به نام هستی بخش

گزارش پروژه

درس شناسایی الگو

نویسنده: حمیدرضا ابوئی

شماره دانشجویی: ۴۰۲۶۱۷۵۰۹

استاد: دکتر دلیری

۹ دی ۱۴۰۲

توضیح دیتاست

دیتاست استفاده شده در این پروژه Daily and Sports Activities می‌باشد که در آدرس زیر قابل دسترسی می‌باشد:

https://archive.ics.uci.edu/dataset/256/daily+and+sports+activities

مجموعه داده شامل داده های حسگر حرکت ۱۹ فعالیت روزانه و ورزشی است که هر کدام توسط ۸ نفر به سبک خاص خود به مدت ۵ دقیقه انجام می شود. پنج سنسور Xsens MTx روی تنه، بازوها و پاها استفاده می شود. هر یک از ۱۹ فعالیت توسط هشت نفر (۴ زن، ۴ مرد، بین ۲۰ تا ۳۰ سال) به مدت ۵ دقیقه انجام می شود. مدت زمان کل سیگنال برای هر فعالیت هر موضوع ۵ دقیقه است.

از آزمودنی ها خواسته می شود تا فعالیت ها را به سبک خود انجام دهند و محدودیتی در نحوه انجام فعالیت ها وجود نداشته باشد. به همین دلیل، تغییرات بین فردی در سرعت و دامنه برخی از فعالیت‌ها وجود دارد.

این فعالیت ها در سالن ورزشی دانشگاه بیلکنت، در ساختمان مهندسی برق و الکترونیک و در یک منطقه مسطح در فضای باز در محوطه دانشگاه انجام می شود. واحدهای حسگر برای بدست آوردن داده در فرکانس نمونه برداری ۲۵ هرتز کالیبره شده اند. سیگنال های ۵ دقیقه ای به بخش های ۵ ثانیه ای تقسیم می شوند به طوری که ۴۸۰ (=۶۰*۸) بخش سیگنال برای هر فعالیت به دست می آید.

۱۹ فعالیت عبارتند از:

- نشسته (A1)
- ایستاده (A2)
- دراز کشیدن به پشت و سمت راست (A3 و A4)
- بالا و پایین رفتن از پله ها (A5 و A6)
- ایستادن در آسانسور ثابت (A7)
- و حرکت در آسانسور (A8)،
- قدم زدن در پارکینگ (A9)
- راه رفتن روی تردمیل با سرعت ۴ کیلومتر در ساعت (در موقعیت های صاف و شیب دار ۱۵ درجه) (A1
- ۰ و A11)،
- دویدن روی تردمیل با سرعت ۸ کیلومتر در ساعت (A12)
- ورزش روی استپر (A13)
- ورزش بر روی یک کراس مربی (A14)،
- دوچرخه سواری روی دوچرخه ورزشی در موقعیت های افقی و عمودی (A15 و A16)
- قایقرانی (A17)
- پرش (A18)،
- و بازی بسکتبال (A19).

ساختار فایل:

- ۱۹ فعالیت (a) (به ترتیب ذکر شده در بالا)
- ۸ آزمودنی (p)

- ۶۰ بخش(s)

  - ۵ واحد روی تنه (T)، بازوی راست (RA)، بازوی چپ (LA)، پای راست (RL)، پای چپ (LL)
  - ۹ حسگر در هر واحد (شتاب سنج x، y، z، ژیروسکوپ x، y، z، مغناطیس سنج x، y، z)

پوشه های a01، a02، ...، a19 حاوی داده های ثبت شده از ۱۹ فعالیت است.

برای هر فعالیت، زیرپوشه‌های p1، p2، ...، p8 حاوی داده‌هایی از هر یک از ۸ آزمودنی هستند.

در هر زیر پوشه، ۶۰ فایل متنی s01، s02، ...، s60 وجود دارد که برای هر بخش یک فایل وجود دارد.

در هر فایل متنی، ۵ واحد در ۹ حسگر = ۴۵ ستون و ۵ ثانیه در ۲۵ هرتز = ۱۲۵ ردیف وجود دارد.

هر ستون شامل ۱۲۵ نمونه داده است که از یکی از حسگرهای یکی از واحدها در مدت ۵ ثانیه به دست آمده است.

هر ردیف حاوی داده‌هایی است که از تمام ۴۵ محور حسگر در یک لحظه نمونه‌برداری خاص به دست آمده است که با کاما از هم جدا شده‌اند.

ستون های ۱-۴۵ مربوط به:

T_xacc ،T_yacc ،T_zacc ،T_xgyro ،... ،T_ymag ،T_zmag،

RA_xacc, RA_yacc, RA_zacc, RA_xgyro, ..., RA_ymag, RA_zmag,

LA_xacc ،LA_yacc ،LA_zacc ،LA_xgyro ،... ،LA_ymag ،LA_zmag،

RL_xacc ،RL_yacc ،RL_zacc ،RL_xgyro ،... ،RL_ymag ،RL_zmag،

LL_xacc ،LL_yacc ،LL_zacc ،LL_xgyro ،... ،LL_ymag ،LL_zmag.

یعنی:

ستون‌های ۱-۹ مربوط به سنسورهای واحد ۱ (T) است.

ستون های ۱۰-۱۸ مربوط به سنسورهای واحد ۲ (RA) هستند.

ستون های ۱۹-۲۷ مربوط به حسگرهای واحد ۳ (LA) است.

ستون های ۲۸-۳۶ مربوط به سنسورهای واحد ۴ (RL) است.

ستون های ۳۷-۴۵ با سنسورهای واحد ۵ (LL) مطابقت دارد.

این دیتاست missing value ندارد.

