

تمرین دوم

سوال اول :

(الف)

نمودار های زیر نشان دهنده خط دسته بندی برای حالت مختلف توسعه مدل است.

برای این منظور از ۲ سری کد استفاده شده است که در سری اول الگوریتم به صورت دستی توسعه یافته و در سری دوم از کتابخانه **sikit** استفاده گردیده شده است.

برای کد توسعه یافته شده به صورت دستی از تابع زیر برای ساخت مدل پرسپترون **bacth** استفاده شده است:

```
def perceptron(X, y):
    # Add bias term to the features
    learning_rate = 0.0001
    n_iterations = 100
    X['bias'] = 1
    X = X.values
    y = y.values
    n = 0

    # Initialize weights
    weights = np.zeros(X.shape[1])

    for j in range(n_iterations):
        for i in range(X.shape[0]):
            # Compute activation
            xop = X[i]
            yop = y[i]
            activation = np.dot(xop, weights)
            activation = yop * activation

            # Apply step function (threshold at 0)
            if activation <= 0:
                for k in range(len(weights)):
                    weights[k] = weights[k] + learning_rate * y[i] * X[i, k]
                n = 1 + n

    # Update weights after going through all data points
    j += 1
```

```
print(weights)
print(n)
return weights[:-1], weights[-1] # Return weights and bias
```

در کد بالا تابع با دریافت مجموعه داده ها و برجسب داده ها ، و با در نظر گرفتن 0.0001 به عنوان نرخ یادگیری و انجام روند یادگیری به تعداد 100 بار ، مدل پرسپترون مورد نظر را ایجاد میکند.

برای حالت online و دستی ، از کد تغییر یافته به شکل زیر استفاده میگردد:

```
def perceptron(X, y):
    # Add bias term to the features
    learning_rate = 0.0001
    X['bias'] = 1
    X = X.values
    y = y.values
    n = 0

    # Initialize weights
    weights = np.zeros(X.shape[1])

    for i in range(X.shape[0]):
        # Compute activation
        xop = X[i]
        yop = y[i]
        activation = np.dot(xop, weights)
        activation = yop * activation

        # Apply step function (threshold at 0)
        if activation <= 0:
            for k in range(len(weights)):
                weights[k] = weights[k] + learning_rate * y[i] * X[i, k]
            n = 1 + n

    print(weights)
    print(n)
    return weights[:-1], weights[-1] # Return weights and bias
```

در حالت تابع با ورودی مجموعه داده و برجسب های متناظر، یک مدل پرسپترون online با نرخ یادگیری 0.0001 ایجاد میکند که به تعداد داده ها روند یادگیری را انجام میدهد.

برای ایجاد نتایج با استفاده از توابع کتابخانه ای از کد زیر و کتابخانه `sikit` استفاده شده است.

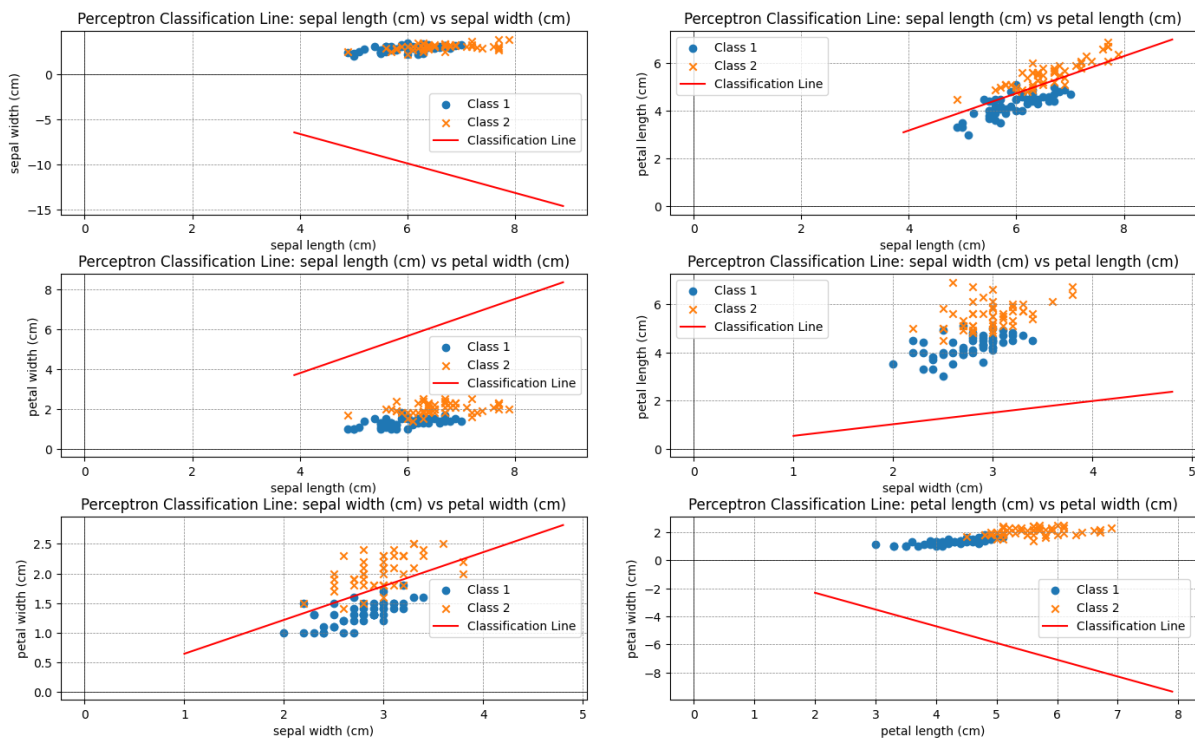
برای حالت `batch`:

```
perceptron_model_batch = Perceptron(max_iter=100000, tol=1e-3, shuffle=False)
perceptron_model_batch.fit(X_train, y_train)
```

