

سید محمد حسینی

۹۸۲۱۲۵۳

کولب:

<https://colab.research.google.com/drive/1Gzv3h8dCmJ3Lxfrhlieyu3e9XdTRbUXR?usp=sharing>

$$L(w, b, \alpha, r, \epsilon, C) = \frac{1}{2} \|w\|^2 - \sum_{i=1}^N \alpha_i (y_i (w^T x_i + b) - 1 + \epsilon_i) - \sum_{i=1}^N r_i \epsilon_i + C \sum_{i=1}^N \epsilon_i^2$$

$$\frac{\partial L}{\partial w} = 0 \Rightarrow w = \sum_{i=1}^N \alpha_i y_i x_i$$

$$\frac{\partial L}{\partial b} = 0 \Rightarrow \sum_{i=1}^N \alpha_i y_i = 0$$

$$\frac{\partial L}{\partial \epsilon_i} = 0 \Rightarrow -\alpha_i - r_i + 2C\epsilon_i = 0 \Rightarrow \epsilon_i = \frac{\alpha_i + r_i}{2C}$$

می دانیم

$$\alpha_i (y_i (w^T x_i + b) - 1 + \epsilon_i) = 0, \quad r_i \epsilon_i = 0$$

$$\Rightarrow \alpha_i = 0 \quad \text{if} \quad y_i (w^T x_i + b) - 1 + \epsilon_i = 0$$

$$r_i = 0 \quad \text{if} \quad \epsilon_i = 0$$

$$\text{if } \epsilon_i = 0 \Rightarrow \alpha_i = 0 \quad (*)$$

$$\text{if } \epsilon_i > 0 \Rightarrow r_i = 0, \quad \epsilon_i = \frac{\alpha_i}{2C} \quad (I)$$

$$y_i (w^T x_i + b) - 1 + \epsilon_i = 0 \xrightarrow{(I)} \frac{\alpha_i}{2C} + y_i (w^T x_i + b) - 1 = 0 \Rightarrow \alpha_i = C(1 - y_i (w^T x_i + b)) \quad (II)$$

$$\epsilon_i > 0 \xrightarrow{(II)} \alpha_i > 0 \quad (**)$$

از این دید  $\max(\alpha_i)$  را می توانیم پیدا کنیم.  $y_i (w^T x_i + b) = 0$  باشد پس

$$\max(\alpha_i) = C \xrightarrow{(**)} 0 \leq \alpha_i \leq C$$

$$S_T = S_B + S_W$$

$$S_B = \sum_i^C (\mu_i - \mu)(\mu_i - \mu)^T N_i$$

$$S_W = \sum_j^{N_i} (x_j - \mu_i)(x_j - \mu_i)^T$$

$$S_T = \sum_i^N (x_i - \mu)(x_i - \mu)^T$$

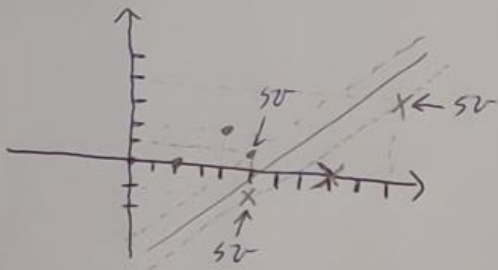
$$x_i = \mu_{c_i} + (x_i - \mu_{c_i}) \Rightarrow S_T = \sum (\mu_{c_i} + (x_i - \mu_{c_i}) - \mu) \times$$

$$\begin{aligned} \Rightarrow S_T = \sum & (\mu_{c_i} - \mu)(\mu_{c_i} - \mu)^T + (\mu_{c_i} - \mu)(x_i - \mu)^T + (x_i - \mu_{c_i})(\mu_{c_i} - \mu)^T \\ & + (x_i - \mu_{c_i})(x_i - \mu_{c_i})^T \end{aligned}$$

با تفکیک این اثرها به فرمول های  $S_W$  و  $S_B$  ، اینها می شود که  $S_T = S_W + S_B$

$$\phi(x_1) = \{(2,0), 1\}, \phi(x_2) = \{(5,1), 1\}, \phi(x_3) = \{(4,2), 1\}$$

$$\phi(x_4) = \{(5,-1), -1\}, \phi(x_5) = \{(8,0), -1\}, \phi(x_6) = \{(10,4), -1\}$$



(الف)

$$\text{Dual } L(\lambda) = \max \sum_{i=1}^6 \alpha_i - \frac{1}{2} \sum_{i=1}^6 \sum_{j=1}^6 \alpha_i \alpha_j y_i y_j k(\phi(x_i), \phi(x_j))$$

s.t.  $\alpha_i \geq 0, \sum_{i=1}^6 \alpha_i y_i = 0$

این به یکی ضرب داخلی، در اینجا =، و متوازن =

$$K = \begin{bmatrix} 4 & 10 & 8 & 10 & 16 & 20 \\ & 26 & 22 & 24 & 40 & 54 \\ & & 20 & 18 & 32 & 48 \\ & & & 26 & 40 & 46 \\ & & & & 64 & 80 \\ & & & & & 116 \end{bmatrix}$$

متوازن

حال به منظور حل این مسئله بهینه سازی، از آن استفاده می کنیم:

کد مورد نظر در در فایل پایتون پیوست قرار گرفته است که به موجب آن مسئله بهینه سازی dual حل شده است و نتایج نمایش داده شده اند.

نتایج آلفا، وزن و بایاس هب شرح زیر می باشد:

Optimal solution found.

Alphas: [4.29184904e-08 9.99999939e-01 8.33386184e-09 7.99999949e-01

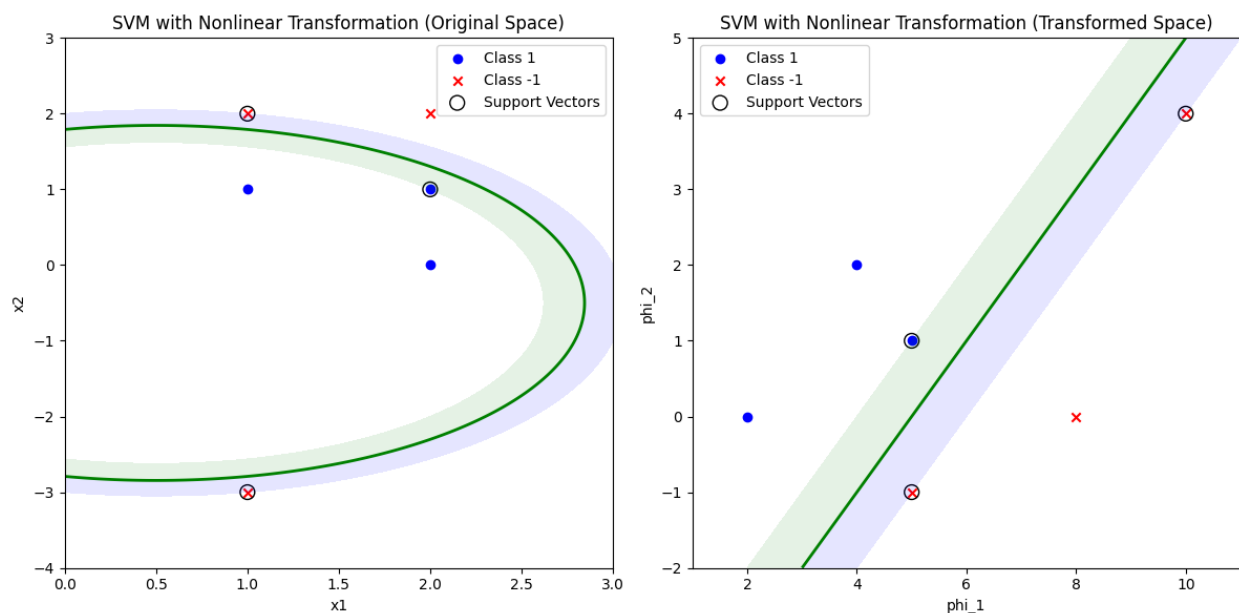
9.19957782e-08 1.99999950e-01]

Support vectors: [1 3 5]

Weight vector w: [-0.99999955 1.00000009]

Bias b: 4.999996855890123

نمایش نتایج در فضای latent و فضای اولیه به شرح زیر می باشد.