تعریف مساله

حال قصد داریم که با روش‌های مختلف، سعی کنیم activity را از روی این داده‌ها به دست آوریم. در ادامه در صورت اضافه آوردن زمان، میخواهیم تلاش کنیم ببینیم آیا می‌توان با استفاده از این داده‌ها، سعی کنیم و آزمودنی ها را از روی داده‌های فعالیت‌های متفاوت آن‌ها به دست آوریم یا خیر. این قسمت اختیاری تعریف شده و در صورت وقت اضافه، انجام خواهد شد.

فاز صفرم مربوط به خواندن داده‌هاست. همانطور که در بخش توضیح دیتاست توضیح داده‌شده است، داده‌ها در یک ساختار نسبتا پیچیده در فایل‌های متنی ذخیره شده اند. بنابراین جهت استفاده از آن‌ها باید داده‌ها را خواند.

در فاز اول، میخواهیم سعی کنیم بدون هیچ پیش پردازش، طبقه‌بندی را انجام دهیم.

در فاز دوم، از روش‌های کاهش بعد استفاده می‌کنیم تا ویژگی‌های برتر را مشخص کرده و طبقه‌بندی کنیم.

در فاز سوم، قصد داریم با اعمال پیش‌پردازش و استخراج ویژگی، مفاهیم معنی داری از داده‌ها استخراج کرده و با استفاده از آن‌ها داده‌ها را طبقه‌بندی کنیم و با حالتی که هیچ پیش پردازشی اعمال نکردیم،

در فاز چهارم قصد داریم پس از اعمال پیش‌پردازش و استخراج ویژگی، ویژگی‌های برتر را با استفاده از روش های کاهش بعد مشخص کرده و با استفاده از آنها طبقه‌بندی را انجام دهیم.

فاز صفرم – خواندن داده‌ها

قبل از خواندن همه داده‌ها، باید ببینیم چگونه می‌توان یک فایل را خواند. بدین منظور می‌توانیم از کتابخانه‌ی pandas استفاده کنیم.

```
df = pd.read_csv(data_path + '/data/a01/p1/s01.txt', header=None)
```

مشاهده می‌شود که یک دیتابیس شامل ۴۵ ستون و ۱۲۵ سطر قابل رویت می‌باشد.

برای این فاز، از سیستم‌عامل و کتابخانه‌ی os و حرکت بین پوشه‌ها استفاده می‌کنیم. داده‌ها در سه سطح پخش شده اند. بنابراین می‌توان همه داده‌ها را با استفاده از سه حلقه‌ی زیر خواند:

```
data_list = []
patient_label_list = []
activity_label_list = []
os.chdir(data_path+"\data\\")
activity_list = os.listdir()

for activity in activity_list:
    print(activity)
    os.chdir(activity)
    patients_list = os.listdir()

    for patient in patients_list:
        os.chdir(patient)
```

```
        segment_list = os.listdir()

        for segment in segment_list:
            data_list.append(read_my_data(segment))
            patient_label_list.append(patient)
            activity_label_list.append(activity)

        os.chdir('../')
    os.chdir('../')
os.chdir('../')
```

فاز اول – بدون پیش‌پردازش، بدون کاهش بعد

در فاز اول، سعی داریم داده‌ها را بدون پیش‌پردازش استفاده کنیم. بنابراین در فاز صفرم از تابع زیر برای خواندن داده‌ها استفاده کردیم:

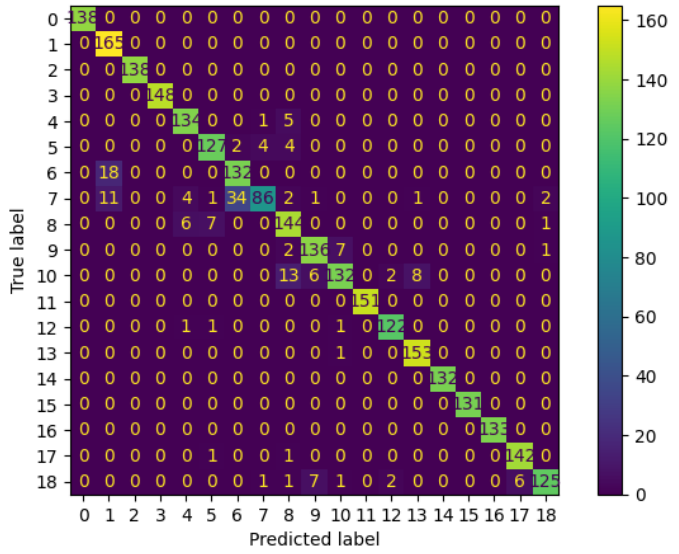
```
def read_my_data(filename):
    """
    A function which will take the text file name and return the data
    """
    df = pd.read_csv('{}'.format(filename), header=None)
    flatten_data = df.to_numpy().flatten()
    return flatten_data
```

ابعاد داده مطابق توضیحات دیتاست برابر است با (۹۱۲۰, ۵۶۲۵)

در ادامه سعی داریم که با استفاده از این دیتاست اولیه، داده‌ها را طبقه‌بندی کنیم. بدین منظور ابتدا داده‌های تست و ترین را از هم جدا می‌کنیم و سپس مدل SVM را بر روی آن آموزش می‌دهیم. از الگوریتم one-vs-one استفاده می‌کنیم تا بتوانیم چند کلاسه پیاده سازی کنیم.

```
X_train, X_test, y_train, y_test = train_test_split(data_list,
activity_label_list, test_size=0.3)

svc_model=SVC()
ovo_model=OneVsOneClassifier(svc_model)
ovo_model.fit(X_train,y_train)
y_pred = ovo_model.predict(X_test)
fig, ax = plt.subplots(figsize=(8, 5))
cmp = ConfusionMatrixDisplay(confusion_matrix(y_test, y_pred))
cmp.plot(ax=ax)
plt.show()
print(accuracy_score(y_test, y_pred))
```

