تمرین سری سوم

سوال اول :

الف)

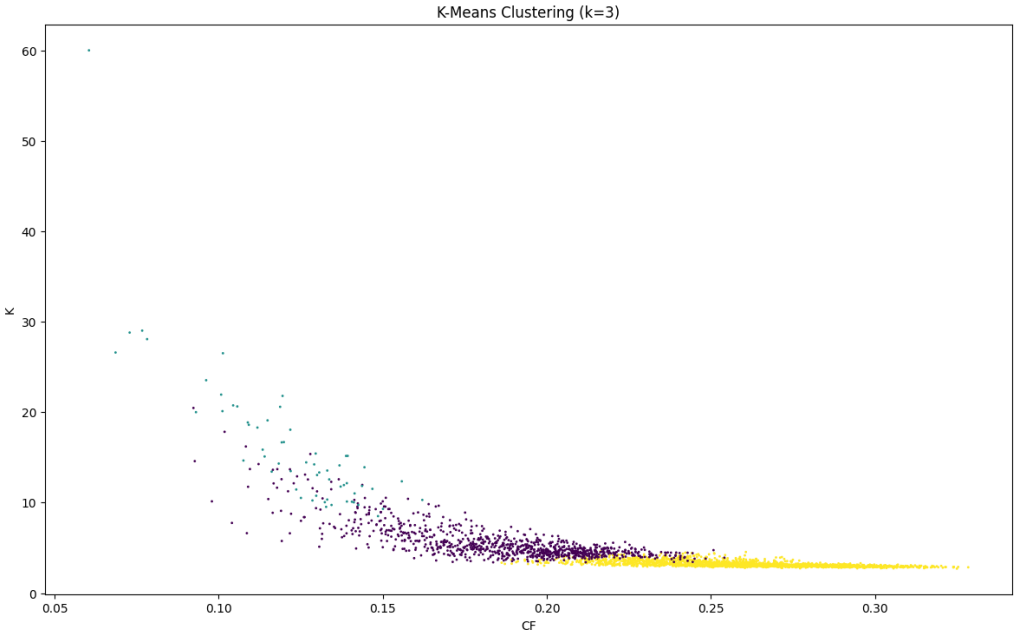
برای استفاده از الگوریتم k-means و نمایش جدا شدن داده ها از کتابخانه sikit و کد زیر استفاده شده است:

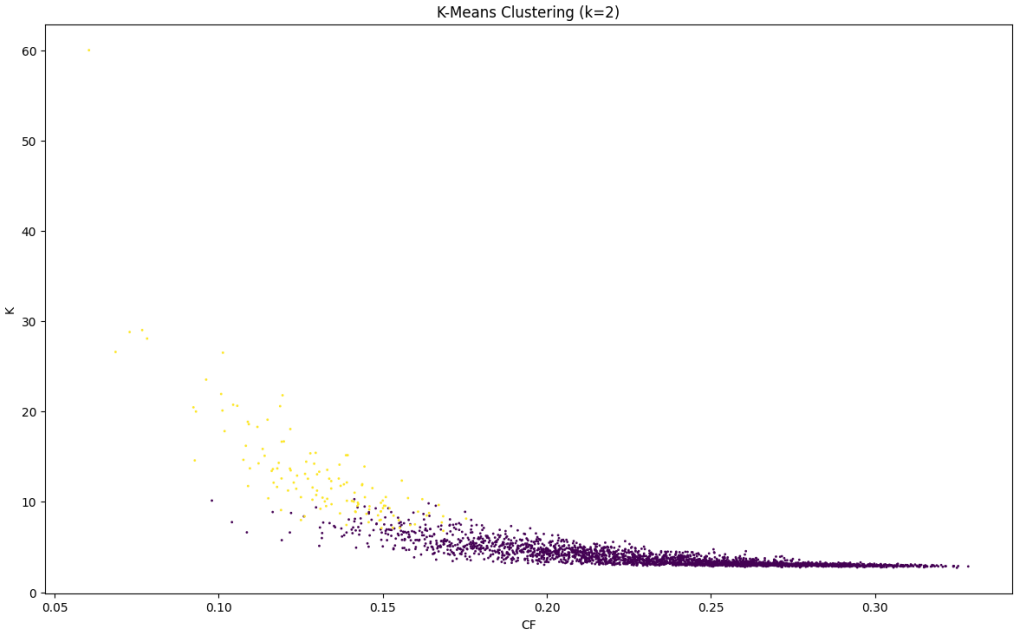
```
n_clusters = 2

# Apply k-means clustering
kmeans = KMeans(n_clusters, random_state=42)
df['cluster'] = kmeans.fit_predict(scaled_data)
```

با تغییر مقدار n_cluster میتوان تعداد دسته را کنترل نمود.