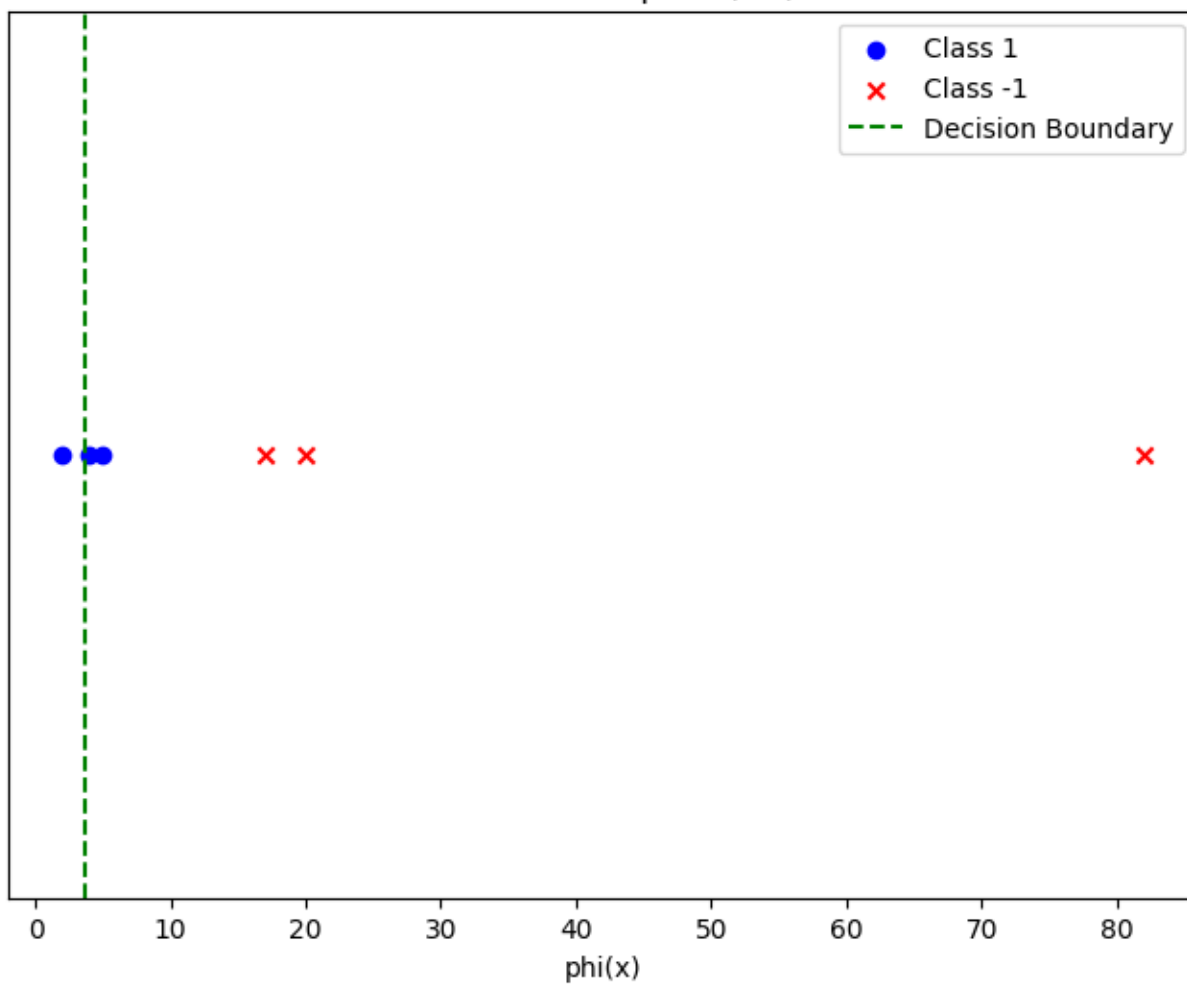
تصویر سمت راست که مربوط به فضای latent است نشان می دهد که شهود ما از بردارهای پیشنیان در بخش پیشین سوال درست بوده است.

حال به معرفی یک تبدیل غیر خطی که به فضای یک بعدی برویم اما تفکیک پذیری باقی بماند. بدین منظور از محور  $x$  نمودار بالا سمت راست الهام می گیریم که به جز در یک نقطه، در باقی نقاط تفکیک پذیری را ایجاد کرده بنابراین با یک تغییر کوچک در فرمول  $x_1^2 + x_2^2$  به نتیجه مطلوب خواهیم رسید. این تبدیل به شرح زیر می باشد:

$$x_1^2 + x_2^4$$

استفاده از کرنل فوق به شرح زیر می باشد:

Transformed Space (1D)