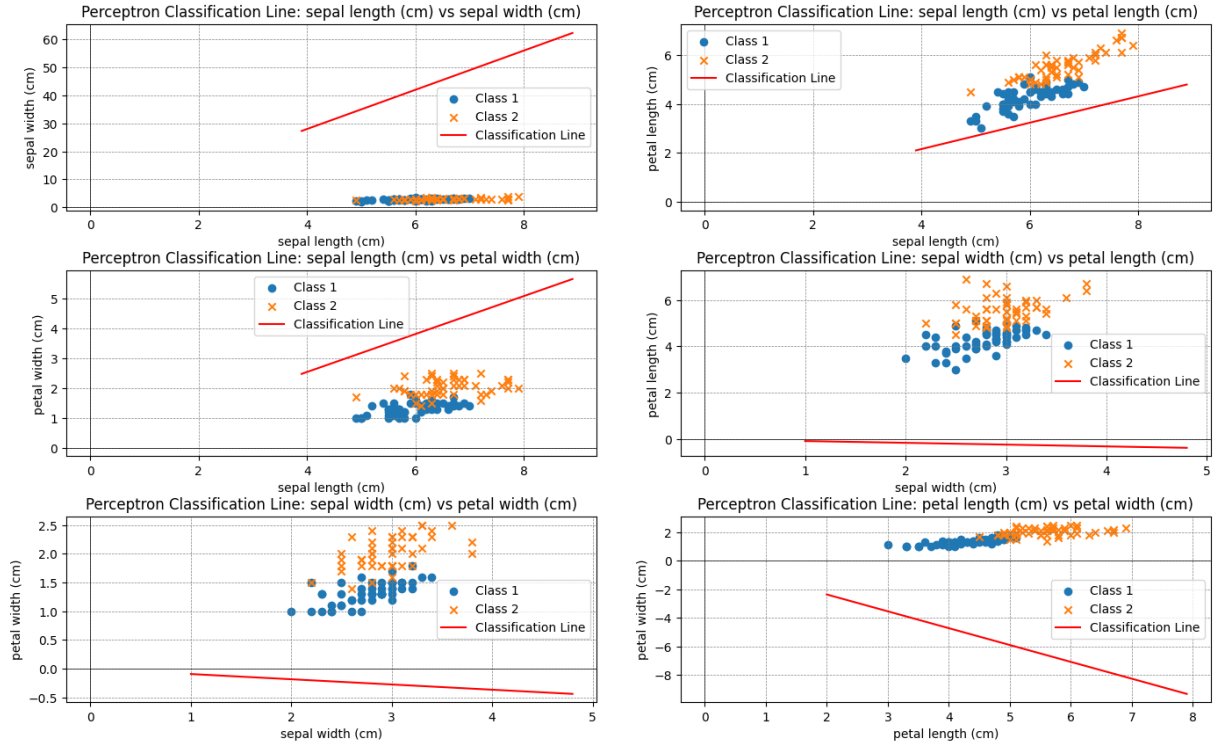
برای حالت `online`:

```
perceptron_model = Perceptron(max_iter=100000, tol=1e-3)
perceptron_model.fit(X_train, y_train)
```

