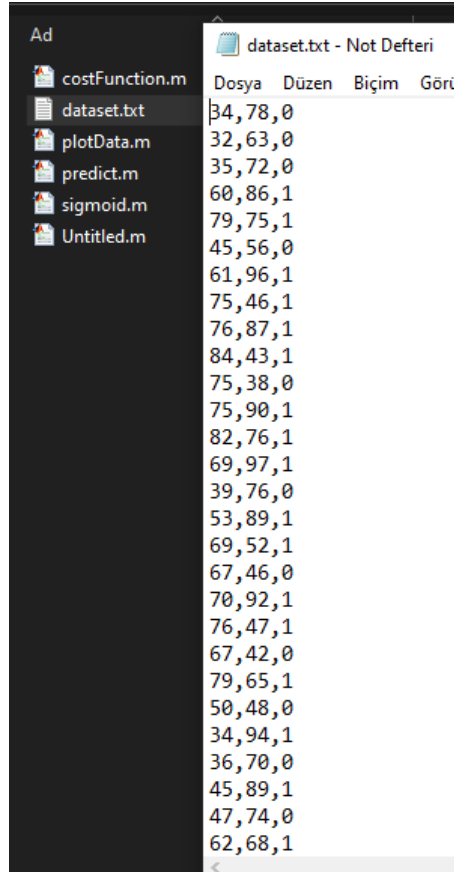


# LOGISTIC REGRESYON

Bir öğrencinin önceki başvuru sahiplerinden bir üniversite geçmiş verilerine kabul edilip edilmediğini tahmin etmek için bir lojistik regresyon modeli oluşturalım:

- 2 puan

- kabul kararı (0 ve 1) , olsun.



The screenshot shows a MATLAB file explorer window with a list of files on the left and a preview of the 'dataset.txt' file on the right. The files listed are: costFunction.m, dataset.txt, plotData.m, predict.m, sigmoid.m, and Untitled.m. The 'dataset.txt' file is selected, and its contents are displayed in a table with two columns: 'Dosya' (File) and 'Görü' (View). The data is as follows:

Dosya	Görü
dataset.txt	34,78,0
plotData.m	32,63,0
predict.m	35,72,0
sigmoid.m	60,86,1
Untitled.m	79,75,1
	45,56,0
	61,96,1
	75,46,1
	76,87,1
	84,43,1
	75,38,0
	75,90,1
	82,76,1
	69,97,1
	39,76,0
	53,89,1
	69,52,1
	67,46,0
	70,92,1
	76,47,1
	67,42,0
	79,65,1
	50,48,0
	34,94,1
	36,70,0
	45,89,1
	47,74,0
	62,68,1

