PHELMA - Grenoble INP

Machine Learning

5PMBMLD0

SVM-KMeans-PCA-Sequence 4

Students :
Allan Dizet
Matteo Marengo

28/10/2022

1 Introduction

This Sequence deals through several exercises about SVM-Kmeans and PCA. We will study particularly two exercices : Predicting a breast cancer with SVM classification and PCA // Digit images clustering with Kmeans clustering.

2 Exercise 4 - Predicting a breast cancer

2.1 Dataset description

We look at the variable content of the variable cancer as shown in Fig 5.

```
Features:  ['mean radius' 'mean texture' 'mean perimeter' 'mean area'
 'mean smoothness' 'mean compactness' 'mean concavity'
 'mean concave points' 'mean symmetry' 'mean fractal dimension'
 'radius error' 'texture error' 'perimeter error' 'area error'
 'smoothness error' 'compactness error' 'concavity error'
 'concave points error' 'symmetry error' 'fractal dimension error'
 'worst radius' 'worst texture' 'worst perimeter' 'worst area'
 'worst smoothness' 'worst compactness' 'worst concavity'
 'worst concave points' 'worst symmetry' 'worst fractal dimension']


Labels:  ['malignant' 'benign']
Data set dimensions:  (569, 30)
Values: [0,1]
Positive case: 357
Negative case: 212
```

FIGURE 1 – Variable contents of the variable cancer

The labels names are ['malignant','benign']. Their values are [0,1]. There are 569 samples in the data set. 357 are positive cases and 212 are negative cases.

2.2 SVM classification

We split the data set into two parts : 70 % for the training data set and 30% for the test data set. We train the model with a linear and a Gaussian SVM without and with normalization. We compare the results of the accuracy, precision, and recall in the Tab below.

```
scaler = StandardScaler().fit(X_train)
X_train = scaler.transform(X_train)
X_test = scaler.transform(X_test)

svc_model = svm.SVC(kernel='linear')
svc_model.fit(X_train, y_train)
```

scripts/linearsvc.py

	Accuracy	Precision	Recall
Linear SVM	96.48	96.44	97.99
Rbf SVM	90.45	87.28	99.20
Linear SVM + Normalization	98.74	98.41	99.60
Rbf SVM + Normalization	98.24	98.02	99.20

With and without normalization, the linear SVM is proven to be more efficient than the Rbf SVM (especially for the accuracy and Precision). Furthermore, we observe that we always have better results with normalization. Therefore, it is important to do it before the classification. We can also underline that with normalization the two models have similar results.

2.3 Dataset analysis

The name of the first 10 features are [radius, texture, perimeter, area, smoothness, compactness, concavity, concave-points, symmetry and fractal-dimension].
We observe that with only 10 features there are a lot of them that are overlapping between malign and benign (obvious on data feature distribution and sample repartition according to each feature). When we look in closer details the correlation curves this trend is confirmed.

Therefore, we can do classification with fewer features and still have good results. It is also interesting for the time of calculus.

2.4 Principal component analysis

In order to keep 99%, 95% and 90% of the total variance, it is necessary to have respectively 17, 10 and 7 features.

Variance	100	99	95	90	85	80
Nb Components	30	17	10	7	6	5
Accuracy	98.74	98.74	98.49	96.99	98.24	98.24
Precision	98.41	98.41	98.40	97.60	98.40	98.40
Recall	99.60	99.60	99.20	97.60	98.80	98.80

```
pca = PCA(n_components=2)
principalComponents = pca.fit_transform(X_train)
print(pca.n_components_)
print(pca.explained_variance_ratio_)
```

scripts/pca.py

By using the attribute explained_variance_ratio_, you can see that the first principal component contains 43.70% of the variance and the second principal component contains 19.42% of the variance. Together, the two components contain 63.12% of the information.

