0052621 - Statistical Foundation of Machine Learning

Machine Learning Method used in Research Questions

Prepared by: Xhulio Isufi
Instructor: Piter Libin
Date: 25/05/2022

Contents

1 Research Question 1

When a patient is accommodated in the hospital, he or she is accompanied with a file that holds their features such as gender, pregnancy result, disease history. If the patient is hospitalised in order to treat breast cancer, another file with more detail information can come in handy. This file contains things such as: mean size of the core tumor, mean of local variation in radius lengths, mean of severity of concave portions of the contour etc. However, this information needs extensive time to be analyzed by a team of experts, and taking into consideration the coming patients, it becomes an impossible task for a human in real-time. To assists first responders and hospital planners in such situations, machine learning methods become necessary to automatize the decisions and efficiently allocate hospital resources.

This first part of assignment concerns the use of support vector machine (SVM) method to handle breast cancer data in real time. The dataset we are utilizing comes from Kaggle website. There are 569 people in this experiment, and each of them has 31 distinctive features. The main research question is: *Can SVM classify correctly patients for diagnosis of breast tissues with malignant or benign .*

1.1 Dataset

In the above section we, elaborated the task that we will solve in this assignment. This section discusses the dataset on which that task will be solved and analyzes patients' features. We also discuss the preprocessing steps of the data.

1.1.1 Features & their representations

`https://www.kaggle.com/code/mehmetramazanektirir/breast-cancer-analysis/data` is the url of our dataset that we collected from kaggle. The patients of this experiment are 569 and each of them has 31 features including:

- ID
- Diagnosis
- Radius Mean
- Date of first symptom
- Fractal Dimension Mean
- Texture Mean
- Perimeter Mean
- Area Mean
- Smothness Mean
- Compatcness Mean
- Conactivity Mean
- Symmetry Mean

- Radius Se
- Texture Se
- Perimeter Se
- Area Se
- Smoothness Se
- Compactness Se
- Symmetry Se
- Conectivity Se
- Fractal Dimension Se
- Radius Worst
- Texture Wors
- Perimeter Worst
- Area Worst
- Smoothness Worst
- Concavity Worst
- Symmetry Worst
- Fractal Diimension Worst

1.1.2 Data Pre-Processing

In this subsection we will talk about the pre-procession of the data.

• In our dataset, the diagnosis were set as M (malignant) and B (benign). The problem is that our SVM method doesn't accept string as input or output so we replace M with 1 and B with 0.

• We removed the first column which is the ID number, since it is not relevant to our input, either our output.

1.2 Numerical Results

Table 1: Support Vector Machine Accuracy

	Test 1	**Test 2**	**Test 3**	**Test 4**	**Average**	**Standard Deviation**
Train Accuracy	95.6%	96.48%	97.94%	97.65%	96.91%	0.9
Validation Accuracy	98.23%	99.11%	92.03%	95.57%	96.23%	2.7
Test Accuracy	93.91%	95.65%	95.65%	97.39%	95.65%	1.3

From the Table 1 we can say that the accuracy of support vector machine for Task 1 is: 97.39%. The model is powerful enough to predict 9/10 breast cancer correctly. Moreover the model can't go higher than 97.39% due to small amount of training data. We ran the program 4 times and took the average accuracy of train, validation and test which is approximately the same and as a result of that we can say that we have avoided over-fitting and under-fitting. We believe that 4 tests may seem little statistically, but we have seen a consistency between the test which can be proven by the small value of the standard deviation. Also we used shuffling data before running the program as a tool to help us to avoid over-fitting and under-fitting. We chose our regularized parameter C = 750, kernel = 'lienar' and kernel coefficient $\gamma = 1$.