نتایج برای حالت دو دسته و سه دسته به صورت زیر است:


K-Means Clustering (k=3)
K
CF
0
10
20
30
40
50
60
0.05
0.10
0.15
0.20
0.25
0.30


K-Means Clustering (k=2)
K
CF
0
10
20
30
40
50
60
0.05
0.10
0.15
0.20
0.25
0.30

ب)

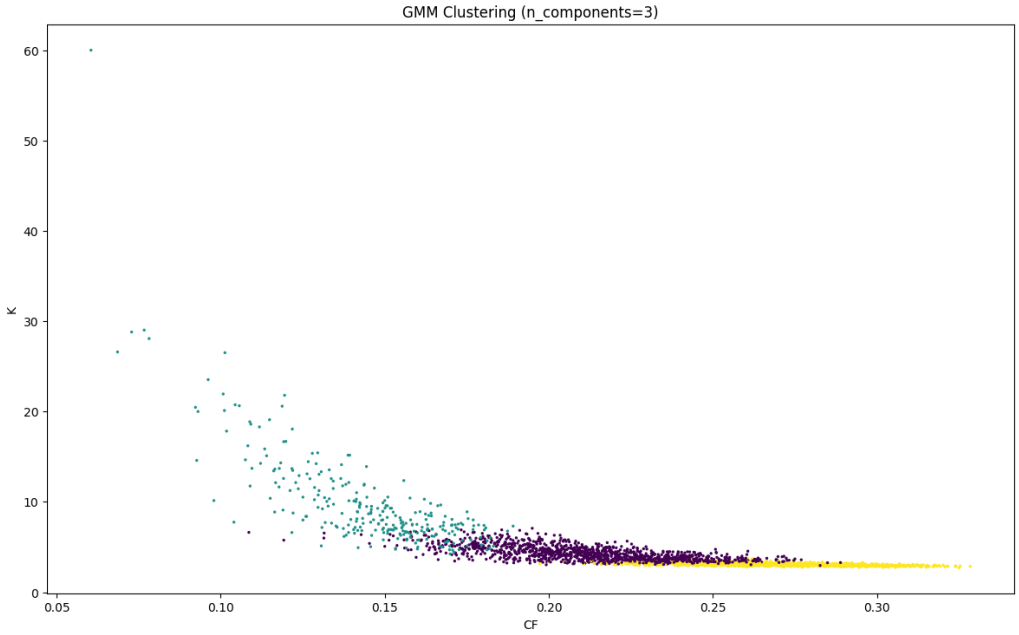
در این قسمت برای حالتی که از الگوریتم GMM استفاده بشود، کد زیر توسعه یافته است :