Verileri kabul veya ret gören şekilde ayrıca ana kodlarımızı görelim:

```
% veri setini tükleyelim.
data = load('C:\Users\ASUS\Desktop\DataSetOdv\dataset.txt');
X = data(:, [1,2]);
%örnek puanı.
y = data(:, 3);
%örnek sıfır veya bir.

plotData(X,y);

xlabel('Soru skoru 1');
ylabel('Soru skoru 2');

legend('Kabul', 'Ret');

[m, n]= size(X);
X = [ones(m, 1) X];
%x'e müdahale terimi ekler.

%uydurma parametrelerini başlatmak için.
initial_theta = zeros(n +1, 1);

% maliyet ve gradyan hesaplamak için.
[cost, grad] = costFuction(initial_theta, X, y);

fprintf('İlk tetadaki maliyet (sıfırlar): %f\n', cost);

disp('İlk tetadaki gradyan (sıfırlar):'); disp(grad);

% fminunc için seçeneği ayarlamak için.
options = options(@fminunc, 'Algorithm', 'Quasi-Newton', 'GradOnj', 'on', 'MaxIter', 400);

%optimal teta'yı elde etmek için fminunc çalıştırmak için.
[theta,cost] = fminunc(@(t)(costFunction(t, X, y)), initial_theta, options);

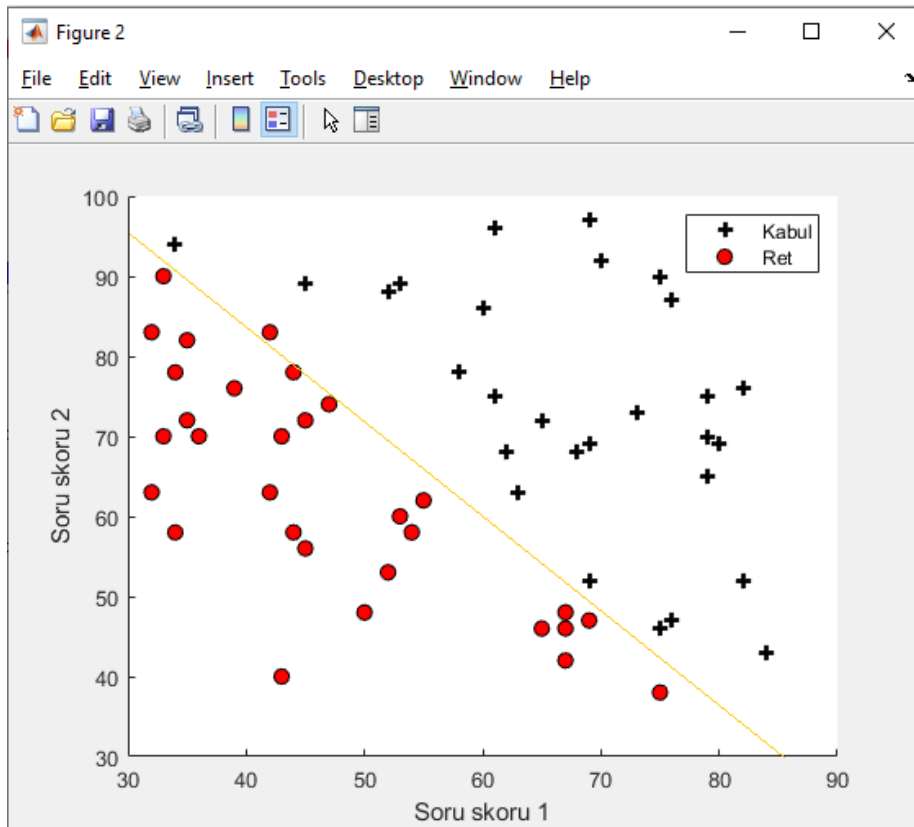
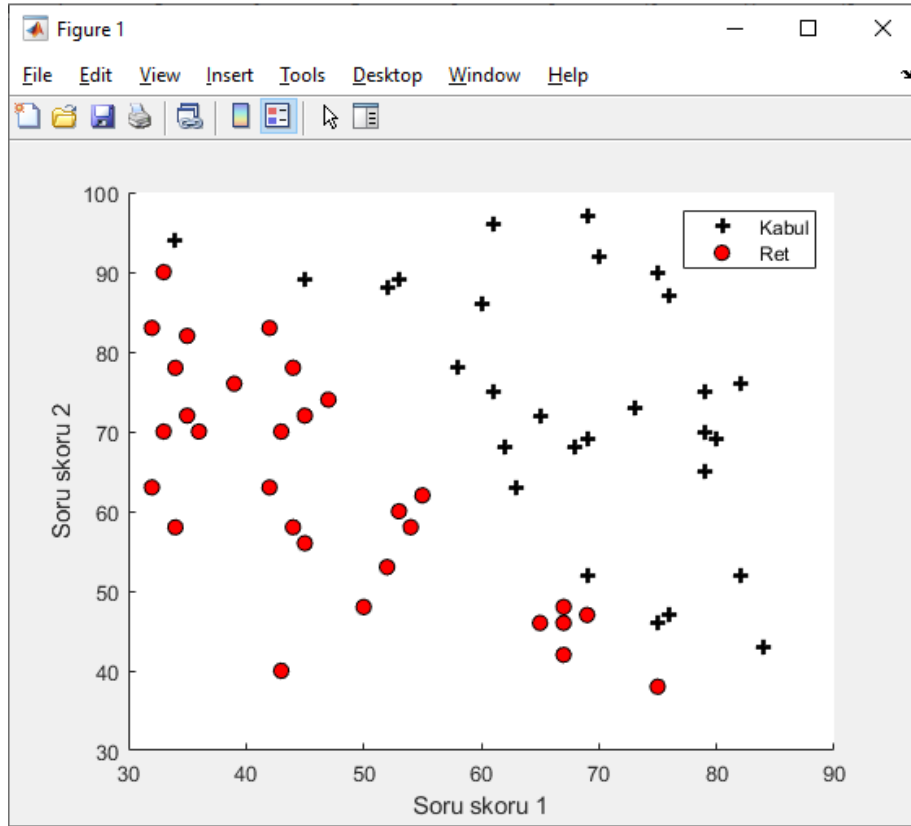
plotDecisionBoundary(theta, X, y);
hold on;
xlabel('Soru skoru 1');
ylabel('Soru skoru 2');
legend('Kabul','Ret');
hold off;

%45 ve 85 puan alan bir öğrencinin olasılığını tahmin etmek için.
prob = sigmoid([1 45 85] * theta);
fprintf('45 ve 85 puanları olan bir öğrenci için kabul olasılığını tahmin ediyoruz %f\n\n',
prob);
```