1.2.1 Parameter Tuning

In this section we will talk about the decision that we made in solving the task with support vector machine. First we separate our data in 60% train 20% validation 20% test with 341, 113, 113 patients respectively. Secondly we need to choose our regularized parameter C, type of kernel and kernel coefficient: γ. We use GridSearchCV function from sklearn framework to find the most efficient combination between kernel, C and γ. We searched for C in the interval [0.01:10000] and chose 20 values in that interval and searched for γ in the interval [0.0005:10] and chose 15 values in that interval. The function tries out each combination, compare the accuracy between them and find out the best approach. In or case we come in conclusion that C = 750 and $\gamma = 1$ and kernel of type linear were the best decisions.

2 Research Question 2

In this section we will continue to work with the same database as we were using in the first research question. We will like to address which patient feature influences the most the classification accuracy of the machine learning method used in the first research question? Answering this question we aim to identify to which features it should be put more care when filling the patient file.

By answering these research questions, we believe that Support Vector Machine for patient's treatment will have a big impact on how the facilities of the hospital should be rearranged and how to prioritize which patient should be treated first.

Table 2: Research Question 2 - SVM

	Train Accuracy	**Validation Accuracy**	**Test Accuracy**
Radius Mean	97.65%	91.15%	94.78%
Texture Mean	97.36%	95.57%	95.65%
Perimeter Mean	96.77%	94.69%	97.39%
Area Mean	96.77%	97.34%	94.78%
Smoothness Mean	97.65%	93.8%	95.65%
Compactness Mean	96.18%	96.46%	97.39%
Concavity Mean	97.36%	93.8%	90.4%
Concave Points Mean	97.36%	96.46%	91.3%
Symmetry Mean	96.77%	97.34%	92.17%
Fractal Dimension Mean	96.77%	95.57%	97.39%
Radius	97.06%	93.8%	93.04%
Texture	96.18%	98.23%	96.52%
Perimeter	97.65%	97.34%	93.04%
Area	97.06%	89.38%	95.65%
Smoothness	96.48%	93.8%	96.52%
Compactness	97.06%	92.92%	93.91%
Concave Points	96.48%	92.03%	95.65%
Symmetry	95.3%	94.69%	98.26%
Fractal Dimension	96.48%	96.46%	96.52%
Radius Worst	96.77%	95.57%	93.04%
Texture Worst	96.77%	91.15%	95.65%
Perimeter Worst	96.48%	97.34%	93.91%
Area Worst	97.06%	99.11%	93.91%
Smoothness Worst	97.65%	93.8%	97.39%
Compactness Worst	97.94%	96.46%	96.52%
Concavity Worst	98.24%	96.46%	92.17%
Concave Points Worst	95.3%	94.69%	94.78%
Symmetry Worst	95.3%	97.34%	96.52%
Fractal Dimension Worst	95.89%	94.69%	97.39%

Table 2 shows the accuracy of our model divided into train, validation and test samples, where one feature misses. The first column is named after the missing feature. In comparison with the research question 1 we ran the test only once since we saw a consistency in the task 1. We believe that this is not the best approach statistically but we stick to this method because doing more tests required much more time effort. Furthermore we shuffled the data before running the program as in our main task to avoid over-fitting and under-fitting which we achieved since the accuracy between train, validation and test is approximately the same. From the table above we observe that the feature that plays the most crucial role is the mean of severity of concave portions of the contour (Concavity Mean) since without it the accuracy drops with: 5.25%, then it follows by the mean for number of concave portions of the contour (Concave Points Mean) where the accuracy drops by: 4.32%. Moreover due to the reason that our first research question avoided over-fitting and under-fitting we use the same parameters in this question.

3 Research Question 3

In this section we will continue to discuss about Support Vector Machines. More precisely we will discuss the effect that the degree has when we use polynomial kernel by visualizing it.

3.1 Dataset

For this task we will use synthetic data. Synthetic data generation is quicker, more customizable, more scalable than real-world data. Data scientists may load synthetic data into machine learning algorithms to simulate any condition. Synthetic test data may be used to simulate 'what if' situations, making it an excellent tool for proving a theory or modeling numerous possibilities. Real-world records can be replaced by synthetic data, which is more accurate and scalable.