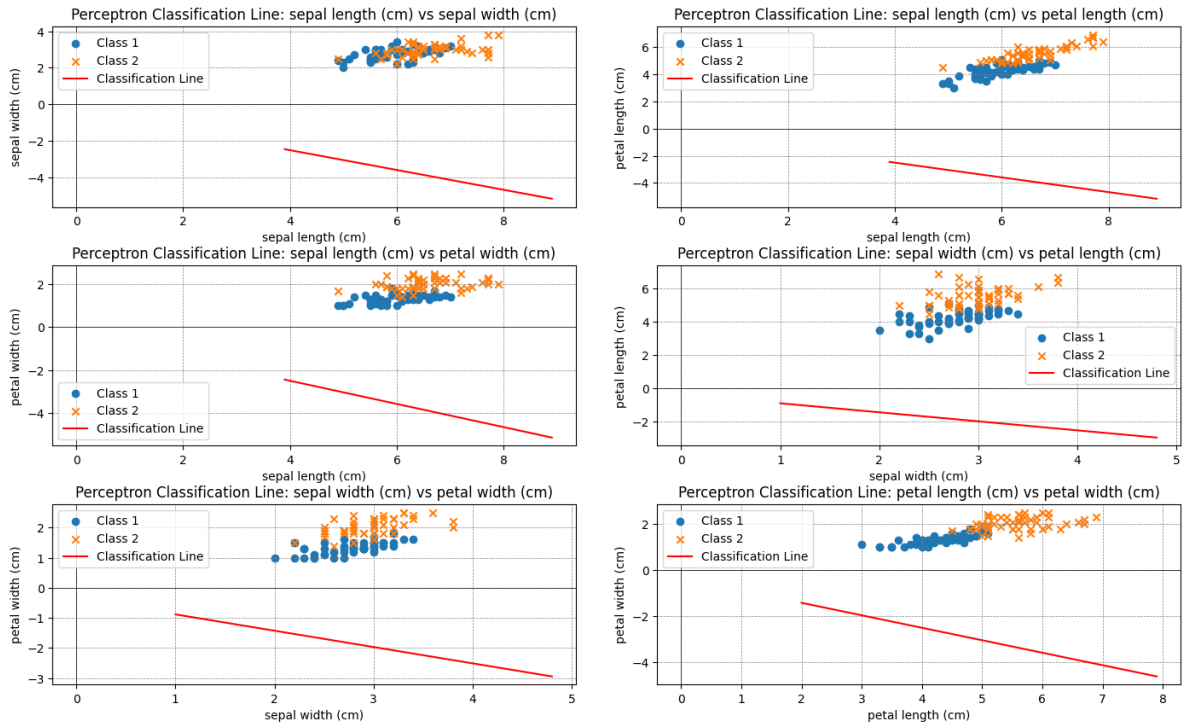
batch perceptron manual



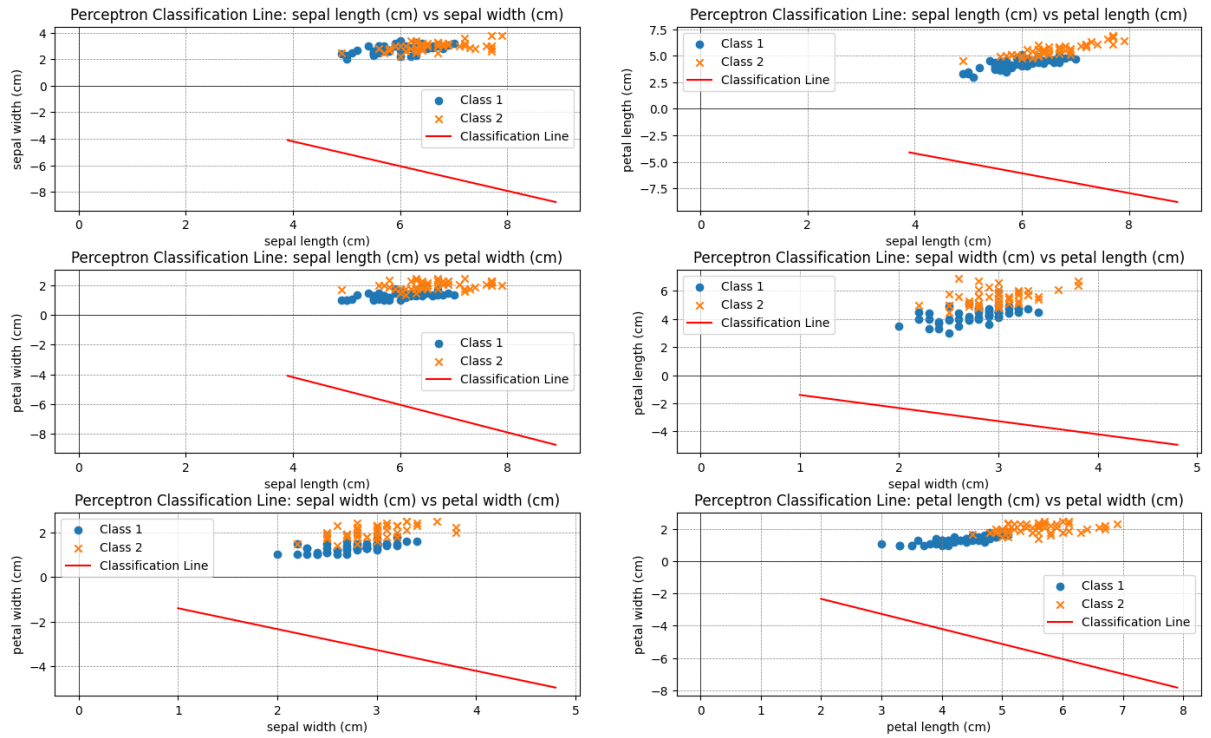
online perceptron manual



batch perceptron with sikit



online perceptron with sikit



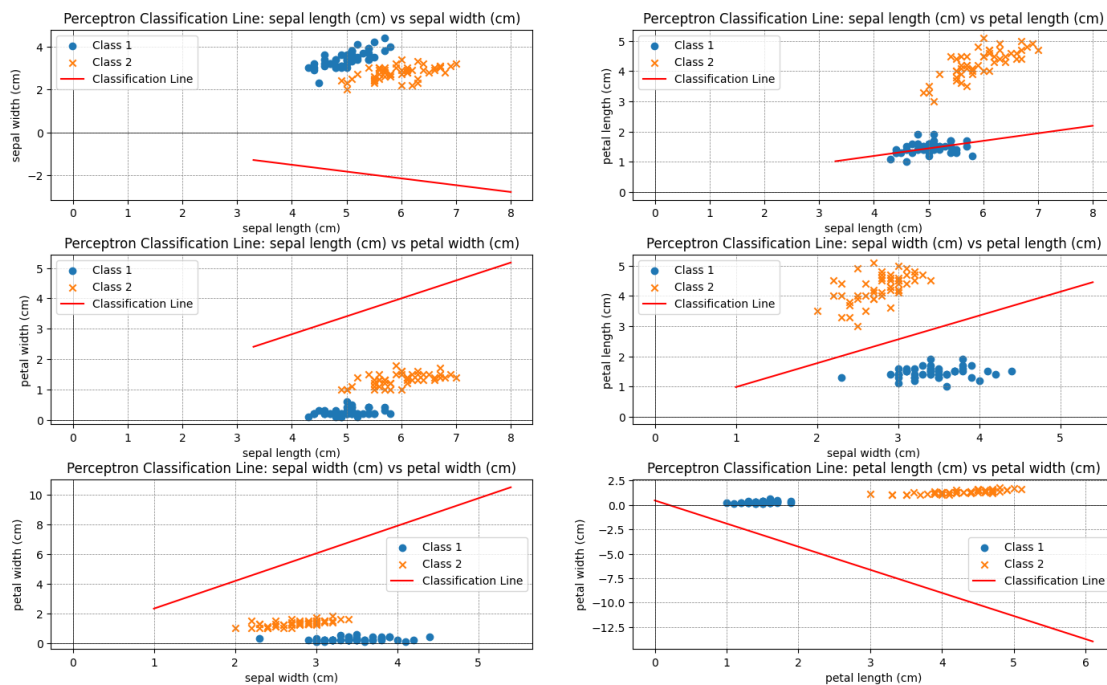
همانطور که از نمودارها قابل ملاحظه است این الگوریتم دقت کافی برای رسم خط دسته بندی را دارا نیست. این مسئله فارغ از توسعه کد به صورت دستی و یا استفاده از کتابخانه ها است.

(ب)

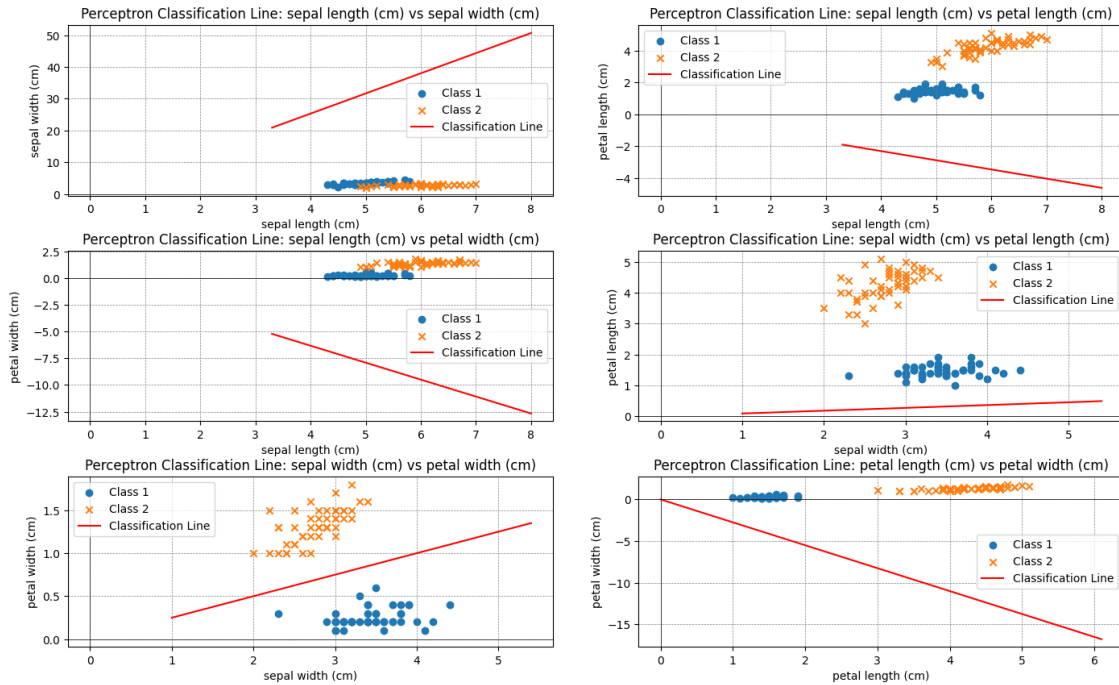
در این قسمت، ورودی داده ها تغییر پیدا کرده است تا مدل برای بررسی دو کلاس اول و دوم توسعه پیدا کند.

از کد های مرحله قبل برای توسعه مدل ها با ایجاد تغییر در ورودی استفاده شده است.