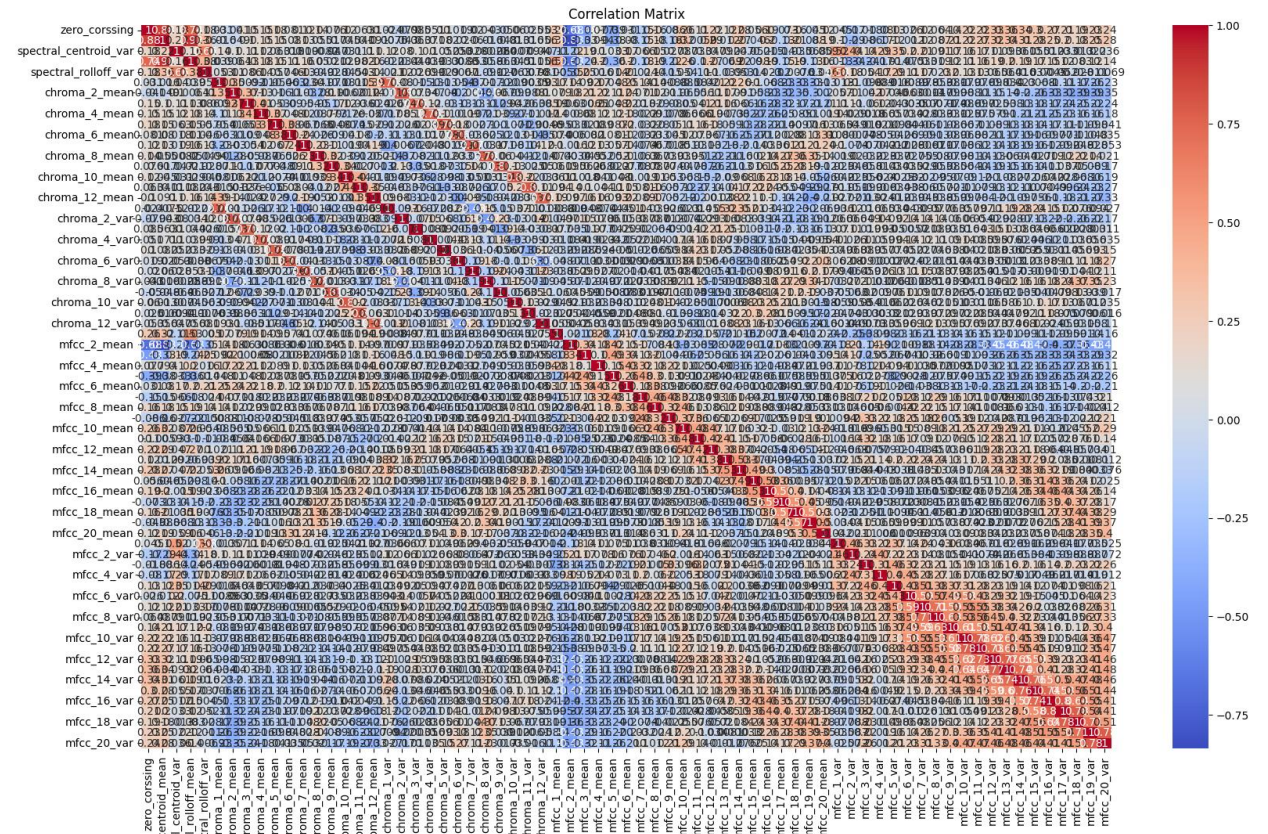




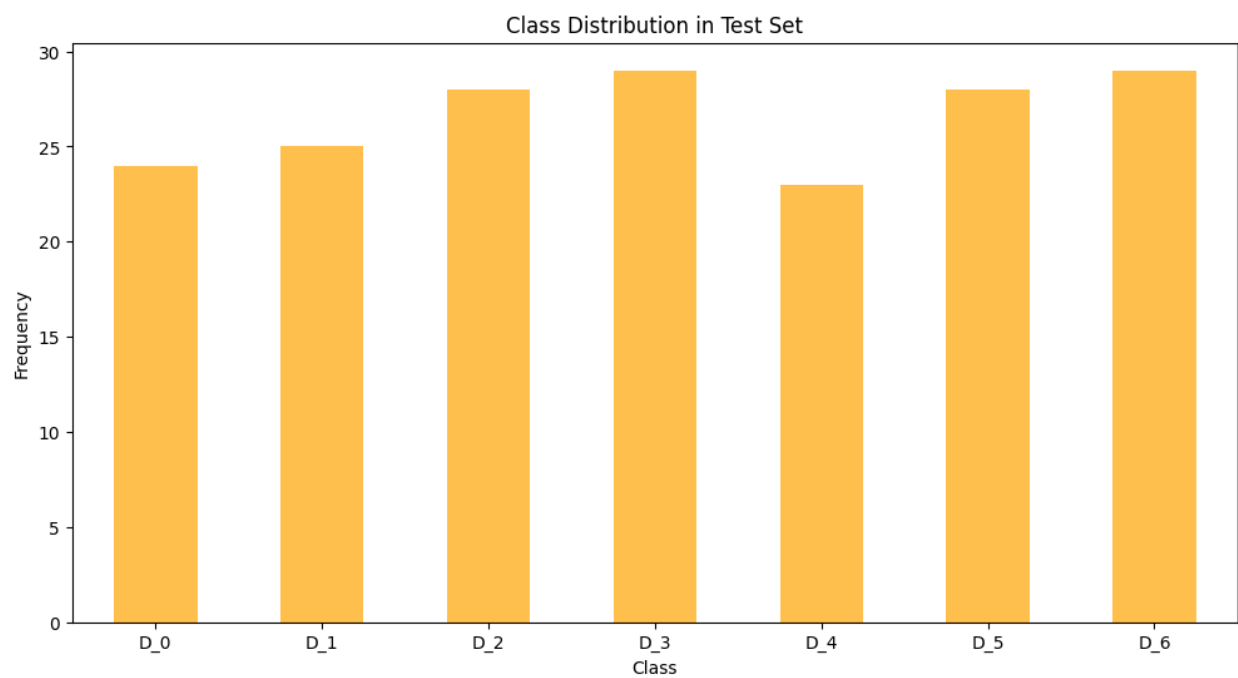
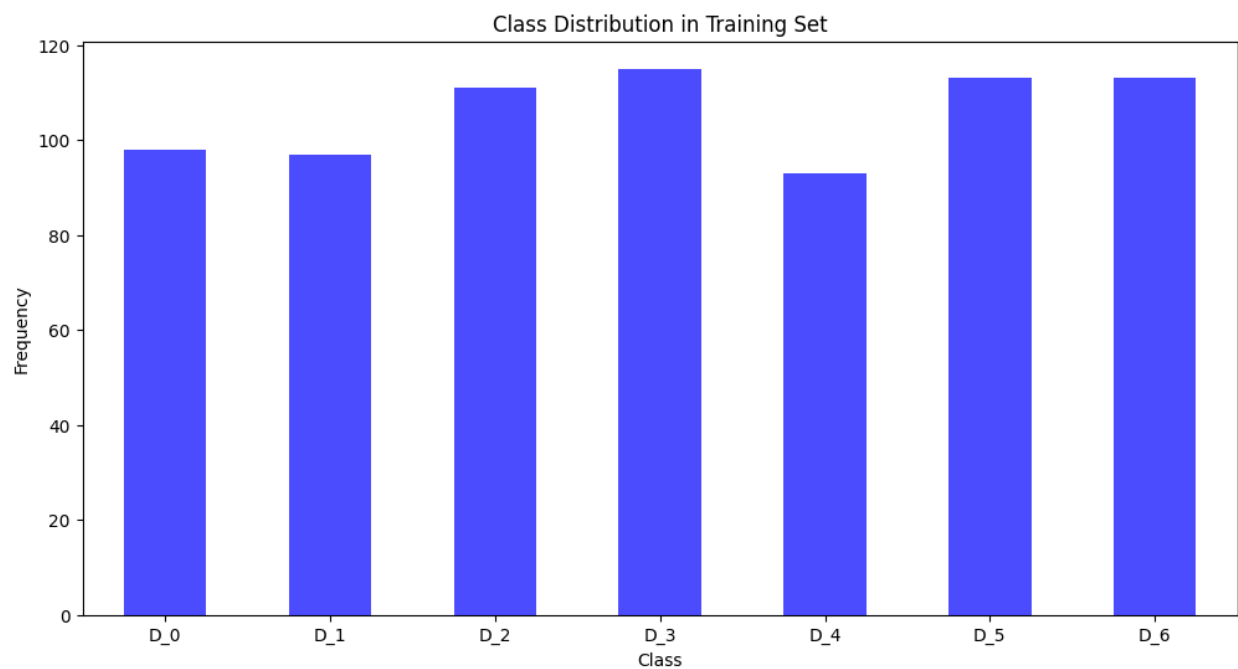
## پرسش ۲

```
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2,
random_state=53, stratify=y)
```

با استفاده از دستور فوق دیتا را با نسبت ۰/۲ به دو زیر گروه آموزش و ارزیابی تقسیم کردیم که با پارامتر stratify توزیع را در این تقسیم حفظ کرده‌ایم.