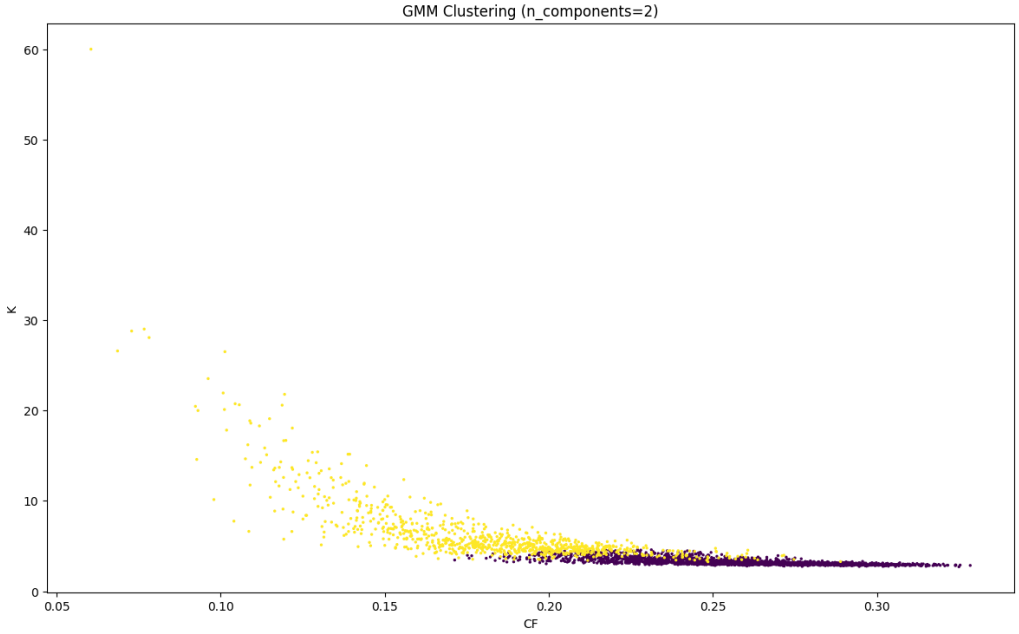
```
# Choose the number of components (clusters)
n_clusters = 2

# Apply GMM clustering
gmm = GaussianMixture(n_clusters, random_state=42)
df['cluster'] = gmm.fit_predict(scaled_data)
```

که با تغییر مقدار n_cluster میتوان تعداد دسته ها را کنترل نمود.

نتایج حاصل به صورت زیر میباشد:





ج)

برای یافتن مولفه های اصلی و وابستگی هر کدام، ازکتابخانه sikit و کد زیر استفاده شده است :

```
# Choose the number of principal components
n_components = min(features.shape[0], features.shape[1])  # Use min for safety
pca = PCA(n_components=n_components)

# Fit the PCA model and transform the data
principal_components = pca.fit_transform(scaled_features)

# Get the explained variance ratio for each principal component
explained_variance_ratio = pca.explained_variance_ratio_
```

Principal Component Indices (PCI):

[[2 3 1 0]

[1 0 3 2]

[0 3 2 1]

[2 3 1 0]]

Correlation with PC1:

Parameter	Correlation with PC1
CF	-0.459560
K	0.487109
peak	0.533754
RMS	0.516368

سوال دوم

الف)

با توجه به انجام تمرینات در فضای python ، از مجموعه داده های مربوط به سرطان پستان موجود در کتابخانه sikit استفاده شده است. برای فراخوانی مجموعه داده مورد نظر از کد زیر استفاده شده است:

```
from sklearn.datasets import load_breast_cancer
b_cancer = load_breast_cancer()
```

این مجموعه داده، شامل 569 داده و 30 ویژگی زیر است.

1. mean radius
2. mean texture
3. mean perimeter
4. mean area
5. mean smoothness
6. mean compactness
7. mean concavity
8. mean concave points
9. mean symmetry
10. mean fractal dimension
11. radius error
12. texture error
13. perimeter error
14. area error
15. smoothness error
16. compactness error
17. concavity error
18. concave points error
19. symmetry error
20. fractal dimension error
21. worst radius
22. worst texture
23. worst perimeter
24. worst area
25. worst smoothness
26. worst compactness
27. worst concavity
28. worst concave points
29. worst symmetry
30. worst fractal dimension

برای توسعه مدل شبکه عصبی با یک لایه پنهان و چهار پرسپترون ، از کتابخانه sikit و کد زیر استفاده شده است :

```
b_cancer = load_breast_cancer()

X_train = b_cancer.data[:469, :4]
X_test = b_cancer.data[470:568, :4]
y_train = b_cancer.target[:469]
y_test = b_cancer.target[470:568]

# Feature Scaling
scaler = StandardScaler()
X_train_scaled = scaler.fit_transform(X_train)
X_test_scaled = scaler.transform(X_test)

# Create an MLP classifier with 1 hidden layer and 4 perceptrons
mlp_classifier = MLPClassifier(hidden_layer_sizes=(4,), max_iter=1000,
random_state=42, learning_rate_init=0.01, solver='sgd', momentum=0.9,
verbose=True)

# Train the classifier on the scaled training data
mlp_classifier.fit(X_train_scaled, y_train)

# Make predictions on the scaled test data
y_pred = mlp_classifier.predict(X_test_scaled)
```