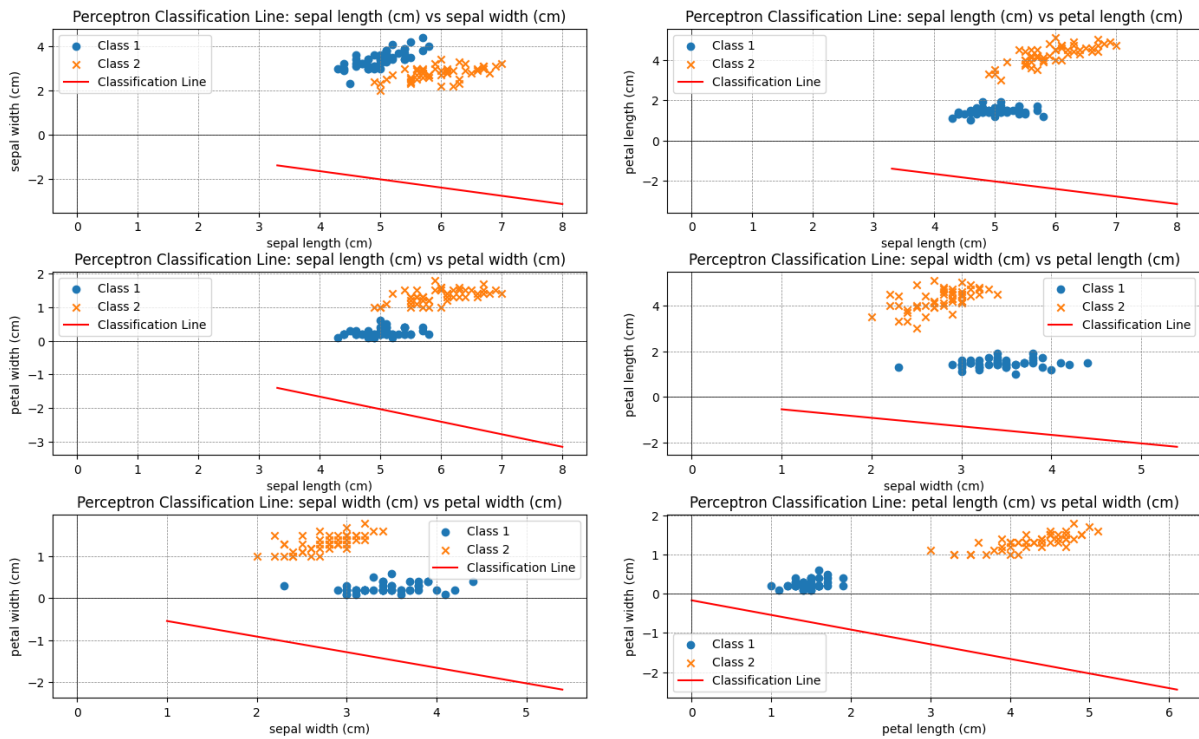
batch perceptron manual



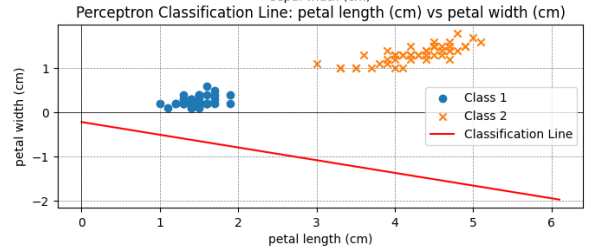
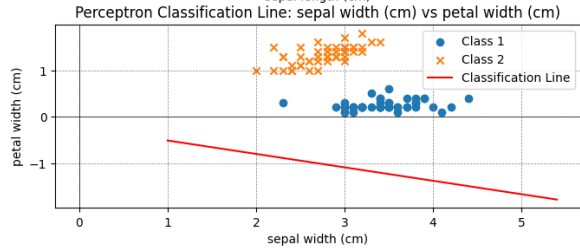
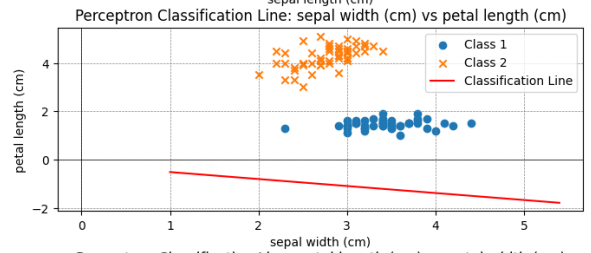
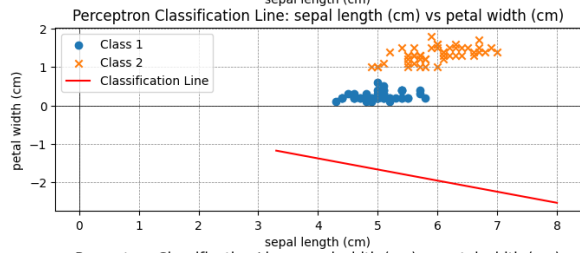
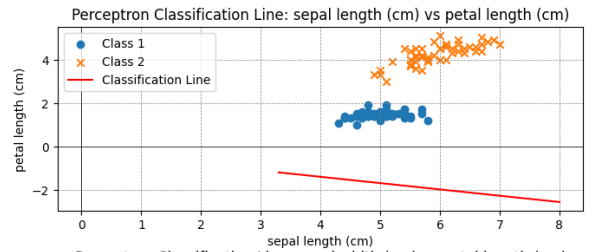
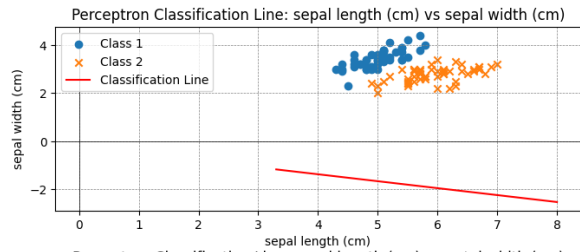
online perceptron manual



batch perceptron with sikit



online perceptron with sikit



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 ایجاد مدل SVM با کرنل های مختلف از کتابخانه `sikit` و کد زیر استفاده شده است :

```
X = b_cancer.data[:, :9]
y = b_cancer.target

# Split the data into training and testing sets
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2,
random_state=42)

# Initialize SVM classifier with a linear kernel
svm_classifier = SVC(kernel='poly', degree=2)

# Train the SVM classifier
svm_classifier.fit(X_train, y_train)

# Make predictions on the test set
predictions = svm_classifier.predict(X_test)
```

که در این کد با تغییر مقادیر ورودی میتوان مجموعه داده را تنظیم کرد. برای تغییر کرنل مورد استفاده در مدل SVM میتوان آرگومان تابع را تغییر داد.

در حالت اول کرنل خطی :

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

در حالت مرتبه 2 :

```
svm_classifier = SVC(kernel='poly', degree=2)
```

در حالت مرتبه 3 :

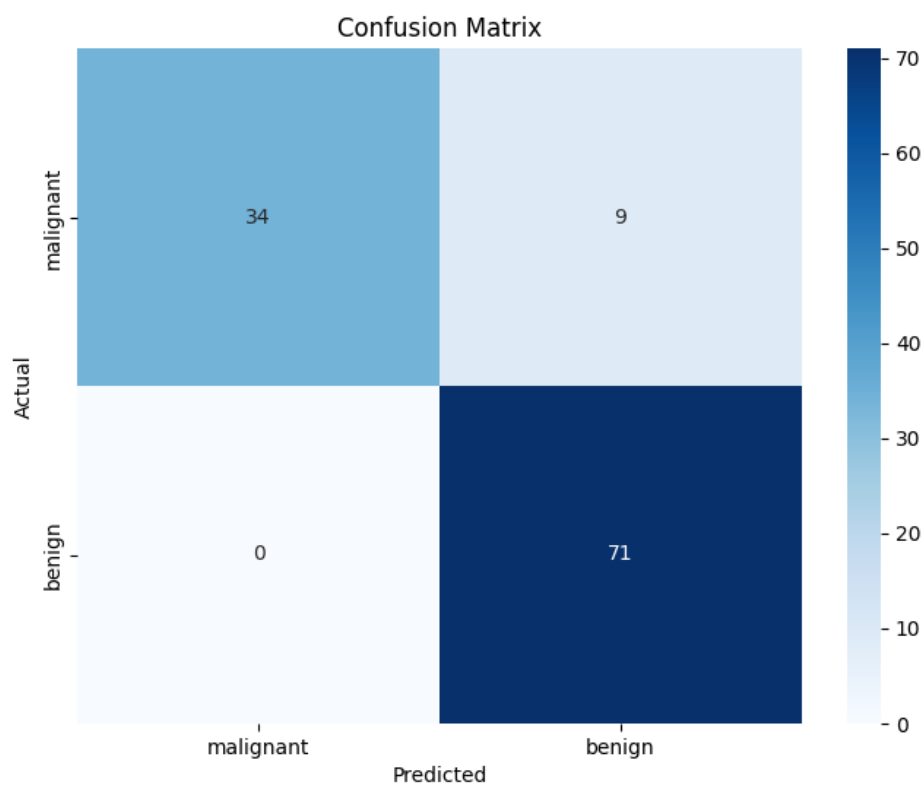
```
svm_classifier = SVC(kernel='poly', degree=3)
```

