**AIL302m - Machine Learning**

TRAN THANH DUONG

SE160185

Classification and fine-tuned hyperparameters with SVM and GridSearchCV

**July 06, 2022**

[COLAB NOTEBOOK](https://colab.research.google.com/drive/1F0wuTEYwOhCU2OmLA4DF5zlhHFWHLKvl?usp=sharing)

# Overview

Support Vector Machine (SVM) is one of the most state-of-the-art models in Machine Learning, used for classification and regression analysis. SVM maximizes the distance between the two categories by mapping training examples to points in space. Then, based on which side of the gap they fall, new samples are projected into that same area and predicted to belong to a category. SVMs may effectively do non-linear classification by implicitly translating their inputs into high-dimensional feature spaces, this technique is known as the kernel trick. In this report, I will use the power of SVM for classification problems and GridSearchCV - A well-known scikit-learn library for fine-tuning the hyperparameters automatically to get the best performance.

# 

# Theories about SVM in scikit-learn library

SVM library in scikit-learn supports **Support Vector Classification (SVC)** and even **Support Vector Regression (SVR)**. This model, we just focus on **SVC**.

### 

There are a lot of fine-tuned hyperparameters in SVC, but just pay attention to the following important parameters:

### C

C is a regularization parameter, which tells the SVM optimization **how much you want to avoid misclassifying each training example**. For large values of C, the optimization will pick a smaller-margin hyperplane if it performs better at accurately classifying every training point. On the other hand, if C is very tiny, the optimizer will search for a larger-margin separating hyperplane.

### 

### Kernel

Scikit-learn supports built-in **linear** kernel and non-linear kernel: **poly**, **rbf** (radial basis function) and **sigmoid**.

| **Kernel** | **Formula** | **Parameters** |
| --- | --- | --- |
| linear |  | No parameter |
| polynomial |  | d: degree  r: coef0  y: gamma |
| sigmoid |  | r: coef0  y: gamma |
| rbf |  | y: gamma |

### Degree

This parameter is just available for the polynomial kernel. The higher the degree, the more expensive the computation.

### Gamma

A single training example's effect is determined by the gamma parameter, with low values denoting "far" and large values denoting "near."

# SVM with random hyperparameters

In this section, I used default kernel metrics (rbf) and high regularization with **C=1000**. I will train on two kinds of dataset: original dataset (without oversampling) and oversampling dataset.

### 

### 

### 

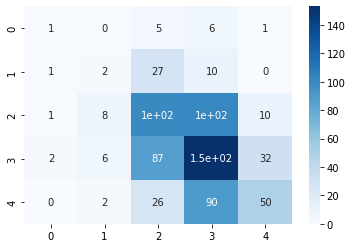
### Results

| **Train/test** | **Metrics** | **Original dataset** | **Compare** | **Oversampling dataset** |
| --- | --- | --- | --- | --- |
| Train | **Accuracy** | 0.54889 | **<** | 0.62232 |
| **F1-Score** | 0.53212 | **<** | 0.60426 |
| Test | **Accuracy** | 0.42382 | **>** | 0.31855 |
| **F1-Score** | 0.41147 | **>** | 0.33069 |

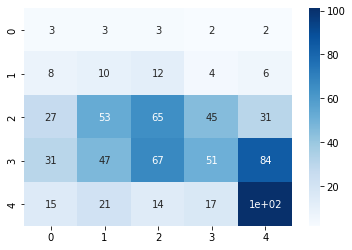
### 

### Confusion matrix on testing dataset

**Training on original dataset**



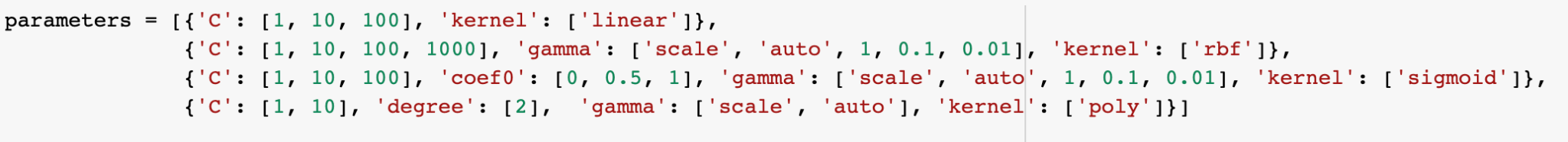
**Training on oversampling dataset**



# SVM with GridSearchCV

GridSearchCV is a powerful library in scikit-learn for fine-tuning hyperparameters automatically to get the best estimator.