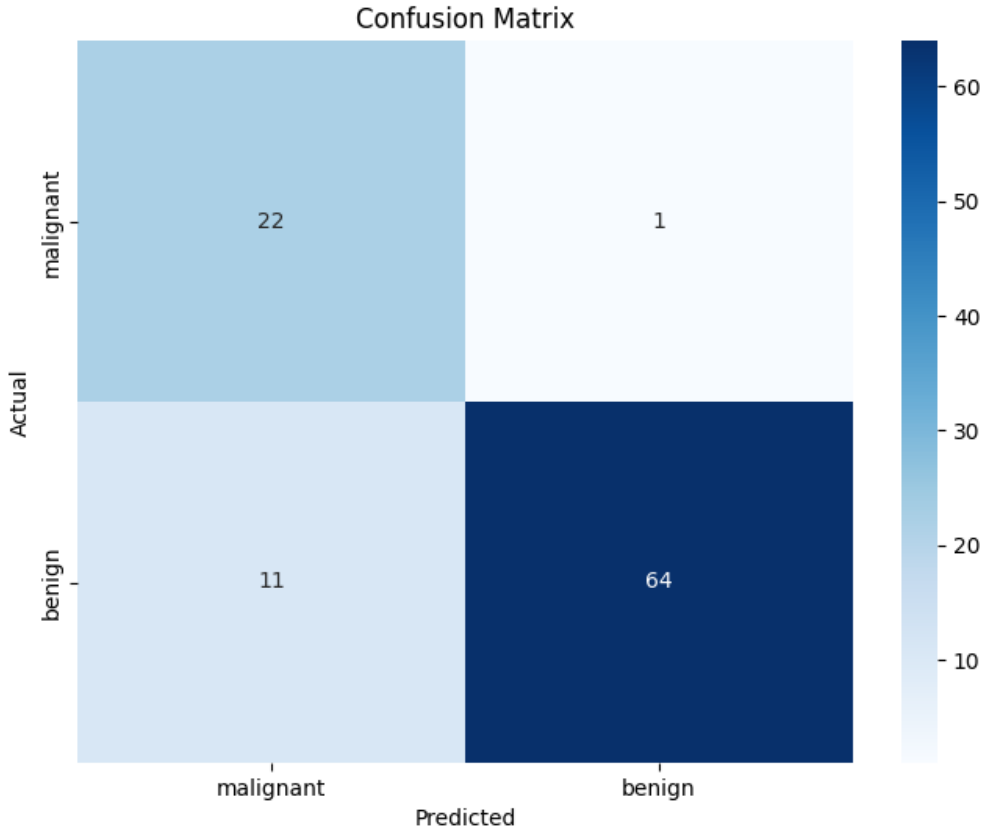
برای حالتی که دارای دو لایه پنهان و پنج پرسپترون است، کد زیر مورد استفاده قرار میگیرد :

```
mlp_classifier = MLPClassifier(hidden_layer_sizes=(5, 5), max_iter=1000,
random_state=42, learning_rate_init=0.01, solver='sgd', momentum=0.9)
```

با توجه به تفاوت بانک داده در متلب و پایتون ، داده های 0 تا 469 برای آموزش مدل و داده های 470 تا 568 برای تست استفاده شده است.