و در حالت گوسی :

```
svm_classifier = SVC(kernel='rbf')
```

به ازای در نظر گرفتن چهار ویژگی اول ، ماتریس در هم ریختگی ، دقت و داده های اشتباه دسته بندی شده زیر حاصل میشود:

خطی



Accuracy: 0.9210526315789473

False Predictions:

[۵۸۴.۱ ۹۰.۴۳ ۱۵.۷۹ ۱۳.۸]

[۷۱۶.۶ ۹۸ ۲۵.۷۴ ۱۵.۰۸]

[۵۵۹.۲ ۸۸.۴ ۲۰.۸۲ ۱۳.۴۸]

[۵۸۸.۹ ۹۰.۶۳ ۲۲.۲۹ ۱۳.۷۷]

[۶۰۲.۴ ۹۱.۴۳ ۱۷.۰۵ ۱۳.۹۶]

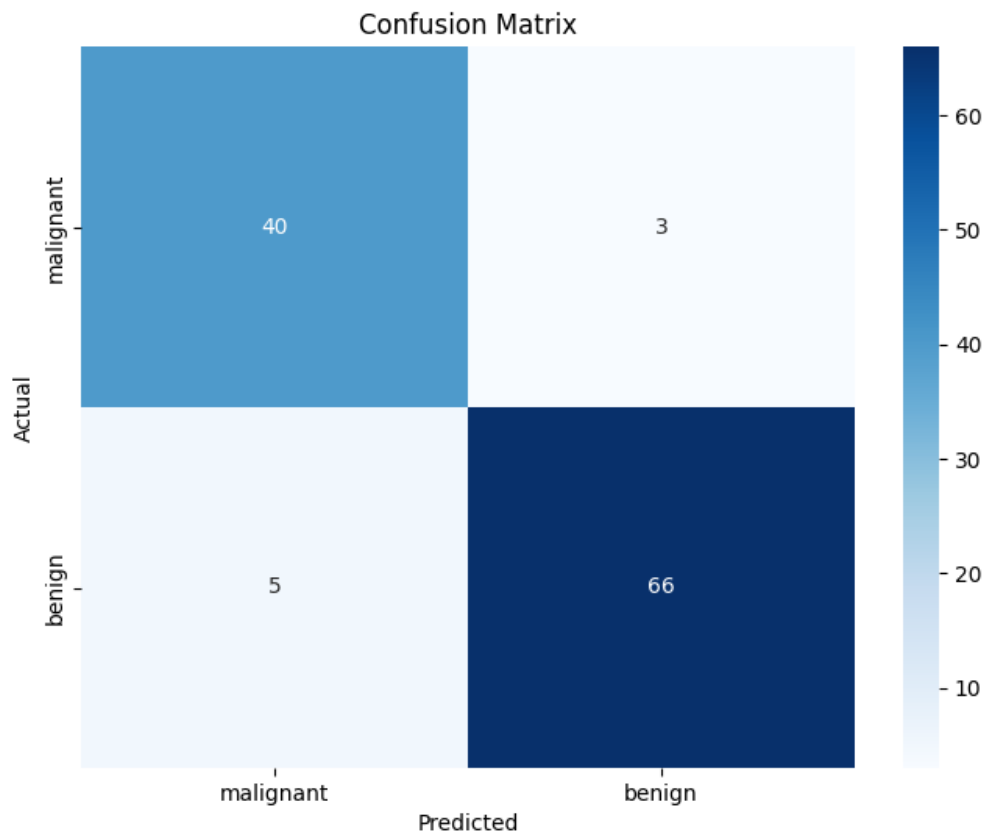
[۶۴۸.۲ ۹۴.۲۵ ۲۱.۴۶ ۱۴.۴۸]

[۷۱۳.۳ ۱۰۳.۲ ۱۷.۲۷ ۱۵.۳۲]

[۴۷۵.۹ ۸۳.۹۷ ۲۴.۰۴ ۱۲.۴۶]

[[۵۹۷.۸ ۹۱.۵۶ ۲۳.۷۵ ۱۳.۸۱]

مرتبہ دوم



Accuracy: 0.9298245614035088

False Predictions:

[۵۸۴.۱ ۹۰.۴۳ ۱۵.۷۹ ۱۳.۸]

[۶۶۲.۷ ۹۴.۵۷ ۲۴.۰۲ ۱۴.۶۲]

[۵۲۹.۴ ۸۷.۰۲ ۲۲.۵۴ ۱۳.۱۱]

[۵۵۹.۲ ۸۸.۴ ۲۰.۸۲ ۱۳.۴۸]

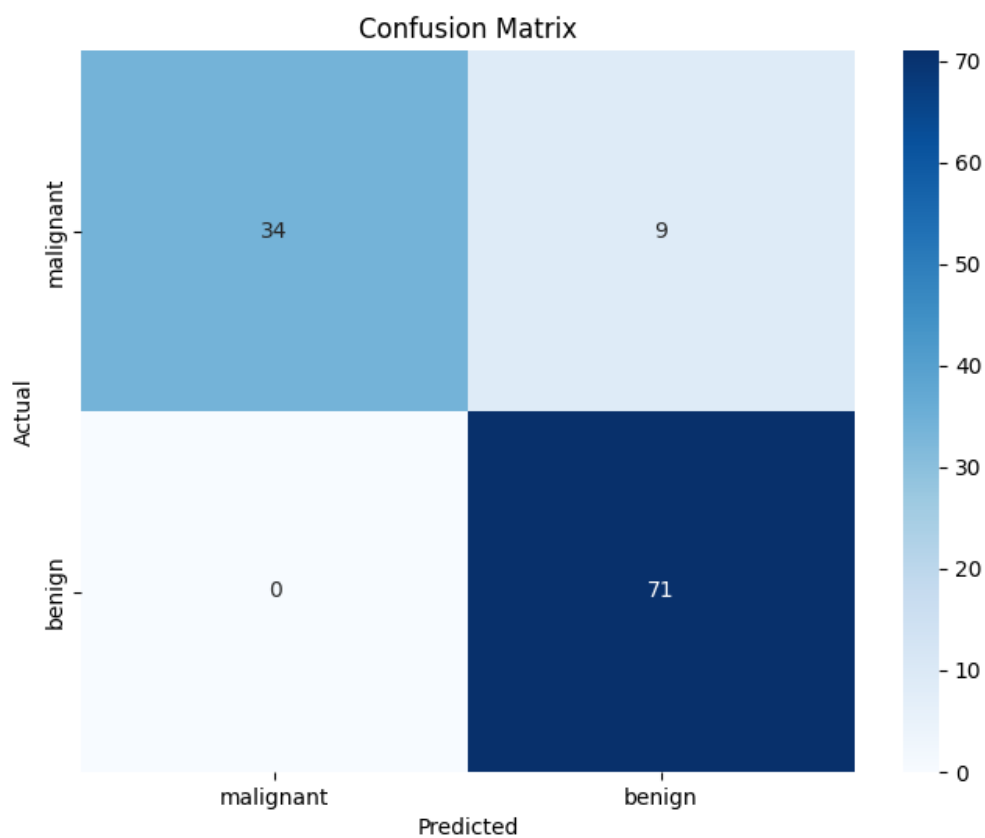
[۶۴۶.۱ ۹۲.۲۵ ۲۶.۹۹ ۱۴.۴]

