TP 1 - Apprentissage Statistique Appliqué

Nokri Amale, Rahis Erwan, Vuillemot Bertrand

2020 - 2021

Contents

1 Partie 1

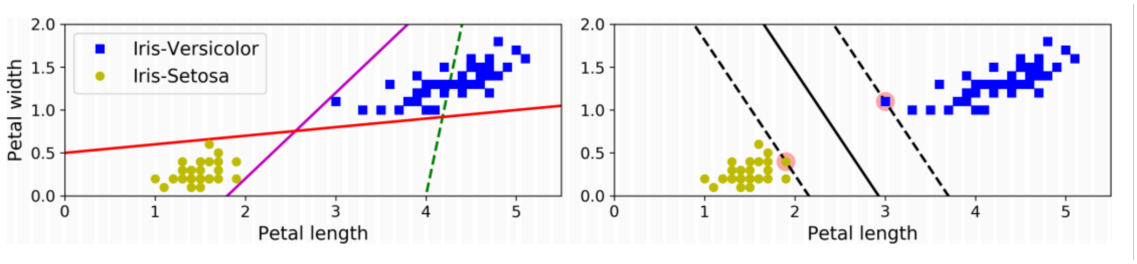
1.1 Cross-Validation with GridSearchCV

Question: Explain in your report what happens when we run clf.fit(X_train, Y_train). What is the complexity for each of the three following cases?
The line clf.fit(X_train, Y_train) here uses the fit method on the object clf and taking the train sample. We give the features X and the outputs Y. The object clf is from the class GridSearchCV which allows us to find the best hyperparameters among a list we chose. It is taking as parameter an object named knn of the class KNeighborsClassifier(), a dictionary named parameters containing the number of neighbors to be tested in the knn algorithm (1 to 5 here) and the cv parameter referring to the number of folds to be used in the cross-validation. Basically it will perform a 3-folds cross-validation on a kNN model with 1 to 5 neighbors on the train sample and it will allow us to keep the best model. The kNN algorithm is parametered with the default metric which is the Euclidean distance : $\sqrt{\sum_{i=1}^{n}(x_i - y_i)^2}$. The functions are all part of the sklearn package.

Question:What is the test accuracy? What would be the accuracy of random guess?
The test accuracy is the measure of how often the points are correclty classified. In our case the accuracy is 0.875. It means that 87.5% of the time, the points are correctly classified on the test sample. If we did a random guess we would randomly choose an output in the range 0 to 9 so the accuracy would converge towards $\frac{1}{10}$ according to the LLN.

Question: What is LinearSVC() classifier? Which kernel are we using? What is C? (this is a tricky question, try to find the answer online)
LinearSVC means Linear Support Vector Classification. We are using a linear kernel. The parameter C represents the regularization weights, ie the penalty applied on the loss function. The loss function used here is the Squared Hinge Loss : $l(y) = \max(0, 1 - t \cdot y)$

Add description of SVC



:
Question : What is the outcome of np.logspace(-8, 8, 17, base=2)? More generally, what is the outcome of np.logspace(-a, b, k, base=m)?
The outcome of np.logspace(-8, 8, 17, base=2) is a logarthmic space going from 2^{-8} to 2^8 with 17 numbers equally spaced on log scale. The logspace function from the numpy package will return k numbers going from m^{-a} to m^b spaced on a log scale with a log base m.

Question : What is the meaning of the warnings? What is the parameter responsible for its appearence?
The warning tells us that the algorithm did not converge, it did not reach the stop criterion. The parameter responsible for its appearence is the max_iter parameter. Its value is not sufficient for the algorithm to converge. The data variance is maybe too large for the algorithm to efficiently

perform the SVM.

Question : What did we change with respect to the previous run of LinearSVC()?
We are running the svc function which is by default a RBF SVC and not a linear SVC. RBF means radial basis function. We added a parameter MaxAbsScaler() to scale the absolute data between 0 and 1 and thus reduce the variance.

Question : Explain what happens if we execute

```
pipe.fit(X_train, y_train)
pipe.predict(X_test, y_test)
```

Those lines will execute a SVM with a maxabsscaler parameter but with no C parameter which is by default 1.0.

Question : what is the difference between StandardScaler() and MaxAbsScaler()? What are other scaling options available in sklearn?
StandardScaler will normalize the data : $\frac{x-m}{\sigma}$ with m the mean and σ the standard deviation of data. It differs from StandarScaler because absolute values are mapped in the range [0,1].

Add other options for scaling

Question : Using the previous code as an example achieve test accuracy ≥ 0.9 . You can use any method from sklearn package. Give a mathematical description of the selected method. Explain the range of considered hyperparamers.
Answer: choose an algorithm and test

1.2 Visualizing errors

The Logistic Regression function from skLearn package is able to give us the probabilities of each outcome. The figure 1 shows pictures, outcome, real value and probabilities for the 4 first mistakes, ie the 4 first times the predicted value is not equal to the real value.