We used make_classification() function from sklearn library in python to generate synthetic data. We created 100 samples, where each of those samples has 8 features. In the end we specify the number of classes to 2, which means that it will be a binary problem.

3.2 Parameter Tuning

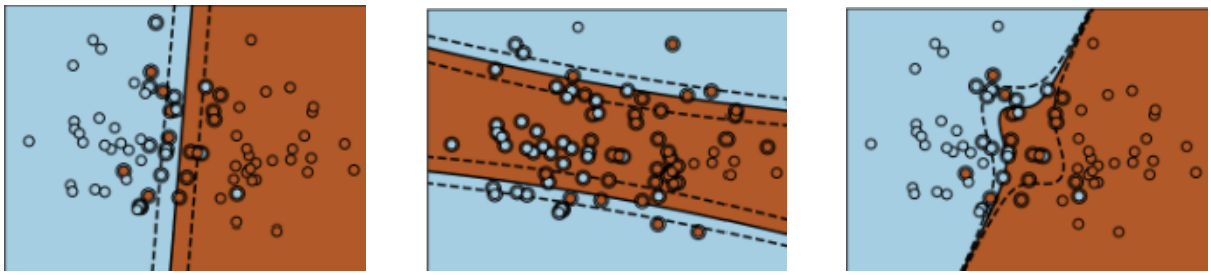
This sub section will be dedicated to the decision that were made towards the parameters that we chose in our training model. In this case we separate our data only in train, 80% and test 20%. Same as in task 1, we use GridSearchCV() function from sklearn library to find the most efficient combination between the reguralization parameter C and γ. We searched for C in the array [1, 0.5, 0.1, 100, 1000] and for γ in the array [1, 0.5, 0.35, 0.1, 0.003, 0.0009]. After running the model the best outcome was C = 1, and $\gamma = 0.003$ with a score of 91% accuracy.

3.3 Numerical Results

Table 3: Support Vector Machine Accuracy

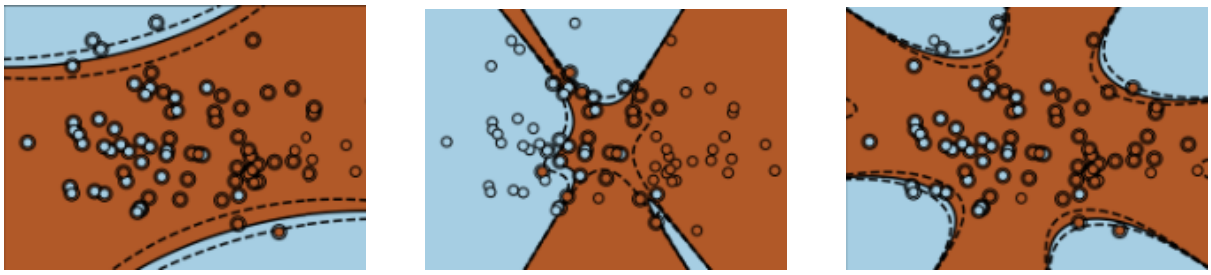
	Test Accuracy
1st Degree	100%
Second Degree	15.78%
Third Degree	84.21%
4th Degree	31.57%
5th Degree	68.24%
6th Degree	26.31%

Table 3 shows the test accuracy for each of the degrees. We can see that when the kernel is linear the accuracy is 100% and then the second highest accuracy happens when the degree of the kernel polynomial is 3, where the accuracy is 84.21%. The lowest accuracy occurs when the degree is 2. The problem with higher degrees is that it performs very well in training, but it doesn't avoid over-fitting, even that we have used the regularized parameter. This can bee seen even in the visualization of the data.



(a) 1st Degree (b) 2nd Degree (c) 3rd Degree

Figure 3.1: Polynomial Degree Kernel



(a) 4th Degree (b) 5th Degree (c) 6th Degree

Figure 3.2: Polynomial Degree Kernel

List of Figures

List of Tables