[۶۰۲.۴ ۹۱.۴۳ ۱۷.۰۵ ۱۳.۹۶]

[۶۸۹.۴ ۹۸.۷۳ ۱۶.۷۴ ۱۵.۰۴]

[[۶۲۳.۹ ۹۲.۵۵ ۲۷.۸۵ ۱۴.۲۲]

مرتبہ سوم



Accuracy: 0.9210526315789473

False Predictions:

[۵۸۴.۱ ۹۰.۴۳ ۱۵.۷۹ ۱۳.۸]

[۷۱۶۶ .۹۸ ۲۵.۷۴ ۱۵.۰۸]

[۵۵۹.۲ ۸۸.۴ ۲۰.۸۲ ۱۳.۴۸]

[۵۸۸.۹ ۹۰.۶۳ ۲۲.۲۹ ۱۳.۷۷]

[۶۰۲.۴ ۹۱.۴۳ ۱۷.۰۵ ۱۳.۹۶]

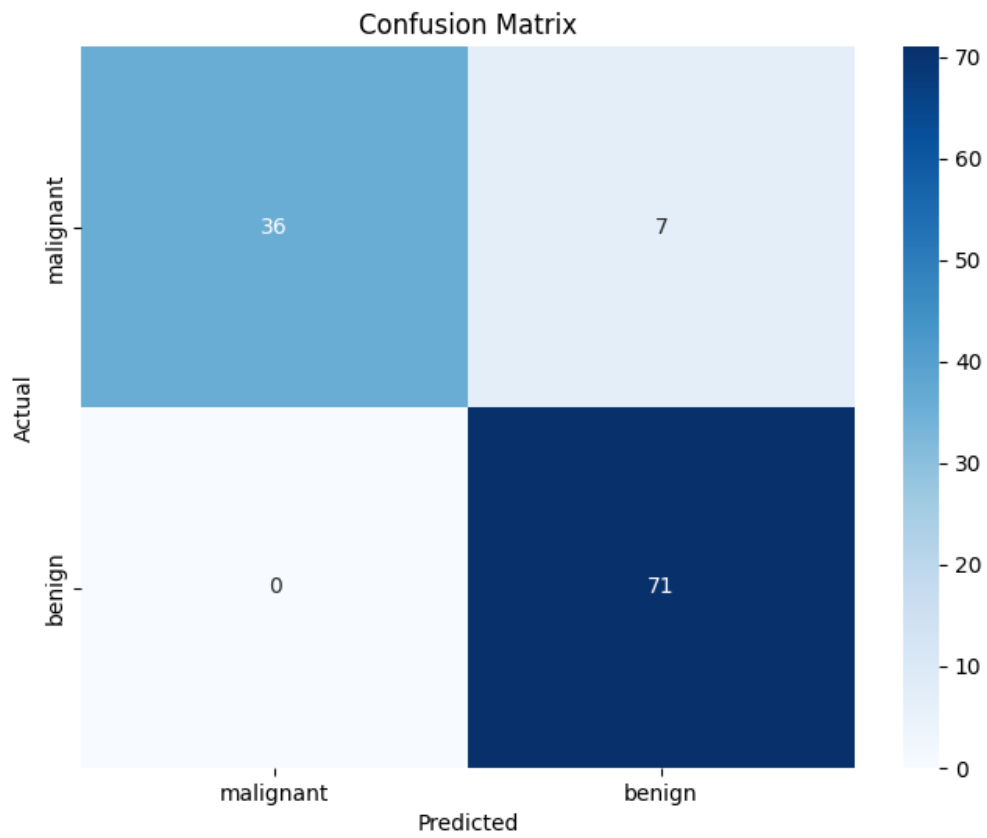
[۶۴۸.۲ ۹۴.۲۵ ۲۱.۴۶ ۱۴.۴۸]

[۷۱۳.۳ ۱۰۳.۲ ۱۷.۲۷ ۱۵.۳۲]

[۴۷۵.۹ ۸۳.۹۷ ۲۴.۰۴ ۱۲.۴۶]

[[۵۹۷.۸ ۹۱.۵۶ ۲۳.۷۵ ۱۳.۸۱]

گوسی



Accuracy: 0.9385964912280702

False Predictions:

[۵۸۴.۱ ۹۰.۴۳ ۱۵.۷۹ ۱۳.۸]

[۵۵۹.۲ ۸۸.۴ ۲۰.۸۲ ۱۳.۴۸]

[۵۸۸.۹ ۹۰.۶۳ ۲۲.۲۹ ۱۳.۷۷]

[۶۰۲.۴ ۹۱.۴۳ ۱۷.۰۵ ۱۳.۹۶]

[۶۴۸.۲ ۹۴.۲۵ ۲۱.۴۶ ۱۴.۴۸]

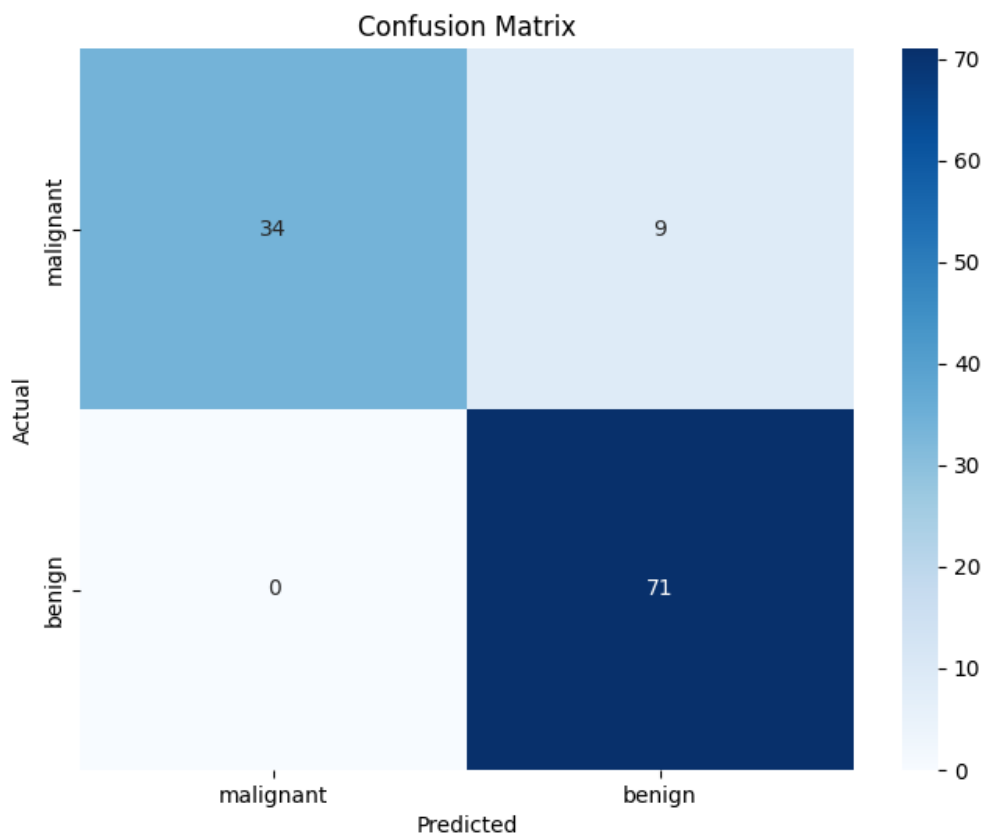
[۴۷۵.۹ ۸۳.۹۷ ۲۴.۰۴ ۱۲.۴۶]

[۵۹۷.۸ ۹۱.۵۶ ۲۳.۷۵ ۱۳.۸۱]

(ب)

با تغییر در ورودی برای بررسی نه ویژگی اول، و استفاده از کد موجود، نتایج زیر حاصل گردیده است.

