نام و نام خانوادگی : صالح زارع زاده

شماره دانشجویی : ۶۱۰۳۹۶۱۰۹

تمرین سری سوم

در ابتدا کتابخانه های مورد نیاز را اضافه میکنیم

```
from sklearn.datasets import fetch_lfw_people
from sklearn.model_selection import train_test_split
from sklearn.metrics import classification_report
from sklearn.decomposition import PCA
from sklearn.svm import SVC
from sklearn.model_selection import GridSearchCV
from sklearn.neural_network import MLPClassifier
```

سپس مقادیر اولیه را مشخص میکنیم

```
lfw_people = fetch_lfw_people(min_faces_per_person=200, resize=0.4)
x = lfw_people.data
y = lfw_people.target
target names = lfw people.target names
X_train, X_test, y_train, y_test = train_test_split(x, y, test_size=0.33, ran
dom_state=21)
```

سپس برای کاهش ابعاد با PCA در آن n_componets را ۳۰۰ قرار میدهیم که اندازه هر عکس را از ۱۸۵۰ به ۳۰۰ میرساند.

```
pca = PCA(n_components=300, svd_solver='randomized',whiten=True).fit(X_train)
X_train_pca = pca.transform(X_train)
X test pca = pca.transform(X test)
```

پارامترهای GridSearchCV که برای ماشین بردار پشتیبان ما است یک estimator و param_grid است را باید مقدار دهی کنیم که estimator یک تابع نمره دهی است که ان را برابر SVC(class_weight='balanced') قرار میدهیم که حالت "balanced" از مقادیر y برای تنظیم خودکار وزن متناسب با فرکانس های کلاس در داده های ورودی به عنوان n_samples / (n_classes * np.bincount (y)) استفاده می کندو پارامترهای param_grid به لیستی از پارامترها و دامنه مقادیر برای هر پارامتر مشخص شده نیاز دارد تا بتواند بهترین ماشین بردار پشتیبان را پیدا کند.

```
param_grid = {'C': [1e3, 5e3, 1e4, 5e4, 1e5 , 5e6 , 1e6],
              'gamma': [0.0001, 0.0005, 0.001, 0.005, 0.01, 0.1 , 0.5 ,1], }
clf = GridSearchCV(SVC(class_weight='balanced'), param_grid)
clf = clf.fit(X_train_pca, y_train)
print("Best estimator:" + str(clf.best_estimator_))
```

```
print("Best parametrs : "+str(clf.best_params_))
y_pred = clf.predict(X_test_pca)
print(classification report(y test, y pred, target names=target names))
```

که در انتها به این نتیجه میرسیم:

```
Best estimator:SVC(C=1000.0, break_ties=False, cache_size=200,
class_weight='balanced',
    coef0=0.0, decision_function_shape='ovr', degree=3, gamma=0.0005,
    kernel='rbf', max_iter=-1, probability=False, random_state=None,
    shrinking=True, tol=0.001, verbose=False)
Best parametrs : {'C': 1000.0, 'gamma': 0.0005}
               precision    recall  f1-score   support

 Colin Powell       0.68      0.90      0.78        69
George W Bush       0.96      0.84      0.90       184

     accuracy                           0.86       253
    macro avg       0.82      0.87      0.84       253
 weighted avg       0.88      0.86      0.86       253
```

سپس پارامتر های MLPClassifier را مشخص میکنیم که عبارت اند از hidden_layer_sizes است که نشان دهنده این پارامتر به ما امکان می دهد تعداد لایه ها و تعداد گره هایی را که می خواهیم در طبقه بندی شبکه عصبی داشته باشیم تنظیم کنیم. هر عنصر در tuple تعداد گره ها را در موقعیت i ام نشان می دهد که i نمایه tuple است و سپس solver آن را 'lbfgs' میگذاریم که یک بهینه ساز از خانواده ی تابع های quasi-Newton است و alpha که برای regularization است را برابر 0.00001 قرار میدهیم که دقت آن افزایش پیدا کند و که نه overfitting داشته باشیم و نه underfitting و همینطور دقت آن افزایش پیدا میکند.