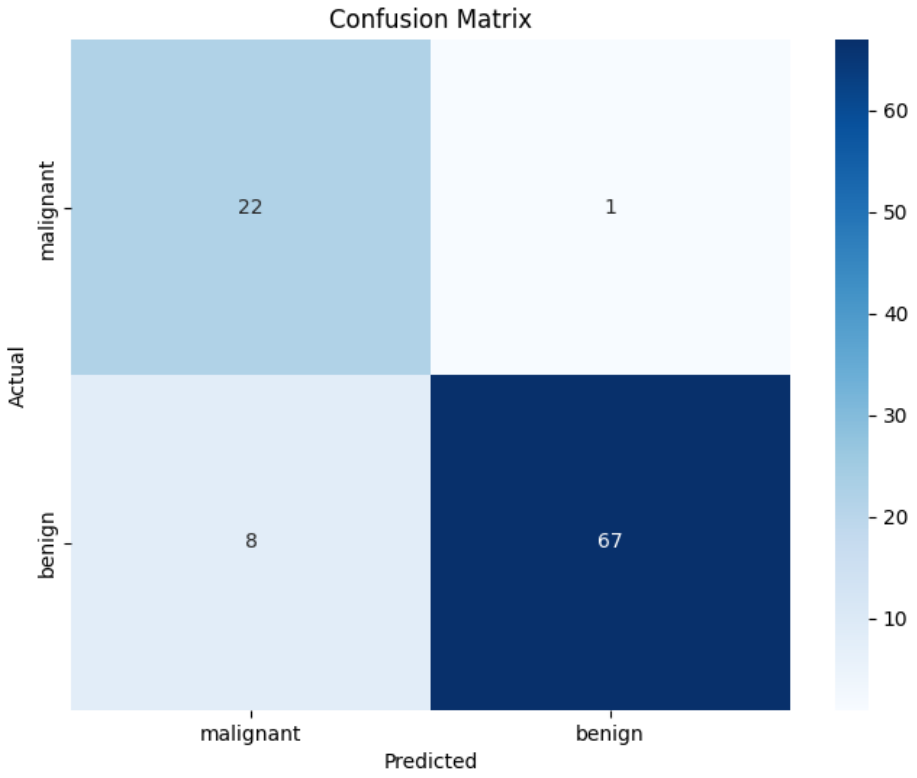
نتایج زیر استخراج شده است:

برای حالت اول :



accuracy: 0.88

حالت دوم:



Accuracy: 0.91

همان طور که مشاهده میشود، با افزایش لایه ها شاهد افزایش دقت مدل هستیم. مدل دوم دارای اطمینان بیشتری می باشد.

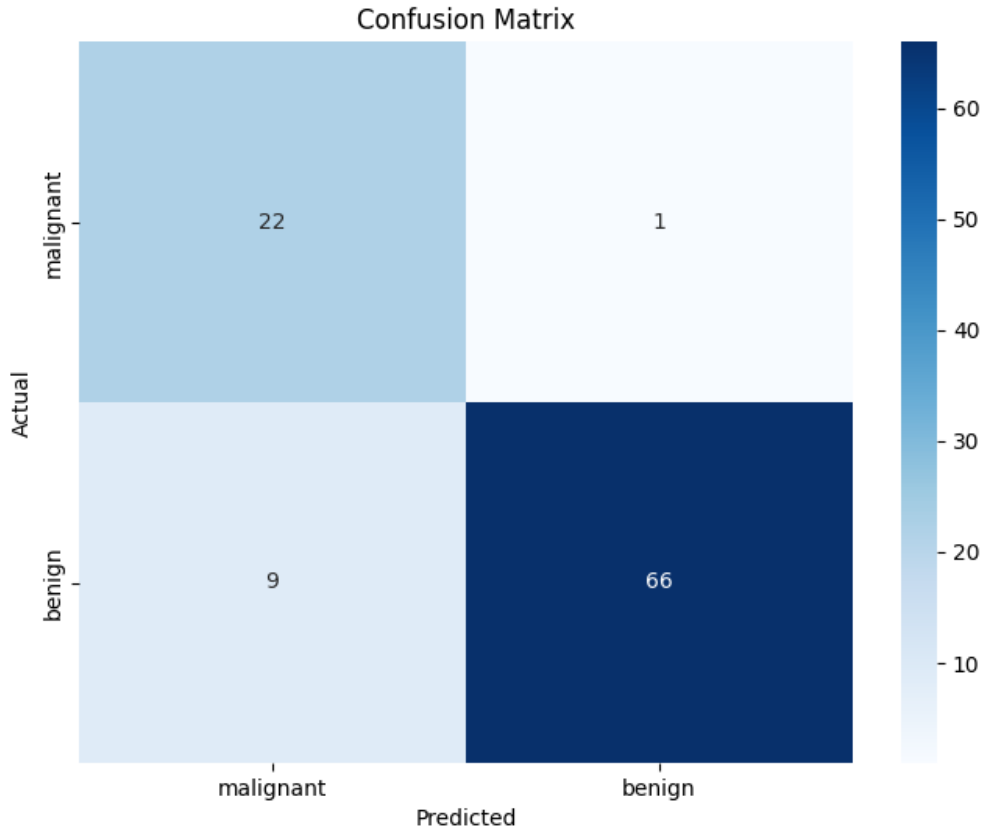
ب)