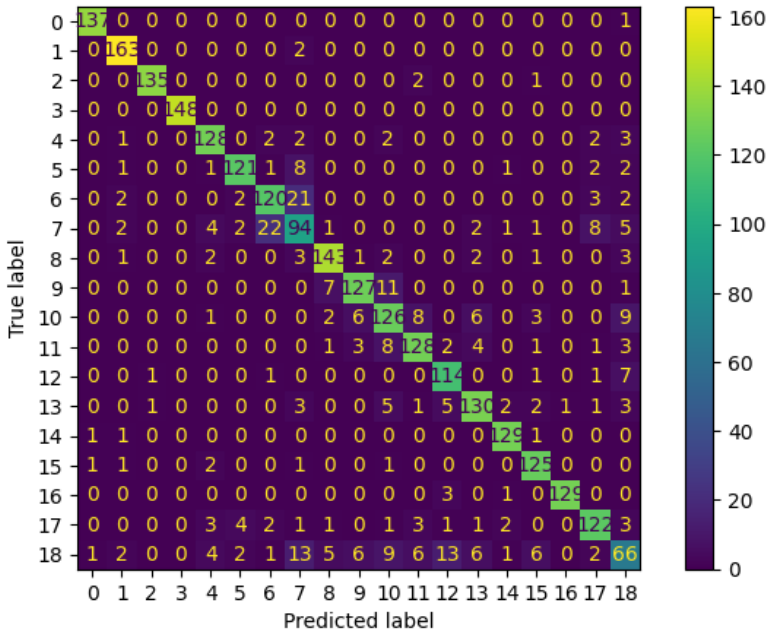
کانفیوژن ماتریس این مدل و داده ها را رسم می‌کنیم:



دقت این مدل برابر است با ۰.۹۳۸۹۶

در ادامه از دو نوع طبقه‌بند دیگر نیز استفاده می‌کنیم. ابتدا از درخت تصمیم استفاده شده است:

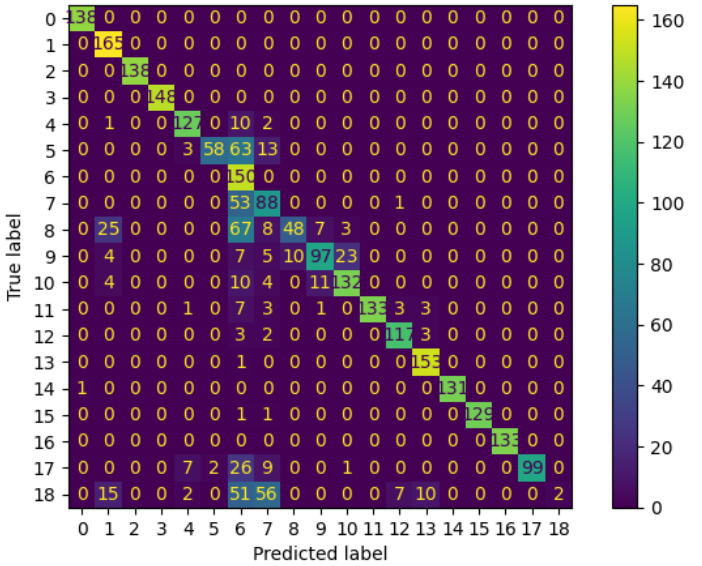
```
model = DecisionTreeClassifier()
model.fit(X_train, y_train)
y_pred = model.predict(X_test)
```



که دارای دقت ۰.۸۷۱۷۱ می‌باشد.

و سپس از طبقه‌بند KNN استفاده می‌کنیم. مقادیر k برابر ۳ و ۵ داده شد که دقت نهایی k=3 بیشتر بود بنابراین از این k برای آموزش استفاده شد.

```
neigh = KNeighborsClassifier(n_neighbors=3)
neigh.fit(X_train, y_train)
y_pred = neigh.predict(X_test)
```



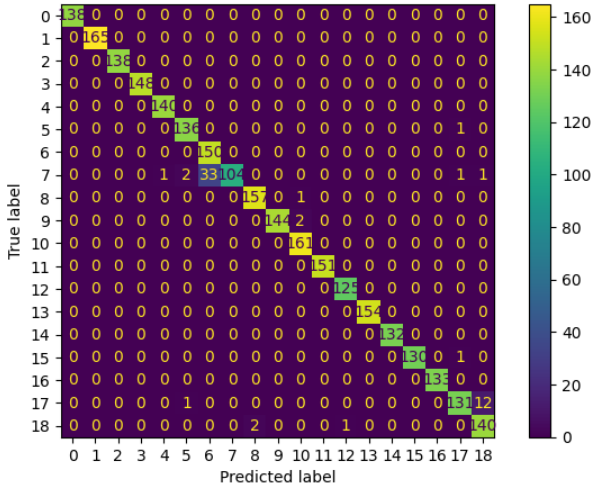
که دارای دقت ۰.۷۹۸۹۷ می‌باشد.

در ادامه قصد داریم که از نرمالیزیشن استفاده کنیم تا ببینیم آیا تغییراتی مشاهده می‌شود یا خیر.

