

ابتدا کتابخانه های لازم را import می کنیم
سپس دیتاست مورد نظر را از روش اول بارگذاری می کنیم.

```
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
```

سپس با استفاده از `train_test_split` مقدار `x_train, x_test, y_train, y_test` را
بدست می آوریم

```
X_train, X_test, y_train, y_test = train_test_split(
    X, y, test_size=0.25, random_state=100)
```

در قسمت بعدی کد، کاهش بعد از 1850 به 150 بُعد را انجام می‌دهیم

```
n_components = 150

pca = PCA(n_components=n_components, svd_solver='randomized',
          whiten=True).fit(X_train)

X_train_pca = pca.transform(X_train)
X_test_pca = pca.transform(X_test)
```

در ادامه برای پیاده سازی SVM یا از کتابخانه `sklearn.svm` استفاده می کنیم.
پارامترهای `GridSearchCV` estimator و `param_grid` است که در آن
estimator یک تابع نمردهی است که آن را با
`SVC(class_weight='balanced')` مقداردهی می کنیم که در این حالت از مقادیر `y`
برای تنظیم خودکار وزن متناسب با فرکانس های کلاس در داده های ورودی استفاده می
کند

`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.

مهمترین پارامترهای مورد نیاز هنگام کار با rbf مدل SVR ، gama و c هستند

```
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=0.005,
kernel='rbf', max_iter=-1, probability=False, random_state=None,
shrinking=True, tol=0.001, verbose=False)
```

```
param_grid = {'C': [1e3, 5e3, 1e4, 5e4, 1e5],
              'gamma': [0.0001, 0.0005, 0.001, 0.005, 0.01, 0.1], }
clf = GridSearchCV(
    SVC( class_weight='balanced'), param_grid
)
clf = clf.fit(X_train_pca, y_train)
```

و نتایج آن :

	precision	recall	f1-score	support
Colin Powell	0.94	0.91	0.92	64
George W Bush	0.95	0.97	0.96	128
accuracy			0.95	192
macro avg	0.94	0.94	0.94	192
weighted avg	0.95	0.95	0.95	192

برای neural network نیز از کتابخانه sklearn.neural_network استفاده می کنیم
'lbfgs' یک بهینه ساز در خانواده روشهای quasi-Newton است.
hidden_layer_sizes: این پارامتر به ما اجازه می دهد تعداد لایه ها و تعداد گره هایی را که می خواهیم در طبقه بندی شبکه عصبی قرار دهیم تنظیم کنیم. هر عنصر در tuple تعداد گره ها را در موقعیت ith نشان می دهد که i نمایه tuple است.
همچنین آلفا یک پارامتر برای اصطلاح regularization ، یا همان penalty term است . که با قرار دادن 0.00001 دقت regularization را افزایش می دهیم.

```
clf_neural = MLPClassifier(solver='lbfgs', alpha=1e-5,
hidden_layer_sizes=(5, 2), random_state=1)
```

```
clf_neural = clf_neural.fit(X_train_pca, y_train)
y_pred = clf_neural.predict(X_test_pca)
```

و نتایج آن :

	precision	recall	f1-score	support
Colin Powell	0.83	0.84	0.84	64
George W Bush	0.92	0.91	0.92	128
accuracy			0.89	192
macro avg	0.88	0.88	0.88	192
weighted avg	0.89	0.89	0.89	192