برای تغییر ورودی و استفاده از نه ویژگی از کد زیر استفاده شده است :

```
X_train = b_cancer.data[:469, :9]
X_test = b_cancer.data[470:568, :9]
y_train = b_cancer.target[:469]
y_test = b_cancer.target[470:568]
```

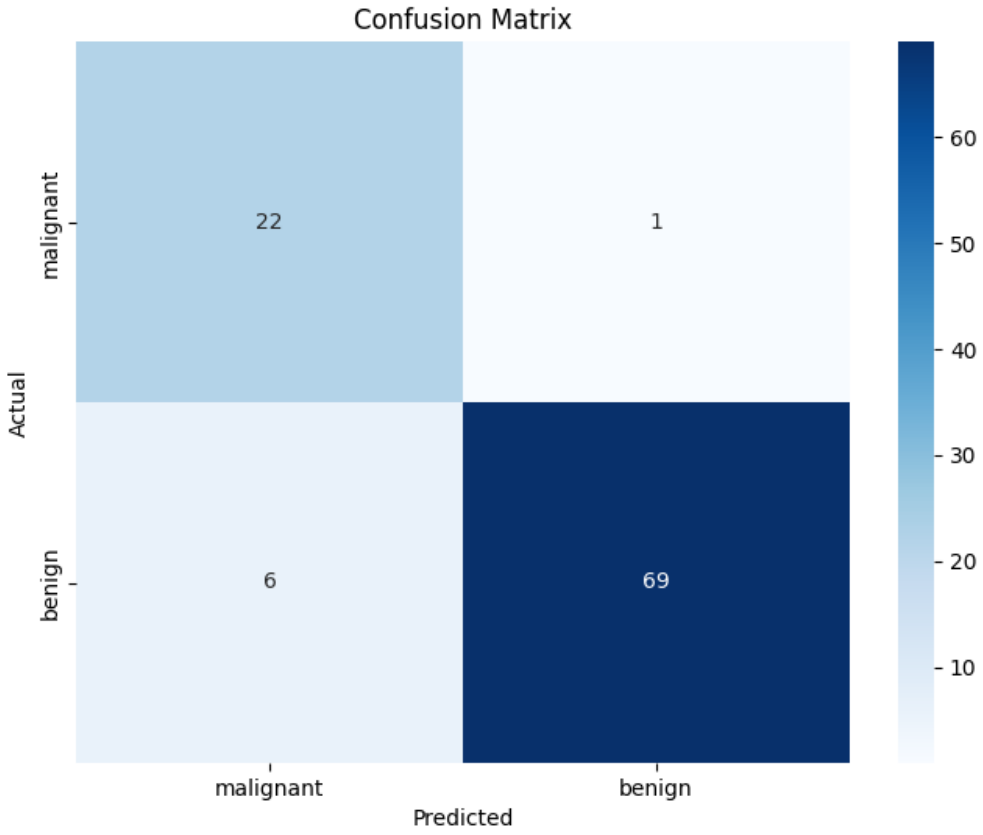
نتایج عملکرد مدل در این حالت به شکل زیر است :

حالت اول :



accuracy : 0.90

وبرای حالت دوم :



accuracy : 0.93

که بازهم شاهد افزایش دقت در حالتی که از ۲ لایه ی پنهان استفاده ، هستیم.

ج) با افزایش تعداد ویژگی های مورد استفاده، شاهد افزایش دقت مدل هستیم.

د) با توجه به تفاوت مجموعه داده ها، این قسمت توسط داده های مورد ارائه در سوال قابل انجام نیست. اما برای ارزیابی داده های جدید میتوان از کد زیر استفاده نمود :

```
predicted_labels = mlp_classifier.predict(new_instances_normalized)
```

که میتوان در آرگومان تابع بالا ، مجموعه شامل داده های جدید را وارد نمود.

و) با توجه به اعداد بدست آمده برای دقت ۲ الگوریتم SVM و NN و مقایسه این اعداد متوجه میشویم که مدل SVM دارای عملکرد بهتری میباشد.