خطی



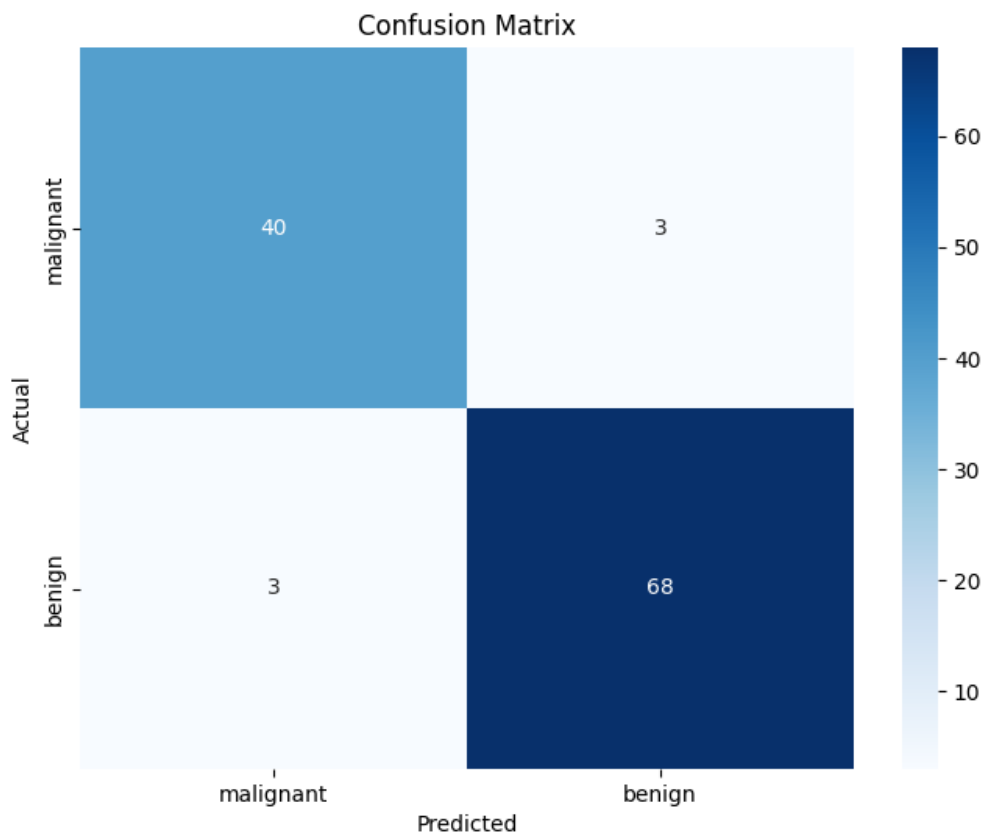
Accuracy: 0.9210526315789473

False Predictions:

[[1.38e+01 1.579e+01 9.043e+01 5.841e+02 1.007e-01 1.280e-01 7.789e-02 5.069e-02 1.662e-01]
[1.508e+01 2.574e+01 9.800e+01 7.166e+02 1.024e-01 9.769e-02 1.235e-01 6.553e-02 1.647e-01]
[1.348e+01 2.082e+01 8.840e+01 5.592e+02 1.016e-01 1.255e-01 1.063e-01 5.439e-02 1.720e-01]
[1.377e+01 2.229e+01 9.063e+01 5.889e+02 1.200e-01 1.267e-01 1.385e-01 6.562e-02 1.834e-01]
[1.369e+01 1.705e+01 9.143e+01 6.024e+02 1.096e-01 1.279e-01 9.789e-02 5.246e-02 1.908e-01]
[1.448e+01 2.146e+01 9.425e+01 6.482e+02 9.444e-02 9.947e-02 1.204e-01 4.938e-02 2.075e-01]
[1.532e+01 1.727e+01 1.032e+02 7.133e+02 1.335e-01 2.284e-01 2.448e-01 1.242e-01 2.398e-01]

1.246e+01 2.404e+01 8.397e+01 4.759e+02 1.186e-01 2.396e-01 2.273e-01 8.543e-02 2.030e-01
1.381e+01 2.375e+01 9.156e+01 5.978e+02 1.323e-01 1.768e-01 1.558e-01 9.176e-02 2.251e-01

مرتبہ دوم

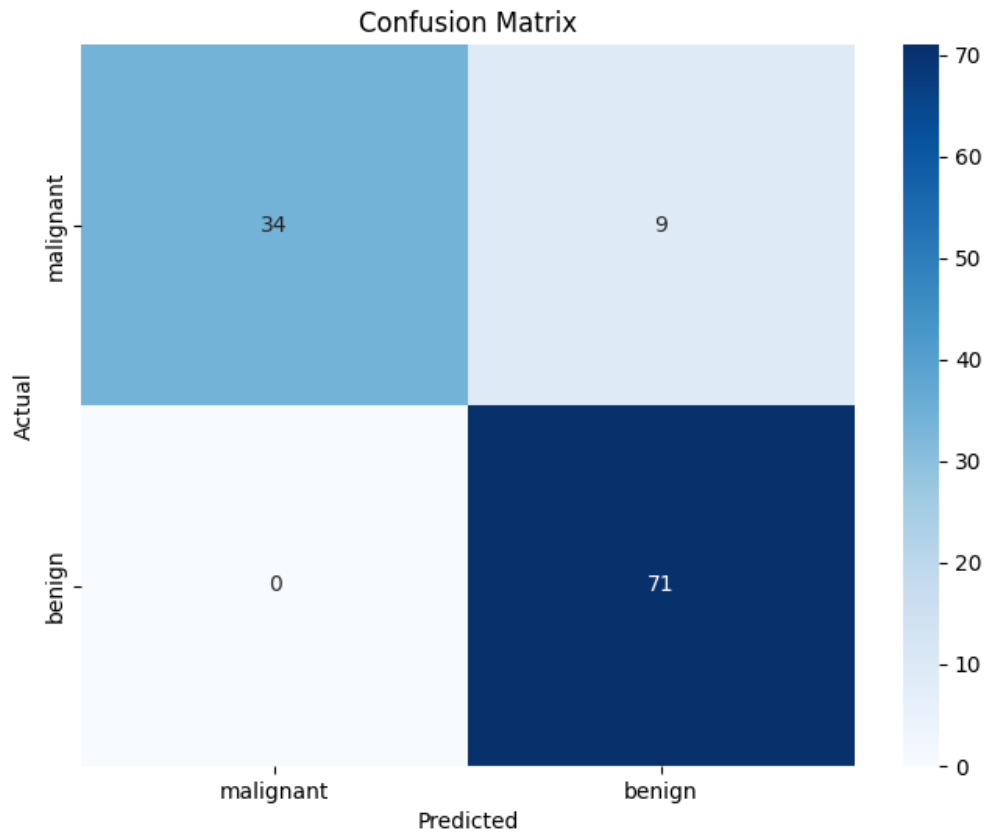


Accuracy: 0.9473684210526315

False Predictions:

1.38e+01 1.579e+01 9.043e+01 5.841e+02 1.007e-01 1.280e-01 7.789e-02 5.069e-02 1.662e-01
1.462e+01 2.402e+01 9.457e+01 6.627e+02 8.974e-02 8.606e-02 3.102e-02 2.957e-02 1.685e-01
1.311e+01 2.254e+01 8.702e+01 5.294e+02 1.002e-01 1.483e-01 8.705e-02 5.102e-02 1.850e-01
1.348e+01 2.082e+01 8.840e+01 5.592e+02 1.016e-01 1.255e-01 1.063e-01 5.439e-02 1.720e-01
1.369e+01 1.705e+01 9.143e+01 6.024e+02 1.096e-01 1.279e-01 9.789e-02 5.246e-02 1.908e-01
1.422e+01 2.785e+01 9.255e+01 6.239e+02 8.223e-02 1.039e-01 1.103e-01 4.408e-02 1.342e-01

مرتبه سوم

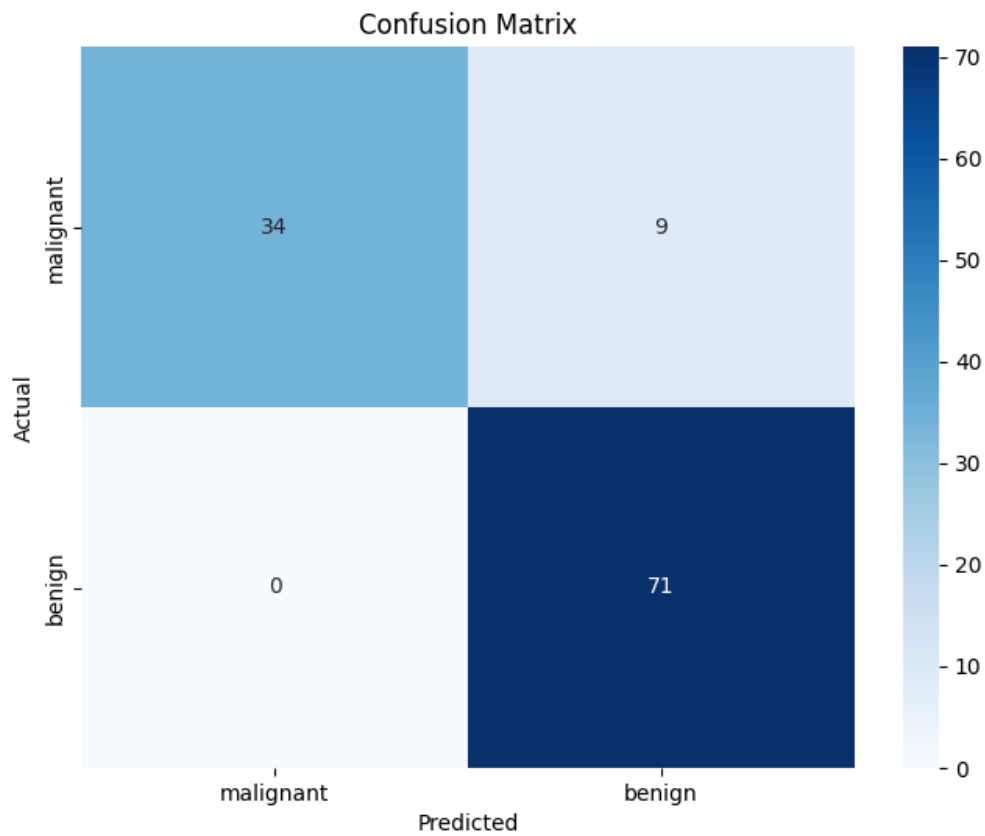


Accuracy: 0.9210526315789473

False Predictions:

```
[[1.38e+01 1.579e+01 9.043e+01 5.841e+02 1.007e-01 1.280e-01 7.789e-02 5.069e-02 1.662e-01[
] 1.508e+01 2.574e+01 9.800e+01 7.166e+02 1.024e-01 9.769e-02 1.235e-01 6.553e-02 1.647e-01[
] 1.348e+01 2.082e+01 8.840e+01 5.592e+02 1.016e-01 1.255e-01 1.063e-01 5.439e-02 1.720e-01[
] 1.377e+01 2.229e+01 9.063e+01 5.889e+02 1.200e-01 1.267e-01 1.385e-01 6.526e-02 1.834e-01[
] 1.396e+01 1.705e+01 9.143e+01 6.024e+02 1.096e-01 1.279e-01 9.789e-02 5.246e-02 1.908e-01[
] 1.448e+01 2.146e+01 9.425e+01 6.482e+02 9.444e-02 9.947e-02 1.204e-01 4.938e-02 2.075e-01[
] 1.532e+01 1.727e+01 1.032e+02 7.133e+02 1.335e-01 2.284e-01 2.448e-01 1.242e-01 2.398e-01[
] 1.246e+01 2.404e+01 8.397e+01 4.759e+02 1.186e-01 2.396e-01 2.273e-01 8.543e-02 2.030e-01[
] 1.381e+01 2.375e+01 9.156e+01 5.978e+02 1.323e-01 1.768e-01 1.558e-01 9.176e-02 2.251e-01[[
```

گوسی



Accuracy: 0.9210526315789473

False Predictions:

```
[[1.38e+01 1.579e+01 9.043e+01 5.841e+02 1.007e-01 1.280e-01 7.789e-02 5.069e-02 1.662e-01[
] 1.508e+01 2.574e+01 9.800e+01 7.166e+02 1.024e-01 9.769e-02 1.235e-01 6.553e-02 1.647e-01[
] 1.348e+01 2.082e+01 8.840e+01 5.592e+02 1.016e-01 1.255e-01 1.063e-01 5.439e-02 1.720e-01[
] 1.377e+01 2.229e+01 9.063e+01 5.889e+02 1.200e-01 1.267e-01 1.385e-01 6.526e-02 1.834e-01[
] 1.396e+01 1.705e+01 9.143e+01 6.024e+02 1.096e-01 1.279e-01 9.789e-02 5.246e-02 1.908e-01[
] 1.448e+01 2.146e+01 9.425e+01 6.482e+02 9.444e-02 9.947e-02 1.204e-01 4.938e-02 2.075e-01[
] 1.532e+01 1.727e+01 1.032e+02 7.133e+02 1.335e-01 2.284e-01 2.448e-01 1.242e-01 2.398e-01[
] 1.246e+01 2.404e+01 8.397e+01 4.759e+02 1.186e-01 2.396e-01 2.273e-01 8.534e-02 2.030e-01[
] 1.381e+01 2.375e+01 9.156e+01 5.978e+02 1.323e-01 1.768e-01 1.558e-01 9.176e-02 2.251e-01[[
```

ج) با توجه به مقادیر دقت مدل ها، شاهد افزایش زیادی در دقت با افزایش تعداد ویژگی ها نیستیم .

د) با توجه به تفاوت بانک داده مورد اشاره در صورت سوال این قسمت قابل حل نمیباشد. اما برای دسته بندی یک داده جدید میتوان از کد دستوری زیر استفاده نمود :

```
predictions = svm_classifier.predict(X_test)
```

که میتوان داده مورد نظر را در آرگومان تابع وارد نمود تا نتیجه دسته بندی به دست آید.