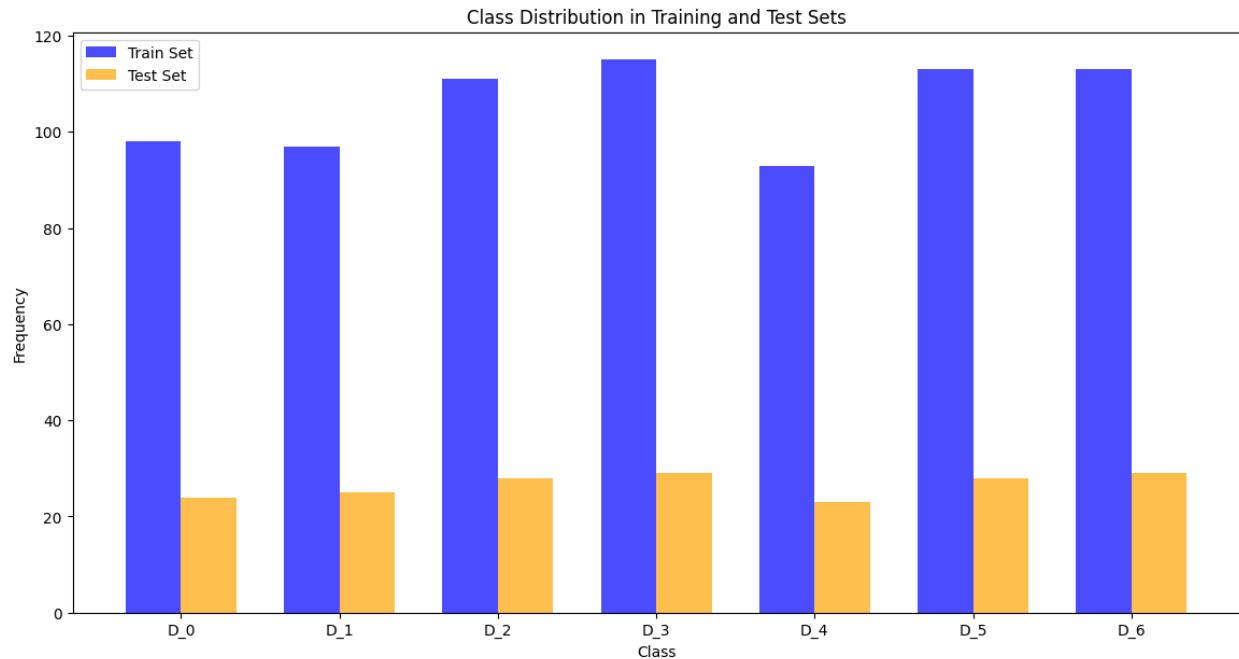


نمودار فوق مربوط به همبستگی رگرسور ها به یکدیگر می‌باشد که با توجه به رنگ این نمودار اکثر رگرسور ها تا حد خوبی نسبت به هم همبستگی ندارند.



دو نمودار فوق مربوط به توزیع داده در دیتاست آموزش و ارزیابی است که به لحاظ نسبت هر کلاس به دیگر بسیار شبیه یکدیگر هستند اما در دامنه هر میله باهم اختلاف دارند.





نمودار زیر توزیع را به صورت مقایسه‌ای آورده است.

در ادامه از الگوریتم RFE استفاده می‌کنیم تا ویژگی‌های مهم را دسته بندی کنیم و نمایش دهیم.

الگوریتم فوق زمان زیادی بابت اجرا می‌گیرد و در طول جلسه امتحان امکان اجرای آن وجود نداشت. بابت نتایج می‌توانید به کد مراجعه کنید.

```
# Use SVM with linear kernel for RFE
```

```
svc = SVC(kernel='linear', C=1)
```

```
rfe = RFE(estimator=svc, n_features_to_select=2, step=1)
```

```
rfe.fit(X_train, y_train)
```

```
# Get RFE results
```

```
ranking = rfe.ranking_
```

```
features_ranking = pd.DataFrame({'Feature': X_train.columns, 'Ranking': ranking})
```

```
features_ranking = features_ranking.sort_values(by='Ranking')
```

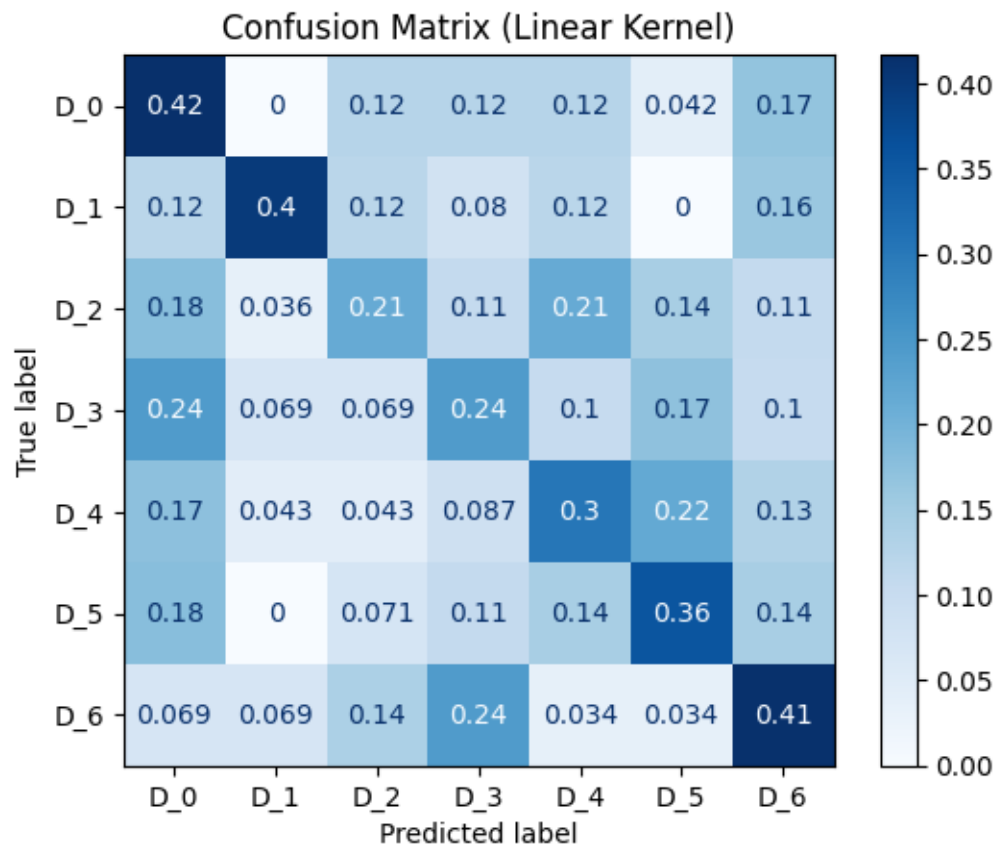
بخش ب)

به منظور ادامه از SVM با کرنل‌های متفاوت استفاده می‌کنیم که به شرح زیر می‌باشد:

```
svm_linear = SVC(kernel='linear')
```

svm\_rbf = SVC(kernel='rbf')