The error in the chunk of code was because the `predict_proba` method returns an array of probabilities within an array. We must then pick the first element of the array (index 0) to obtain the proabilities array (see line 11).

The code is as follows :

```
1  axes = plt.subplots(2, 4)[1]  # creates a grid of 10 plots
2
3  # More details about zip() function here https://docs.python.org/3.3/library/
       functions.html#zip
4  y_pred = clf4.predict(X_test)
5  j = 0 # Index which iterates over plots
6  for true_label, pred_label, image in list(zip(y_test, y_pred, X_test)):
7      if j == 4: # We only want to look at 4 first mistakes
8          break
9      if true_label != pred_label:
10         # Plotting predicted probabilities
11         axes[1, j].bar(np.arange(10), clf4.predict_proba(image.reshape(1, -1))[0])
12         axes[1, j].set_xticks(np.arange(10))
13         axes[1, j].set_yticks([])
14
15         # Plotting the image
16         axes[0, j].imshow(image.reshape((28, 28)), cmap=plt.cm.gray_r,
       interpolation='nearest')
```

```
        axes[0, j].set_xticks([])
        axes[0, j].set_yticks([])
        axes[0, j].set_title('Predicted {}'.format(pred_label)+'/True {}'.format(
    true_label),fontsize=8)
        j += 1
```

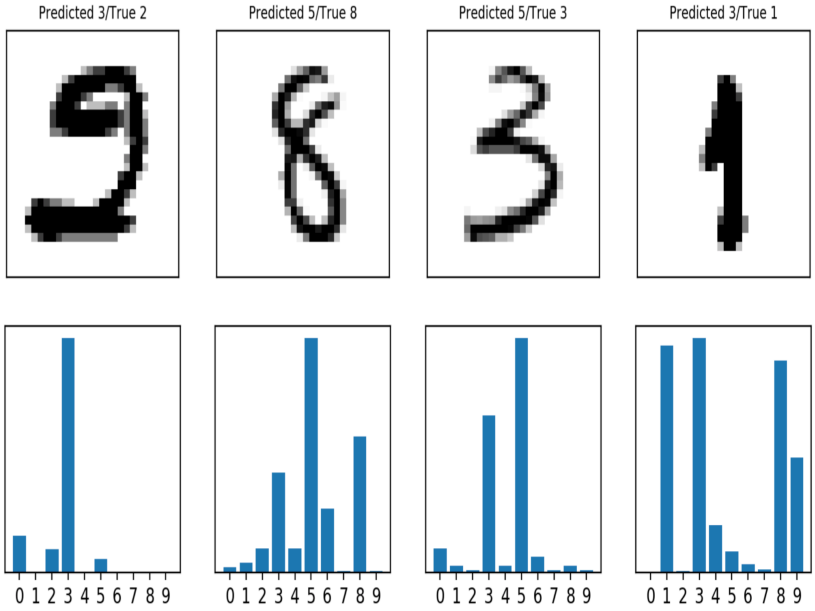


Figure 1: Probabilities of each outcome for the logistic regression

1.3 Changing the loss function

Question: What is balanced_accuracy_score? Write its mathematical mathematical description.
balanced_accuracy_score is a method from the sklearn package that computes the balanced accuracy metric. In classification, the accuracy is the percentage of well-classified individuals on a test sample. The formula is :

$$\frac{TruePositive + TrueNegative}{SampleSize}$$

But this metric does not take into account if the sample is imbalanced. For example if we have more real positives than real negatives. The balanced accuracy takes this into account by taking the average value of true positives divided by real positives and true negatives divided by real negatives as follows :

$$\frac{\frac{tp}{rp} + \frac{tn}{rn}}{2}$$

Question: What is the confusion matrix? What are the conclusions that we can draw from the `confusion_matrix(y_test, clf4.predict(X_test))`**?**

The confusion matrix is the matrix that gives for each classification outcome, real and predicted, the number of individuals. For a binary outcome it is as follow :
The diagonal terms are the number of individuals classified in the right category by the SVM algorithm.

```
[[22  0  0  0  0  0  0  0  0  0]
 [ 0 24  0  0  0  0  0  0  2  0]
 [ 0  0 14  1  1  0  0  0  0  0]
 [ 0  0  0 18  0  3  0  0  1  1]
 [ 0  1  0  0 17  0  0  0  0  2]
 [ 1  0  0  1  0  6  0  1  0  1]
 [ 1  2  1  0  0  0 20  0  0  0]
 [ 0  0  0  0  1  0  0 15  0  0]
 [ 0  2  0  1  0  3  0  0 11  0]
 [ 0  0  0  0  2  0  0  2  1 21]]
```

Here we have a fair amount of well-classified individuals except for the number 5 that has 6 well classified individuals and 6 misclassified individuals.

References