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

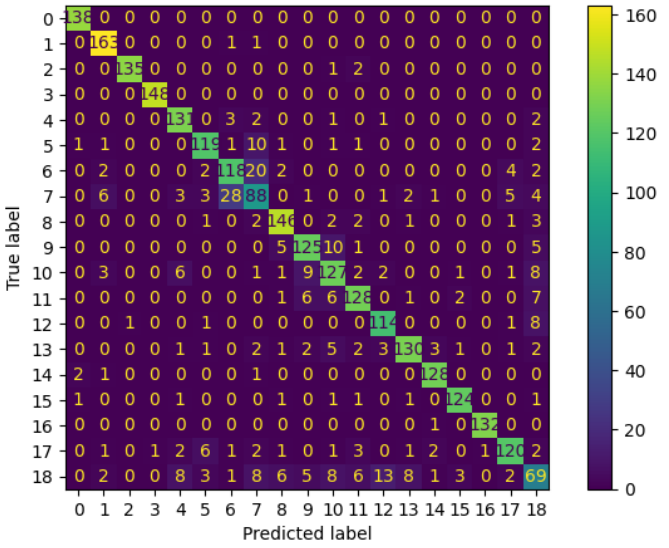
در ادامه از این دیتا برای آموزش مدل‌های SVM، DT و KNN استفاده شد که کانفیوژن ماتریس و دقت آن‌ها به شرح زیر می‌باشد:

ماشین بردار پشتیبان:



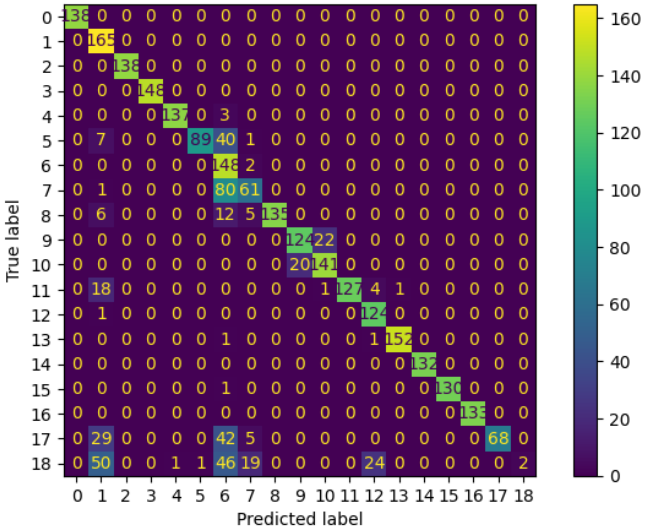
دقت برابر است با: ۰.۹۷۸۴۳

درخت تصمیم:



دقت برابر است با: ۰.۸۷۰۹۷

۳=K نزدیک‌ترین همسایگی:



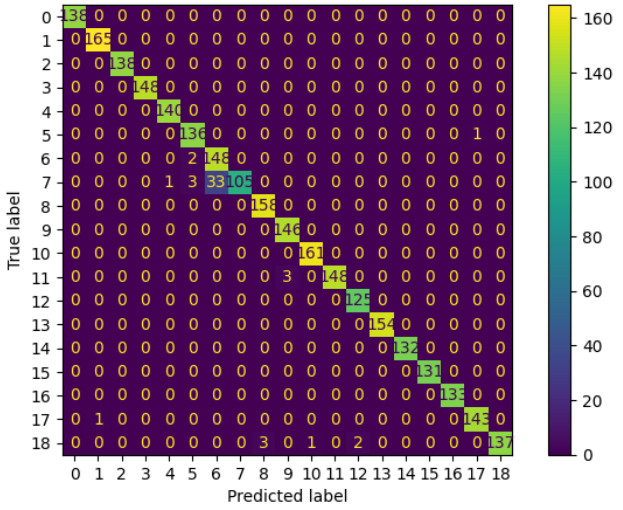
دقت برابر است با: ۰.۸۳۷۷۱

فاز دوم – بدون پیش‌پردازش، با کاهش بعد

در ادامه از روش pca برای کاهش بعد استفاده می‌شود. بدین منظور سعی داریم ۱۰۰ کامپوننت نهایی که تقریبا برابر با ۱ درصد مشاهدات ما می‌باشد را انتخاب کنیم و از آن برای آموزش مدل‌هایمان استفاده می‌کنیم:

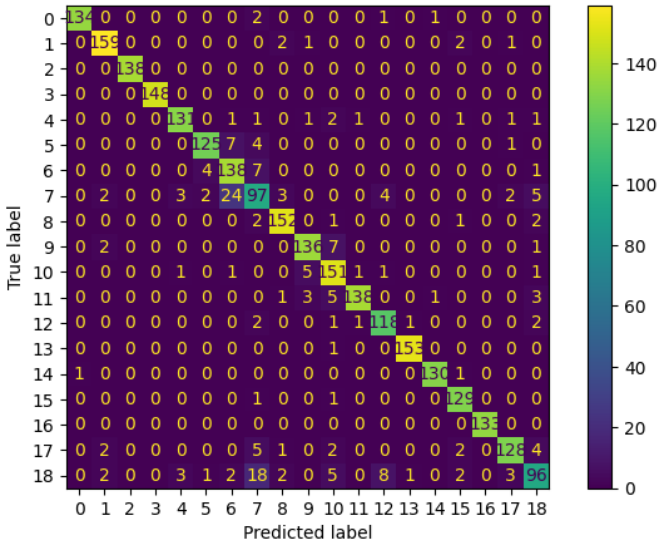
```
pca = PCA(n_components = 100) # Number of features is 1% of observations.
X_train_pca = pca.fit_transform(X_train_normalized)
X_test_pca = pca.transform(X_test_normalized)
```

ماشین بردار پشتیبان:



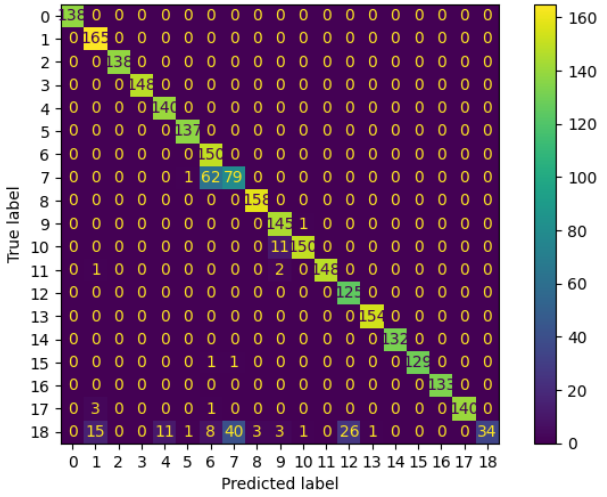
دقت برابر است با: ۰.۹۸۱۷۲

درخت تصمیم:



دقت برابر است با: ۰.۹۲۶۱۶

K نزدیک‌ترین همسایگی:



دقت برابر است با: ۰.۹۲۹۴۵

فاز سوم – با پیش‌پردازش، بدون کاهش بعد

در این فاز ما قصد داریم که از این داده‌های ما که سیگنال می‌باشد، ویژگی استخراج کنیم. ویژگی‌های استخراج شده شامل ویژگی‌های آماری و فرکانسی می‌باشد. ویژگی‌های آماری از جمله کمینه، بیشینه، میانگین، پیک، انحراف استاندارد، چولگی ، کورتوسیس ، توان سیگنال و rms می‌باشد. ویژگی‌های فرکانسی از fft سیگنال حاصل شده که شامل حداکثر، مجموع، میانگین، واریانس، پیک، چولگی و کورتوسیس سیگنال فرکانسی می‌باشد:

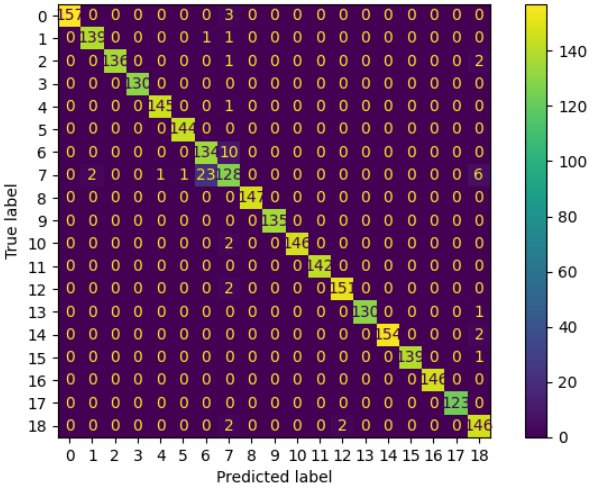
```
def read_data_and_feature_extraction(filename):
    """
    A function which will take the text file name and return the data that
feature extraction
    """
    df = pd.read_csv('{}'.format(filename), header=None)
    X = df.values

    statistical_features =
list(df.min())+list(df.max())+list(df.mean())+list(np.max(df.abs(),axis=0))+list(
df.std())+list(df.skew())+list(df.kurtosis())

    statistical_features.extend(np.mean(X**2,axis=0))#+list(np.sqrt(np.mean(X**2)
))
    statistical_features.extend(np.sqrt(np.mean(X**2,axis=0)))
    ft = fft(X)
    S = np.abs(ft**2)/len(df)
    frequency_features = np.max(S,axis=0)
    frequency_features = np.concatenate((frequency_features,np.sum(S,axis=0)))
    frequency_features = np.concatenate((frequency_features,np.mean(S,axis=0)))
    frequency_features = np.concatenate((frequency_features,np.var(S,axis=0)))
```

```
    frequency_features =
np.concatenate((frequency_features,np.max(np.abs(S),axis=0)))
    frequency_features =
np.concatenate((frequency_features,stats.skew(S,axis=0)))
    frequency_features =
np.concatenate((frequency_features,stats.kurtosis(S,axis=0)))

    features = np.concatenate((statistical_features,frequency_features))


    return features
```

در نهایت ۷۲۰ ویژگی استخراج شده است.

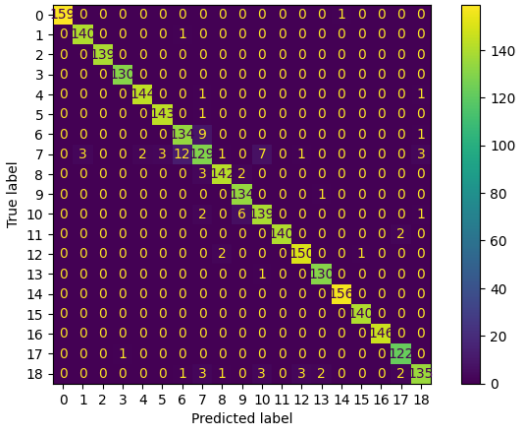
در ادامه، مدل‌های خود را با این ویژگی‌ها آموزش می‌دهیم:

ماشین بردار پشتیبان:



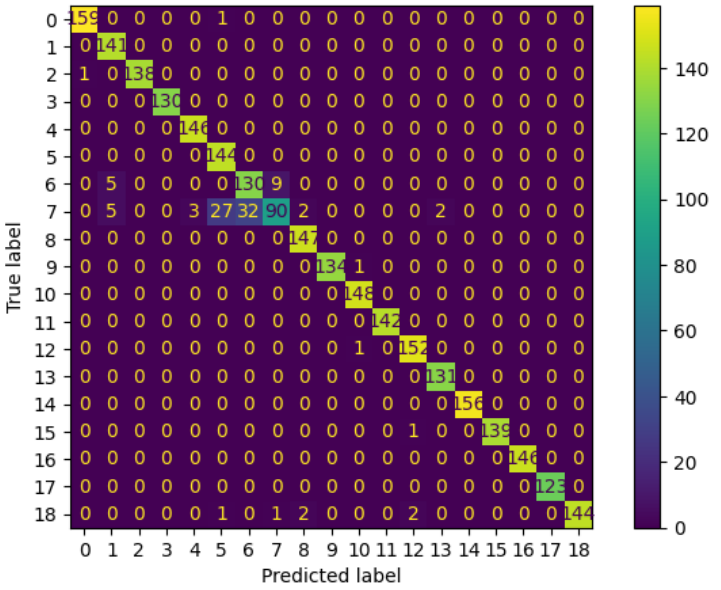
دقت برابر است با: ۰.۹۷۶۶۰

درخت تصمیم:



دقت برابر است با: ۰.۹۶۹۲۹

K نزدیک‌ترین همسایگی:

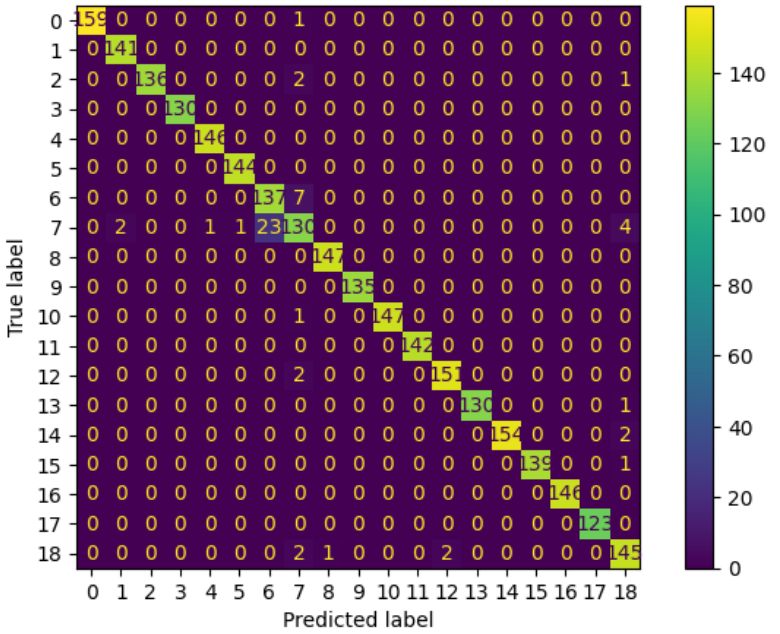


دقت برابر است با: ۰.۹۶۴۹۱

فاز چهارم – با پیش‌پردازش، با کاهش بعد

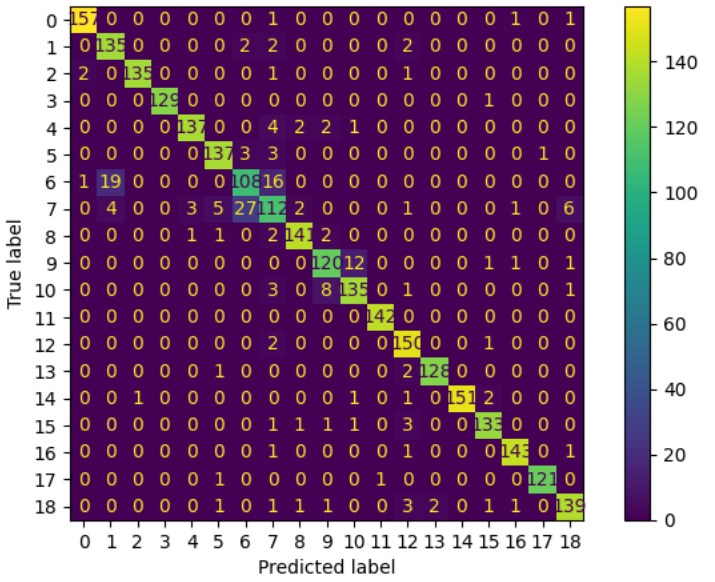
مطابق فاز دوم، از pca برای کاهش ویژگی استفاده می‌کنیم و ۱۰۰ کامپوننت اصلی را استخراج می‌کنیم.

ماشین بردار پشتیبان:



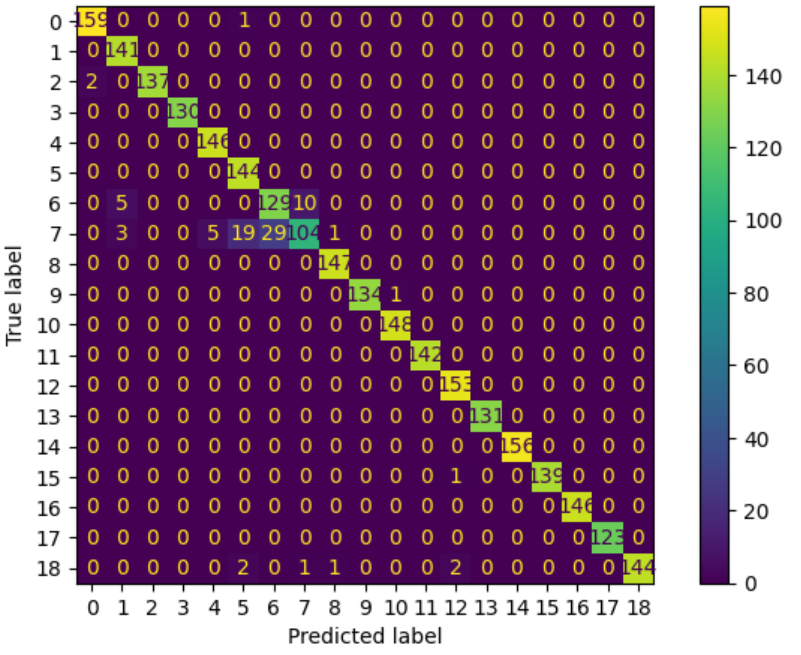
دقت برابر است با: ۰.۹۸۰۲۶

درخت تصمیم:



دقت برابر است با: ۰.۹۳۳۱۱

K نزدیک‌ترین همسایگی:



دقت برابر است با: ۰.۹۶۹۶۶

جمع‌بندی

در نتیجه، می‌توان دقت طبقه‌بندها را مطابق زیر در کنار هم قرار داد و با هم مقایسه کرد.

الگوریتم	فاز اول بدون نرمال‌سازی	فاز اول با نرمال‌سازی	فاز دوم	فاز سوم	فاز چهارم
SVM	۰.۹۳۸۹۶	۰.۹۷۸۴۳	۰.۹۸۱۷۲	۰.۹۷۶۶۰	۰.۹۸۰۲۶
DT	۰.۸۷۱۷۱	۰.۸۷۰۹۷	۰.۹۲۶۱۶	۰.۹۶۹۲۹	۰.۹۳۳۱۱
KNN	۰.۷۹۸۹۷	۰.۸۳۷۷۱	۰.۹۲۹۴۵	۰.۹۶۴۹۱	۰.۹۶۹۶۶

نتیجه می‌گیریم که نرمال‌سازی بر روی درخت تصمیم کمترین تاثیر را دارد. همچنین می‌توان نتیجه گرفت که با استخراج ویژگی‌های معنی دار از داده‌های موجود، به طور کلی می‌توان دقت طبقه‌بند را بالا برد.

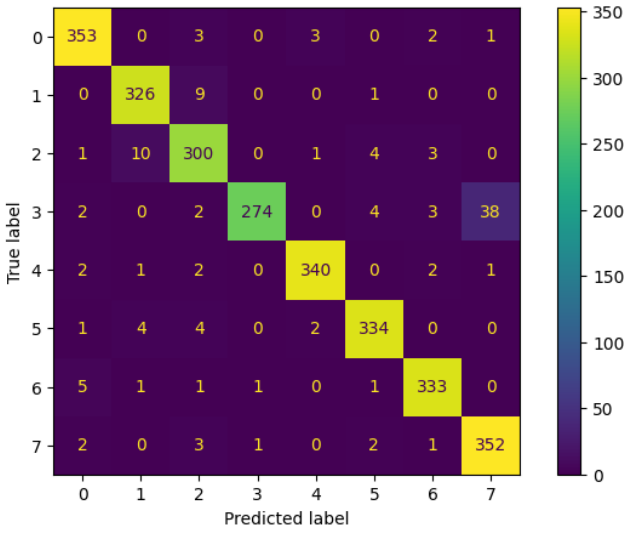
قسمت اختیاری پروژه

در این قسمت، قصد داریم با استفاده از داده‌های مساله، و بررسی کنیم که آیا می‌توانیم با داده‌های حرکتی افراد، خود فرد را تشخیص دهیم یا خیر. بدین منظور از لیبل افراد استفاده می‌کنیم و از ویژگی‌های استخراج شده در فاز سوم بدون استفاده از pca استفاده می‌کنیم.

```
X_train, X_test, y_train, y_test = train_test_split(data_list,
patient_label_list, test_size=0.3)
scaler = StandardScaler()
scaler.fit(X_train)
X_train_normalized = scaler.transform(X_train)
X_test_normalized = scaler.transform(X_test)
```

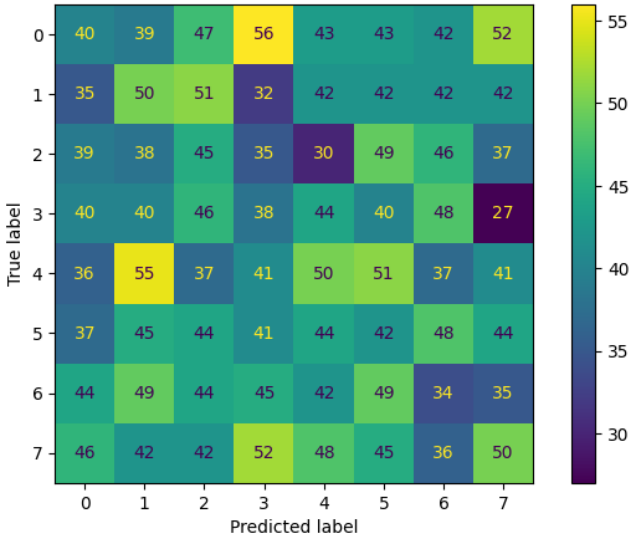
در این قسمت از همان سه طبقه‌بند قبلی استفاده می‌کنیم:

ماشین بردار پشتیبان:



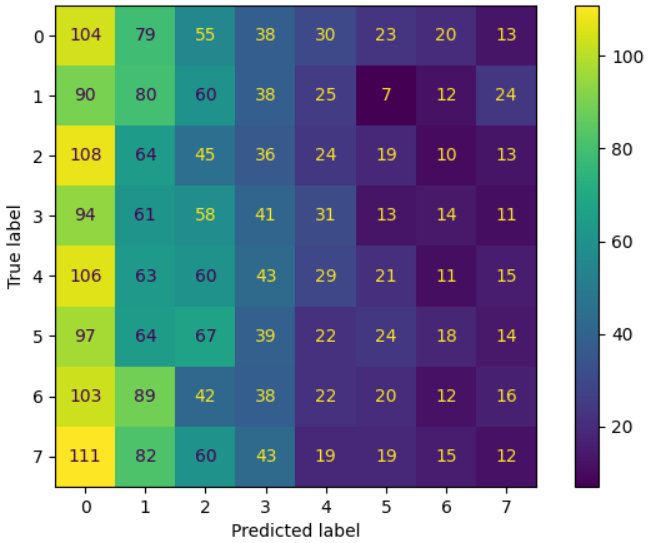
دقت برابر است با: ۰.۹۵۴۶۷

درخت تصمیم:



دقت برابر است با: ۰.۱۲۷۵۵

K نزدیک‌ترین همسایگی:

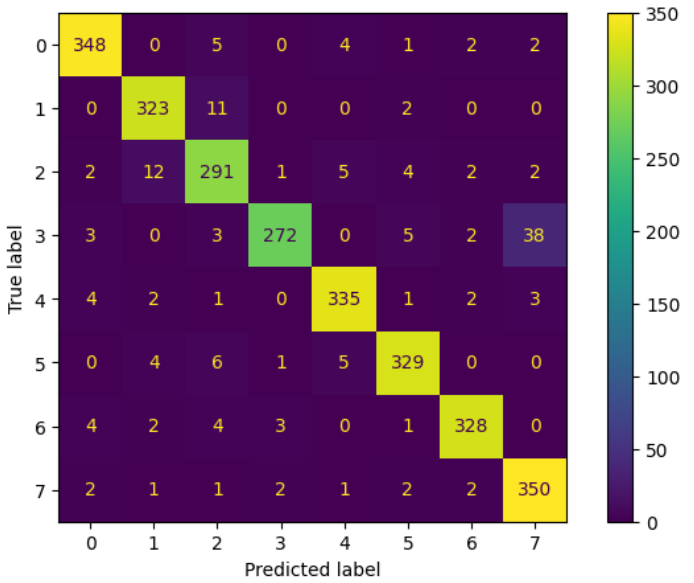


دقت برابر است با: ۰.۱۲۶۸۲

مشاهده می‌شود که می‌توان با استفاده از SVM به خوبی افراد را از روی نحوه‌ی تمرین و حرکات بدن تشخیص داد ولی الگوریتم‌های دیگر مناسب نیستند.

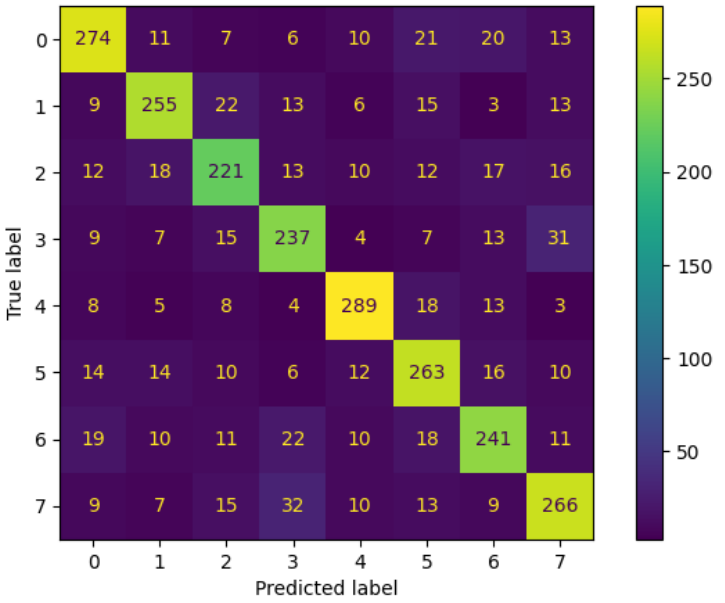
پس از انتخاب ویژگی با استفاده از pca نتایج به شرح زیر می‌باشد:

ماشین بردار پشتیبان:



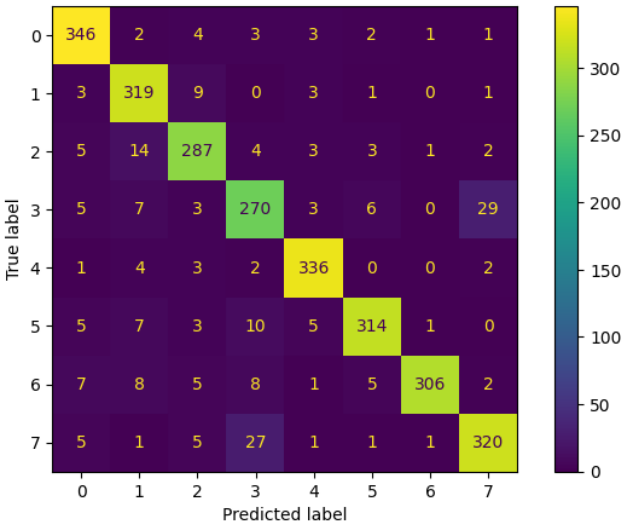
دقت برابر است با: ۰.۹۴۱۵۲

درخت تصمیم:



دقت برابر است با: ۰.۷۴۷۸۰

K نزدیک‌ترین همسایگی:



دقت برابر است با: ۰.۹۱۳۰۱

مشاهده می‌شود که استفاده از pca به عنوان کاهش ویژگی، تاثیر به سزایی روی نتیجه‌ی دو طبقه‌بند DT و KNN می‌گذارد و دقت آن‌ها را به شدت افزایش می‌دهد.

با تشکر.