نتایج حاصل شده از آموزش این دو مدل روی دیتا به شرح زیر خواهد بود:

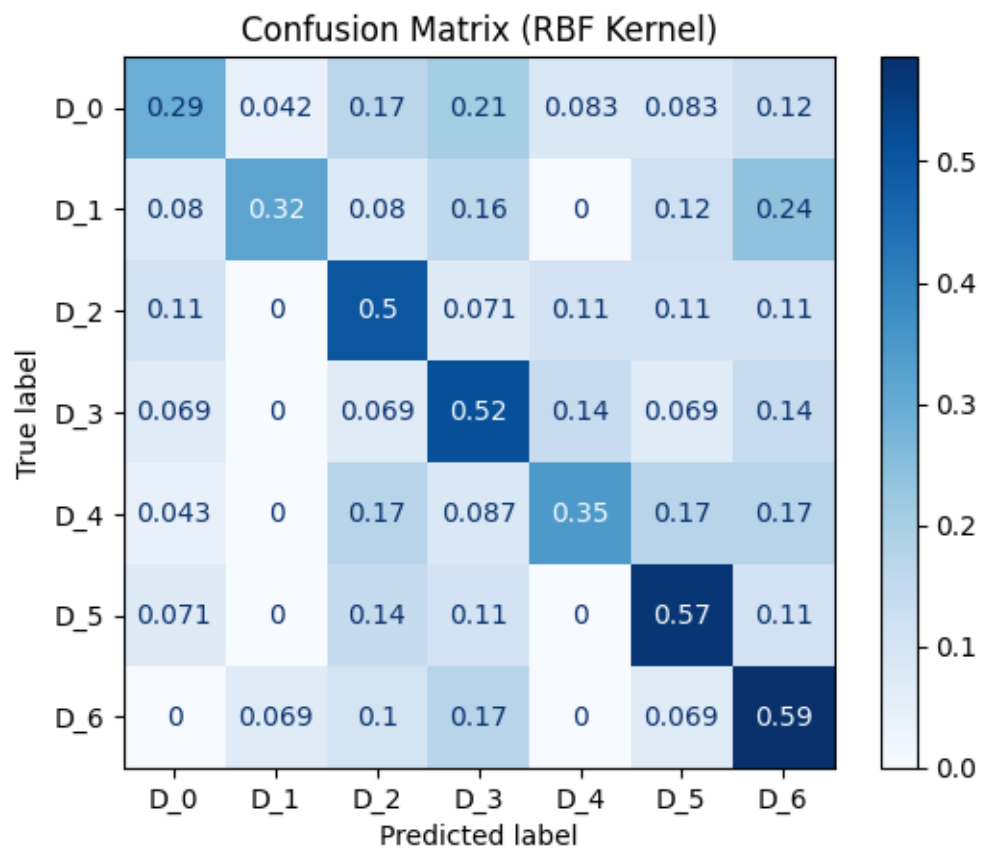


Classification Report for Linear Kernel:

precision recall f1-score support

D_0	0.28	0.42	0.33	24
D_1	0.62	0.40	0.49	25
D_2	0.29	0.21	0.24	28
D_3	0.26	0.24	0.25	29
D_4	0.26	0.30	0.28	23
D_5	0.38	0.36	0.37	28
D_6	0.36	0.41	0.39	29

accuracy 0.33 186  
macro avg 0.35 0.34 0.34 186



### Classification Report for RBF Kernel:

precision recall f1-score support

D_0	0.41	0.29	0.34	24
D_1	0.73	0.32	0.44	25
D_2	0.42	0.50	0.46	28
D_3	0.42	0.52	0.46	29
D_4	0.47	0.35	0.40	23
D_5	0.50	0.57	0.53	28
D_6	0.42	0.59	0.49	29

accuracy 0.46 186

macro avg 0.48 0.45 0.45 186

weighted avg 0.48 0.46 0.45 186

همانطور که از ماتریس درهم‌ریختگی و نتایج accuracy مشخص است، با تغییر کرنل از خطی به RBF دقت مدل افزایش پیدا کرده است که این مسئله در نتیجه غیرخطی بودن کرنل است.

قسمت ج)

تمام جزییات مطرح شده در صورت سوال در بلاک کد پیاده شده است و فرآیندهای متغیر برابر است با:

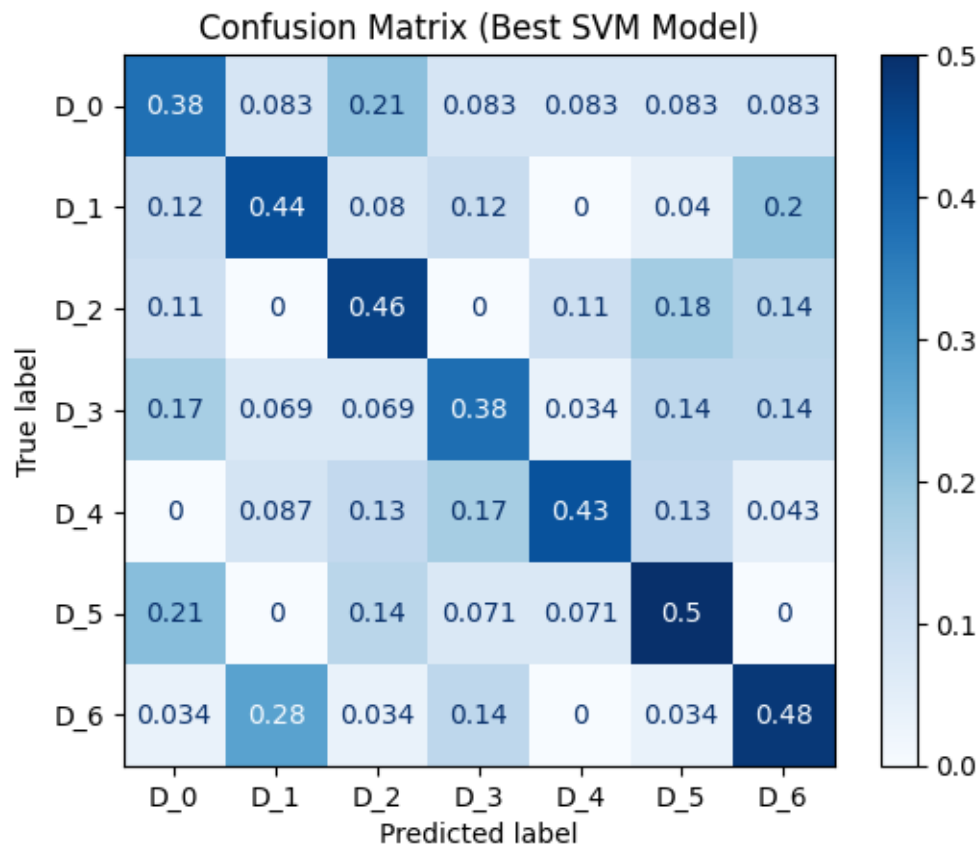
```
param_grid = {  
    'C': [0.1, 1, 10, 100, 1000, 0.01],  
    'gamma': [1, 0.1, 0.01, 0.001],  
    'kernel': ['rbf', "polynomial"]  
}
```

نتیجه جستجوی بهترین مدل عبارت است از:

Fitting 5 folds for each of 32 candidates, totalling 160 fits  
Best Parameters: {'C': 10, 'gamma': 0.01, 'kernel': 'rbf'}

Best Parameters: {'C': 10, 'gamma': 0.01, 'kernel': 'rbf'}

Best Score: 0.41891891891891897



Classification Report:

	precision	recall	f1-score	support
--	-----------	--------	----------	---------

D_0	0.33	0.38	0.35	24
D_1	0.44	0.44	0.44	25
D_2	0.43	0.46	0.45	28
D_3	0.42	0.38	0.40	29
D_4	0.56	0.43	0.49	23
D_5	0.47	0.50	0.48	28
D_6	0.47	0.48	0.47	29

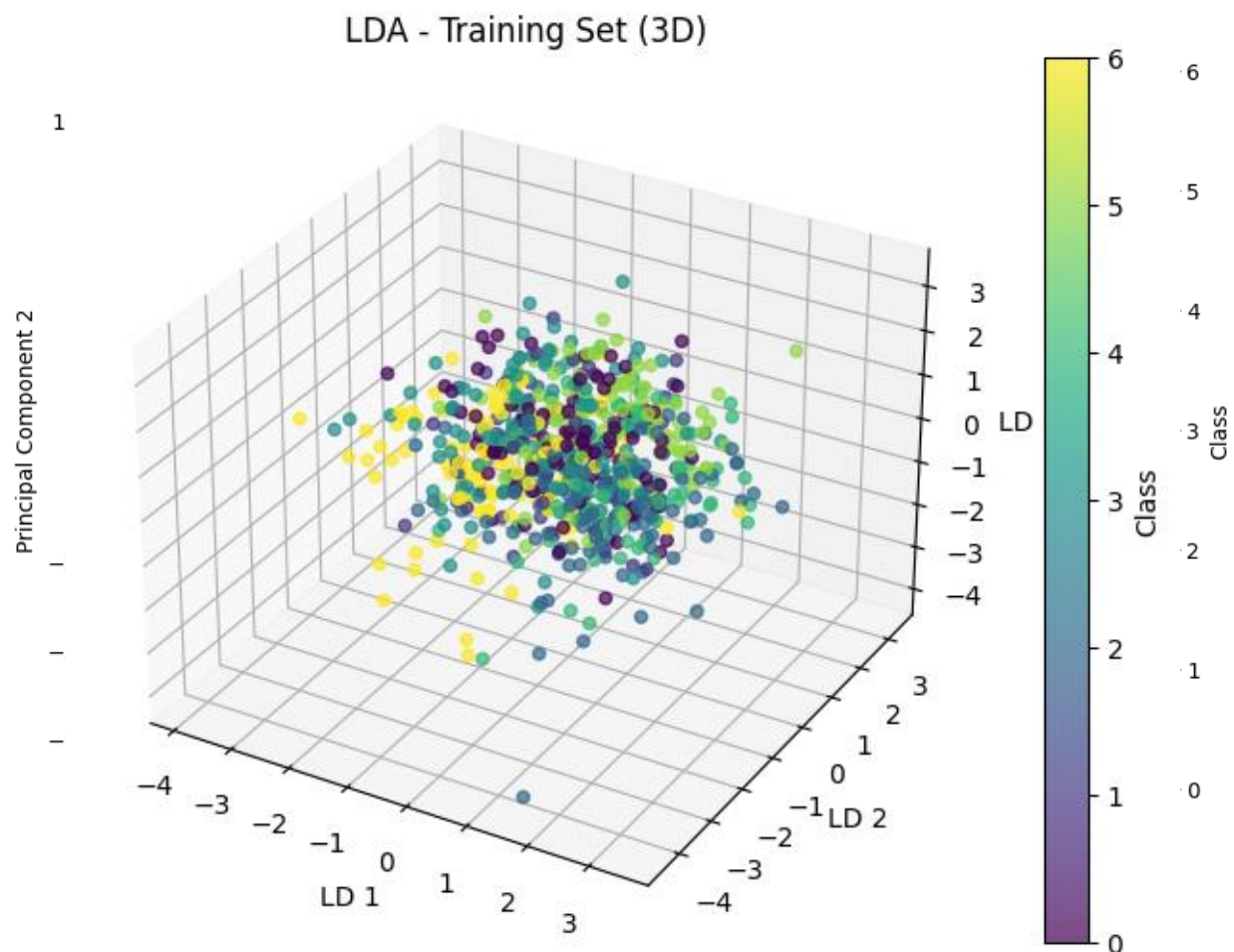
accuracy			0.44	186
macro avg	0.45	0.44	0.44	186
weighted avg	0.45	0.44	0.44	186



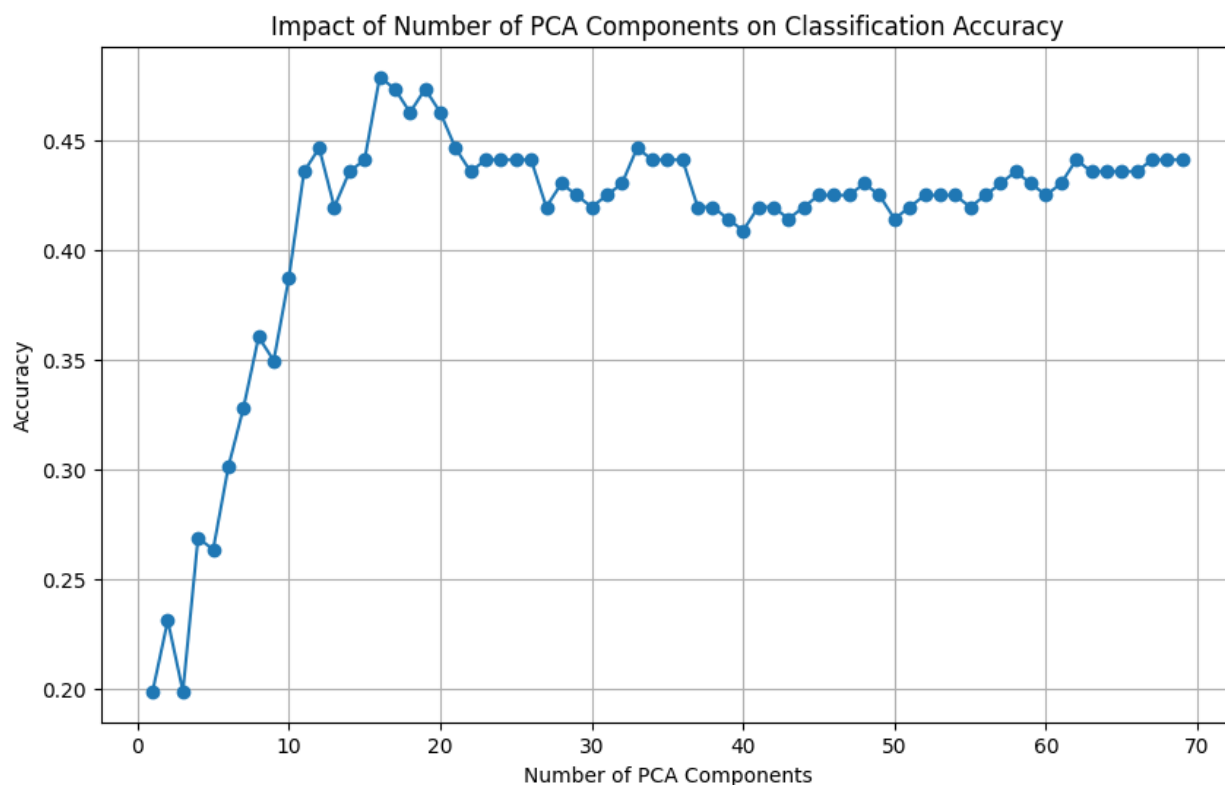
این نکته حائز اهمیت است که مقدار accuracy در مدل RBF با فرآپارامترهای پیش فرض دقت بالاتری کسب کرده است که مقادیر آن در قسمت قبل سوال مشخص شده است.