In this section, I will test some specific parameters for each kernel function.



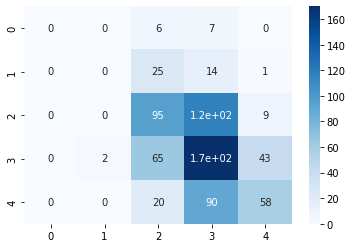
I also run on two kinds of datasets like the above method. The results shows that the best parameters for models in each kind of datasets are:

### Results

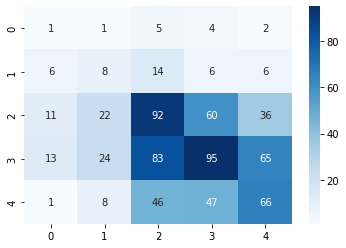
| **Train/test** | **Metrics** | **Original dataset**  **'C': 100,**  **'gamma': 0.01,**  **'kernel': 'rbf'** | **Compare** | **Oversampling dataset**  **'C': 1000,**  **'gamma': 1, 'kernel': 'rbf'** |
| --- | --- | --- | --- | --- |
| Train | **Accuracy** | 0.46844 | **<** | 0.84071 |
| **F1-Score** | 0.44228 | **<** | 0.83665 |
| Test | **Accuracy** | 0.44736 | **>** | 0.36288 |
| **F1-Score** | 0.42584 | **>** | 0.37089 |

### Confusion matrix on testing dataset

**Training on original dataset**



**Training on oversampling dataset**

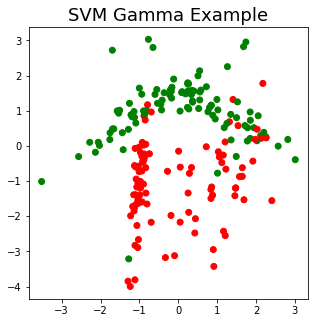


# Discussion

On the first method using SVM with pre-defined hyperparameters, we can see the difference between training on original dataset and oversampling dataset, the test on **original dataset** gives a better accuracy. The reason comes from the imbalanced data, the original dataset is quite imbalanced on class 0 and 1, we can see on the confusion matrix of the original dataset, the model ignored almost all samples with class 0 and 1 and focused on the other classes to give a better accuracy. On the other hand, the oversampling dataset has a balanced dataset, every class is uniform, so the model predicted correctly a few samples of class 0 and 1, but gave worse accuracy.

On the second method using SVM and GridSearchCV, we found out the parameters giving the best performance and see that the high regularization (C=100 or 1000) and medium gamma (gamma=0.01 and 1) gives the best accuracy, which means the datasets work well on smaller-margin hyperplanes and every data points has a large similarity.

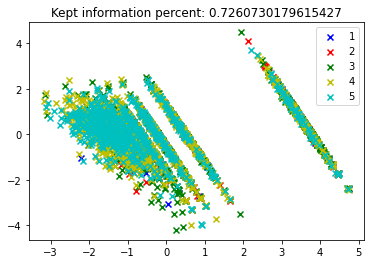
For better understanding, I will plot some cases to show how gamma and C affect the results. I will create a new dataset with 2 features:



| Gamma = 0.0001 | Gamma = 0.001 | Gamma = 0.01 |
| --- | --- | --- |
|  |  |  |
| C=1 | C=10 | C=10000 |
|  |  |  |

As we can see, the higher gamma value is, the "close" of data points to each other in order to be the same cluster, which means the red area is getting smaller and smaller. The parameter C shows the same as gamma, but the higher the value C, the smaller the margin.

I will use PCA (with kept information percent is 72.6%) for **reducing dimensionality** to 2 and plotting to see how it looks.



We can see the data points overlap a lot, if we use high-margin hyperplanes, it will lead to misclassification between classes heavily and give bad results. So the low-margin hyperplanes (high regularization C) will be better. Choosing the higher gamma value in this case doesn't help the model predicting better on the test set, so choosing a medium gamma value is a good choice for both training and testing.