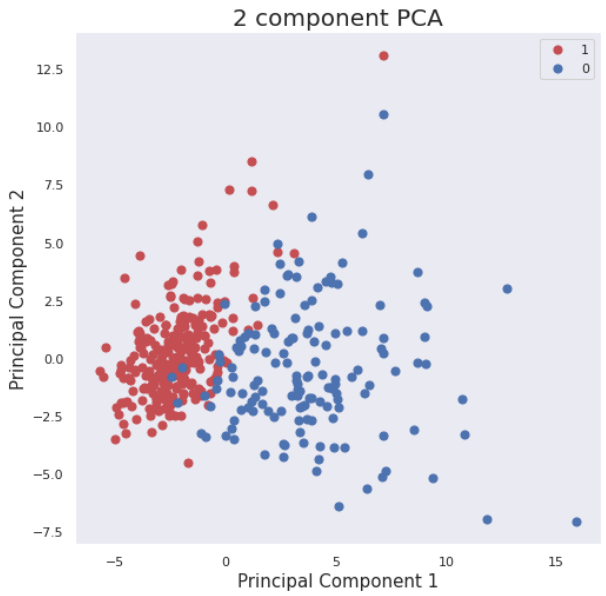

FIGURE 2 – 2 Component PCA

3 Exercise 5 - Digit images clustering

3.1 Loading and visualizing the data

We are going to use for clustering the data inside the variable digits. This feature has a dimension of 1797 x 64.

3.2 Kmeans clustering

FIGURE 3 – Final Digit Centroids

According to Figure 3, we can assume that 1 and 8 are going to be difficult to distinguish, same as 5 and 9.

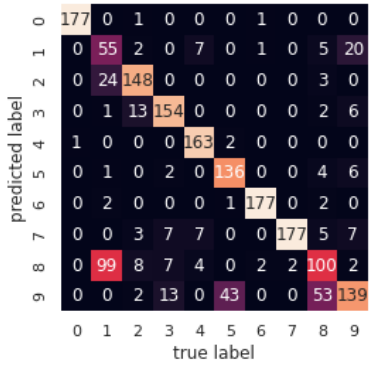

FIGURE 4 – Confusion Matrix

The global accuracy is 79.35%.That is a good result, but it is still not perfect and we want to achieve a way better accuracy. In addition, with the confusion matrix, we can observe that 8 and 1 are estimated at the same probability (99/100). The same for 5 and 9 and 9 and 8 (ratio of 1/3). It confirms our previous assumptions.

3.3 Looking for the best K values

We proceed to the K-mean clustering with k from 1 to 30.

```
sse = []
K = range(1,30)
for k in K:
    kmeanModel = KMeans(n_clusters=k)
    kmeanModel.fit(digits.data)
    sse.append(kmeanModel.inertia_)
```

scripts/sse.py

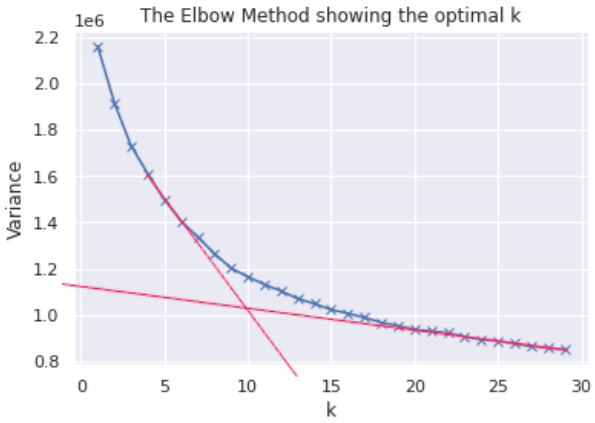

FIGURE 5 – K-Elbow Method Curve
According to the graph, we can determine that the optimal value for k is 10, which correlates with the number of class.

4 Conclusion

We can conclude that SVM - K Means and PCA are powerful techniques to classify big batches of data. PCA allows us to reduce the number of features by eliminating the overlapping data or the one that do not carry enough information. It is then useful to run the classification and discriminate the labels.