بخش د)

موارد ذکر شده در صورت سوال به صورت دقیق پیاده سازی شده است که نتایج آن به شرح زیر می باشد:



همانطور که مشخص است کلاس‌ها به ازای دو و سه پارامتر با ارزش دیتاست بسیار در هم تنیده می‌باشند و تفکیک آنها کار دشواری خواهد بود.

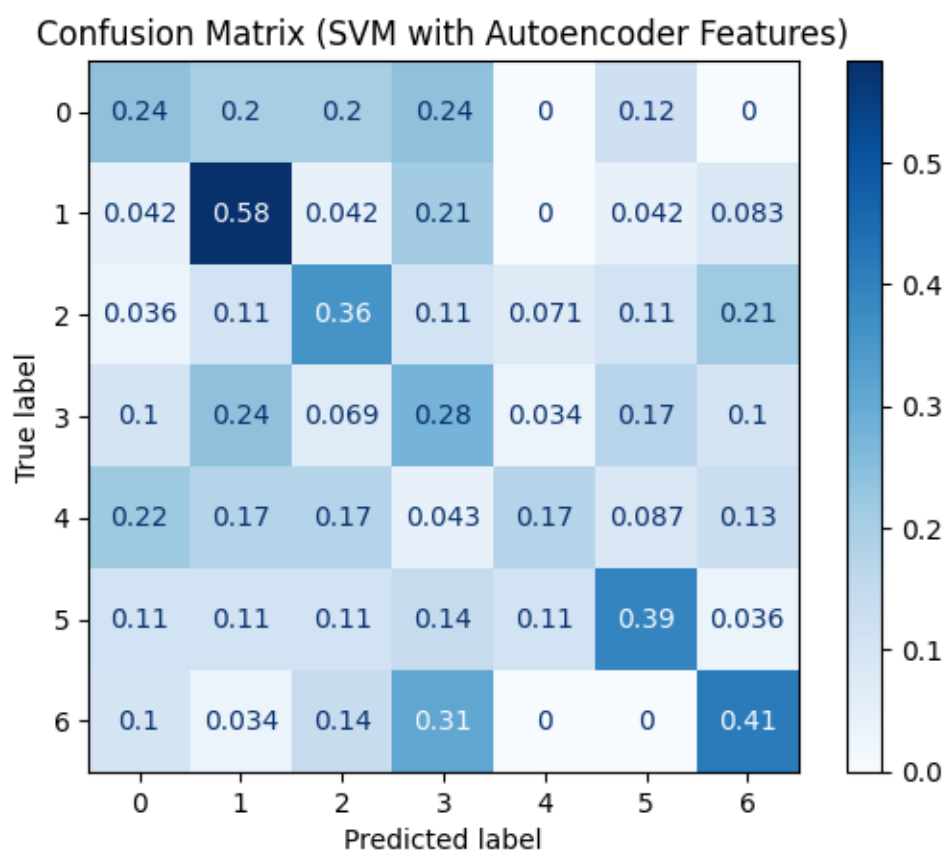
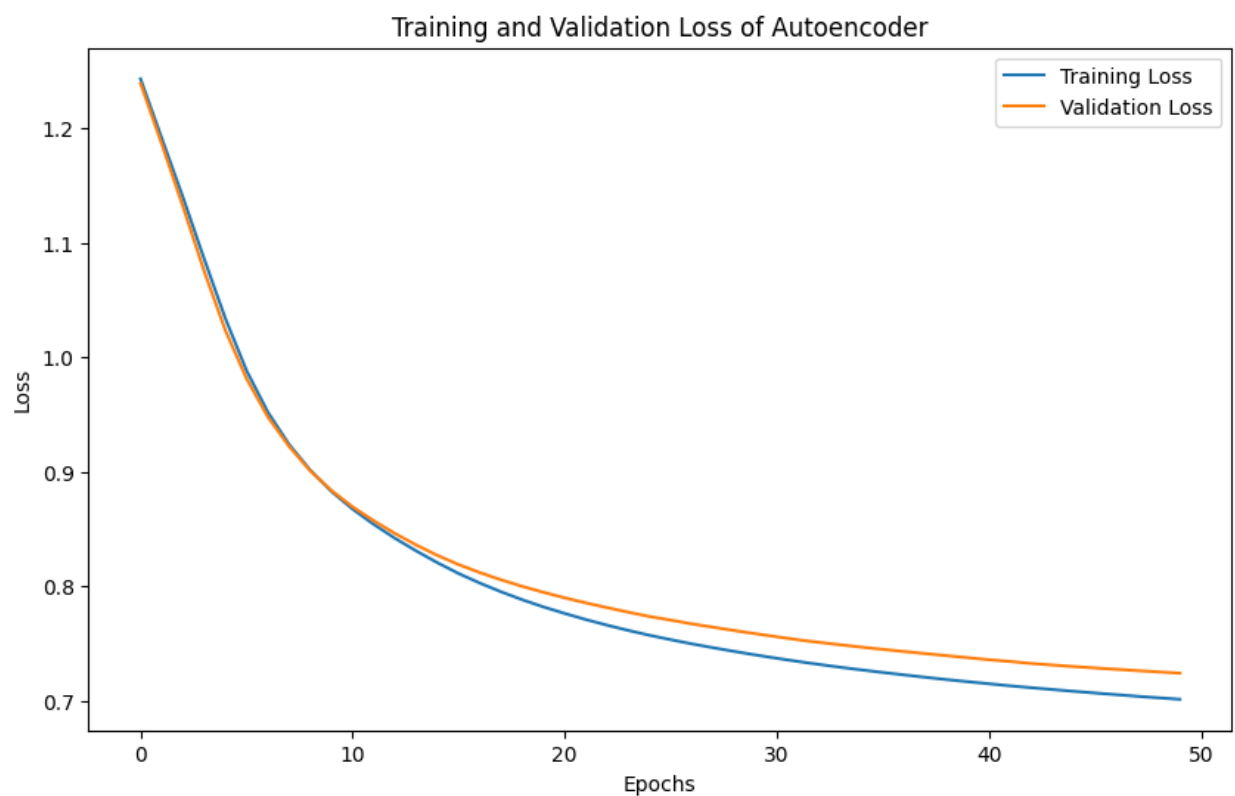


نمودار فوق نشان دهنده تاثیر تعداد PCA در نتایج کلسیفایر می‌باشد. همانطور که مشخص است در حوالی ۱۵ principal component به بهترین نتیجه رسیده‌ایم و بعد از آن نتایج افت داشته است.

