can work in your private Git repository so a reviewer can access it.

You can use any library or any framework you want.

You should keep a research diary with all information about the used approaches and their metrics.

d. Submission

Share your program on your private Git repository with your reviewer to submit it. This repository should contain your working program and text explanation.

Chapter VI. Bonus Part

1. Estimate the training time and prediction time of each classification model and conclude about their computational efficiency.
2. Extract feature importance from Random Forest and Gradient Boosting models. Which features are the most important?
3. Use well-known feature selection algorithms to reduce the feature space. For example you can use feature selection functions implemented in the scikit-learn library. Find the minimal essential number of features that provide almost perfect classification accuracy. What features were selected?
4. Investigate the dependence of the classification accuracy on the size of your neural network. Find the smallest neural network that provides almost perfect classification accuracy. How many trainable parameters are there in this neural network?
5. Make final conclusions and recommendations for solving the problem of diagnosing diseases by symptoms: choose the best classifier, justify your choice, describe the main steps of the algorithm for solving the problem.

Chapter VII. Submission and Peer-Correction

Submit your private Git repository as usual. Only the content of your private git repository will be graded.

Here are the points that your peer-corrector will have to check:

- if all the approaches are tried (classifiers, neural networks),
- if the almost perfect accuracy achieved on the test dataset,
- if the research diary exists.