SIA 3611
Année universitaire 2022-2023

TP Machine Learning

TP 4 : AutoML

Vassilis.Christophides@ensea.fr
guillaume.renton@ensea.fr

Table des matières

Introduction

In previous TP, you have learned to use machine learning for different kind of tasks, from regression to clustering through classification. In this TP, you are going to use the earned knowledge on new datasets for regression and classification.

You are going to use 2 new datasets in this TP. First one is california housing, whose target variable is the value of houses in california, expressed in hundred of thousand of dollars. For each house, a set of 9 features is available. There is a total of 20 060 data.

Second one is MNIST, a very popular dataset for handwritten recognition and image classification. The original dataset is made of 60 000 training images of shape 28x28 of handwritten digits from 0 to 9, and 10 000 images for test dataset. For computational time, you will work on a given random subset of MNIST made of 6000 images in train and 1000 images in test.

Objectives :

- Apply your knowledge on new datasets
- Tune models hyperparameters and explore metrics
- Apply principal components analysis and understand its effects on both dataset
- Understand and use Cross-Validation
- Use AutoML to find interesting models

The notebook corresponding to this TP is available on Moodle. Answer the questions directly on this notebook. The notebook has to be submitted before next week. The work can be done by pair, but one notebook is awaited per student.

1 Getting started with new datasets

Regression

In first part of step 1, you will work on the regression problem with the dataset california housing.

1.1 To do

Execute the following cell to load the california housing dataset and normalize it.

```
from sklearn.datasets import fetch_california_housing
from sklearn.preprocessing import normalize
import numpy as np

X, y = fetch_california_housing(return_X_y = True)
X = normalize(X)
```

1.2 To code

Apply Stochastic Gradient Descent and SVR methods and cross validate your results using 5 folders. For this, you can either use the function cross_val_score (or any other method for cross validation in sklearn) or either compute yourself the cross validation. According to a relevant metric optimize both methods. For SGD you will optimize the value of alpha for both L2 and L1 penalty score. For SVR, you will optimize the kernel. Be careful with the metric if you use cross_val_score, the returned values are often negative.

Question 1

According to your metric, which method obtain the best result ?

Question 2

What is the interest of using cross validation in general ? Is it relevant in this particular case ?

1.3 To code

Transform your data according to principal component analysis, and optimize the number of components according to the same metric than previously for both models.

```
from sklearn.decomposition import PCA
pca = PCA(n_components = 1)
X_pca = pca.fit_transform(X)
```

Question 3

What is the interest of Principal Component Analysis in general ? Is it relevant here ?

Classification

1.4 To do

Execute the following cells to load a subset of MNIST dataset. Since the dataset is already divided into training/test, we won't use cross validation this time.

```
import pickle
with open("data/mnist.pkl", "rb") as f:
    ((X_train, y_train), (X_test, y_test)) = pickle.load(f)
```

1.5 To code

Compute classification on those images using a KNN classifier and an Adaboost classifier. For each classifier, optimize the parameters according to a relevant metric. For the KNN classifier, you will optimize the number of neighbor while for the Adaboost classifier, you will optimize the base estimator along with the number of estimators (for the basis estimator, limit yourself to different depth of decision tree classifier).

Also, for each model, compute the confusion matrix.

Question 4

According to your metric, which method obtain the best results ?

Question 5

According to the confusion matrix, which class if the easiest to classify ? Which ones are the most difficult ? Which ones are the most confused with each other ?

Bonus

For the Adaboost classifier, explore other classifier as base estimators. What are the limitations about those estimators ?

1.6 To code

Transform your data according to principal component analysis, and optimize the number of components according to the same metric than previously for each classifier.

Once again, compute the confusion matrix for each model.

Question 6

Is the use of PCA relevant here ?

Question 7

Did your answers from question 5 changed with PCA ?

2 AutoML

In this second section, we discuss on the utilisation of AutoMl tools, such as auto-sklearn. If you are using colab or don't have auto-sklearn installed (much likely), you may need to run the following cell at first in order to install auto-sklearn. This will require you to restart the runtime (a prompt will invite you to).

Restarting the runtime will clear all your variables and imported libraries, so you will need to import them again.

```
!pip install --force-reinstall scipy==1.6
!pip install --force-reinstall auto-sklearn==0.15
```

2.1 To do

Execute the following cells.

```
from sklearn.model_selection import train_test_split
X, y = fetch_california_housing(return_X_y = True)
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)
```

```
import autosklearn.regression
import sklearn.model_selection
import sklearn.datasets
import os, shutil
from sklearn.metrics import mean_squared_error, mean_absolute_error

automl = autosklearn.regression.AutoSklearnRegressor(
    include = {'regressor': ["libsvm_svr", "sgd"]},
    time_left_for_this_task=120,
    per_run_time_limit=30,
    tmp_folder='/tmp/california_housing_tmp',
)
automl.fit(X_train, y_train, dataset_name='California_Housing')

print(automl.leaderboard())

y_pred = automl.predict(X_test, y_test)
print("MSE = ", mean_squared_error(y_test, y_pred))
print("MRE = ", mean_absolute_error(y_test, y_pred))
```

```
from pprint import pprint
pprint(automl.show_models(), indent=4)
```

Question 8

What are the evaluated models by autoML ? Which model obtain the best performance ? What are the parameters of the best model ?

2.2 To code

With the help of the previous code, use autoML for the classification task on MNIST, by limiting the exploration to KNN and Adaboost.

```
```

Question 9

What are the evaluated models by autoML ? Which model obtain the best performance ? What are the parameters of the best model ?

3 Bonus step

As a bonus step, have fun and remove a maximum of constraints of your autoML model. Which model obtain the best performances ? Describe the parameters of this model. You can do it for either for regression or classification or both.