بخش ۵)

به منظور ویژگی جدید spectral\_centroid\_mean هر موسیقی را از spectral\_centroid\_mean معمول هر instrument کم می‌کنیم تا spectral\_centroid\_mean صدای خواننده پیدا شود. با توجه با اینکه خواننده در کدام پرده و دستگاه می‌خواند، می‌تواند در نتیجه تاثیر بگذارد.

شبکه auto-encoder که طراحی کرده‌ایم داده را با ابعاد ۷۰ دریافت می‌کند و آن را به ابعاد ۱۵ می‌برد (این ابعاد بهترین نتایج را در PCA حاصل می‌کرد) و سپس به ابعاد اولیه بر می‌گرداند. نتیجه حاصل شده به شرح زیر می‌باشد:



### Classification Report:

precision recall f1-score support

0	0.27	0.24	0.26	25
1	0.38	0.58	0.46	24
2	0.34	0.36	0.35	28
3	0.22	0.28	0.25	29
4	0.40	0.17	0.24	23
5	0.44	0.39	0.42	28
6	0.44	0.41	0.43	29

accuracy 0.35 186

macro avg 0.36 0.35 0.34 186

weighted avg 0.36 0.35 0.34 186

نتایج به نسبت افت کرده است که قابل پیشبینی بود زیرا تعداد دیتا کم است و عمق شبکه هم پایین است. برای استخراج ویژگی از bottle-neck استفاده کردیم که ۱۵ ویژگی را برمیگرداند.

(و) کد مربوطه زده شده است اما ارور زیر دریافت می‌شود.

```
ValueError: X has 69 features, but SVC is expecting 70 features as input.
```

مطمئن نیستیم که ایراد از کار من هست یا در csv سکوت سرد زمان داده‌ای کم است.

(ز) این مدل می‌تواند در زمینه کار هیئت ممیزه وزارت ارشاد استفاده شود تا اگر خدایی ناکرده موسیقی از دستگاه‌های متعارف جامعه خارج شد، آن را شناسایی کنند. علاوه بر آن این سیستم می‌تواند در پلتفرم‌های پخش موسیقی مثل spotify به منظور دسته بندی موسیقی‌ها در ژانرهای مختلف استفاده شود.

به عنوان مدل جایگزین یک شبکه عصبی طراحی شده که جزییات آن به شرح زیر است:

Model: "sequential"

Layer (type)	Output Shape	Param #
=====		
dense_6 (Dense)	(None, 64)	4544
dropout (Dropout)	(None, 64)	0

dense_7 (Dense)	(None, 32)	2080
dropout_1 (Dropout)	(None, 32)	0
dense_8 (Dense)	(None, 7)	231

=====

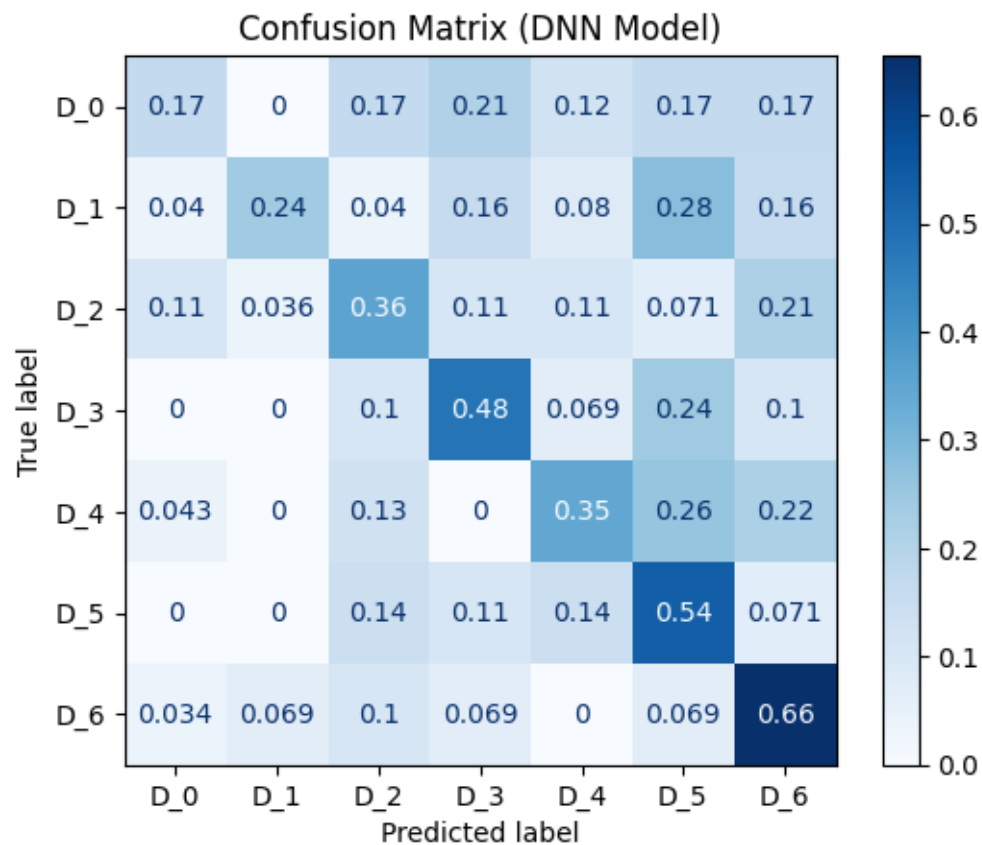
Total params: 6855 (26.78 KB)

Trainable params: 6855 (26.78 KB)

Non-trainable params: 0 (0.00 Byte)

---

که نتیجه آ» به شرح زیر است:



Classification Report:

	precision	recall	f1-score	support
0	0.40	0.17	0.24	24
1	0.67	0.24	0.35	25
2	0.36	0.36	0.36	28
3	0.45	0.48	0.47	29
4	0.36	0.35	0.36	23
5	0.35	0.54	0.42	28
6	0.44	0.66	0.53	29
accuracy			0.41	186
macro avg	0.43	0.40	0.39	186
weighted avg	0.43	0.41	0.39	186





که نتایج حاصل شده قابل مقایسه با مدل SVM است اما همانطور که از نمودار loss مشخص است این مدل همچنان جای آموزش دارد و می‌تواند نتایج بهتر از مدل SVM کسب کند.

مدل نتیجه بهتر کسب کرد اما فرصت قرار دادن آن در انتهای امتحان وجود ندارد.

باز هم مرحله (۳)  $Q=0$  :

$$Q(A_1, A_2) = 0, Q(A_2, A_3) = 0, Q(A_3, B_3) = 0$$

$$Q(B_3, B_4) = -10$$

$$Q(C_2, C_1) = Q(C_1, B_1) = Q(B_1, A_1) = Q(A_1, A_2) = 0$$

$$Q(A_2, A_3) = 0, Q(A_3, A_4) = 10$$

$$Q(C_4, C_3) = 0, Q(C_3, B_3) = 0 + 0.9 \times \max(0, -10) = 0$$

$$Q(B_3, A_3) = 0 + 0.9 \times \max(0, 10) = 9, Q(A_3, A_4) = 10 + 0.9 \times 9 = 18.1$$

	1	2	3	4
A	↗	↔	→	Ⓚ
B	↓	////	↑	↖
C	↘	↔	↖	↖

این سیاست می تواند بهینه نباشد

زیرا از روش greedy search

استفاده می کند و امکان وجود مسیر بهتر وجود دارد