%eğitim setimizde hesaplamak için.

```
p = predict(theta, X);
```

```
fprintf('Eğitim Doğruluğu: %f\n', mean(double(p == y)) * 100);
```



## Maliyet Fonksiyonu:

Cost function in Logistic regression

$$J(\theta) = \frac{1}{m} \sum_{i=1}^m \text{Cost}(h_{\theta}(x^{(i)}), y^{(i)})$$
$$= -\frac{1}{m} \left[ \sum_{i=1}^m y^{(i)} \log h_{\theta}(x^{(i)}) + (1 - y^{(i)}) \log (1 - h_{\theta}(x^{(i)})) \right]$$

Gradient of the cost is a vector of the same length as  $\theta$

$$\frac{\partial J(\theta)}{\partial \theta_j} = \frac{1}{m} \sum_{i=1}^m (h_{\theta}(x^{(i)}) - y^{(i)}) x_j^{(i)}$$

```
function [J, grad] = costFunction(theta, X, y)
```

```
    m = length(y);
```

```
    J = 0;
```

```
    grad = zeros(size(theta));
```

```
    h = sigmoid(X * theta);
```

```
    J = (-1/m)*sum(y.*log(h) + (1-y).*log(1-h));
```

```
    for j = 1:size(grad)
```

```
        grad(j) = (1/m) * sum((h - y) .* X(:,j));
```

```
    end
```

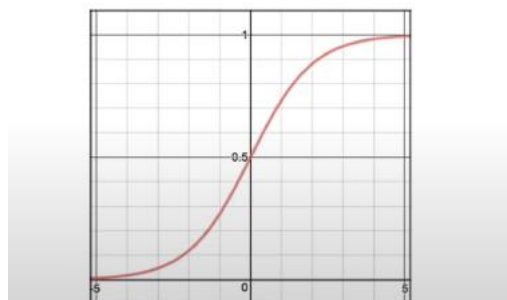
```
end
```

## Sigmoid Fonksiyonu:

Logistic regression hypothesis  $h_{\theta}(x) = g(\theta^T x)$

Sigmoid function  $g(z) = \frac{1}{1 + e^{-z}}$

Code should also work with vectors and matrices



```
function g = sigmoid(z)
```

```
    g = zeros(size(z));
```

```
    [m, n] = size(z)
```

```

        for i = 1:m
            for j = 1:n
                g(i, j) = 1/(1+ exp(-z(i, j)));
            end
        end
    end
end

```

### Tahmin Fonksiyonu:

```

function p = predict(theta, X)

m = size(X, 1);
p = zeros(m, 1);

p = sigmoid(X * theta);

    for i = 1:m
        if (p(i) >= 0.5)
            p(i) = 1;
        else
            p(i) = 0;
        end
    end
end

```

### Temel Grafik Çizdirimi:

```

function plotData(X,y)

figure; hold on;

pos = find(y==1); %olası örnekler.
neg = find(y==0); %olumsuz örnekler.

plot(X(pos,1), X(pos ,2), 'k+', 'LineWidth', 2, 'MarkerSize', 7);
plot(X(neg,1), X(neg, 2), 'ko', 'MarkerFaceColor', 'r', 'MarkerSize', 7);

hold off;

end

```

Matlab Editor interface showing four files: dataset.txt, plotData.m, predict.m, sigmoid.m, and costFunction.m.

**dataset.txt**

```
1 34,78,0
2 32,63,0
3 35,72,0
4 60,86,1
5 79,75,1
6 45,56,0
7 61,96,1
8 75,46,1
9 76,87,1
10 84,43,1
11 75,38,0
12 75,90,1
13 82,76,1
14 69,97,1
15 39,76,0
16 53,89,1
17 69,52,1
18 67,46,0
19 70,92,1
20 76,47,1
21 67,42,0
22 79,65,1
23 50,48,0
24 34,94,1
25 36,70,0
26 45,89,1
27 47,74,0
28 62,68,1
```

**plotData.m**

```
1 function plotData(X,y)
2
3 figure; hold on;
4
5 pos = find(y==1); %olası örnekler.
6 neg = find(y==0); %olumsuz örnekler.
7
8 plot(X(pos,1), X(pos,2), 'k+', 'LineWidth', 2, 'MarkerSize', 7);
9 plot(X(neg,1), X(neg,2), 'ko', 'MarkerFaceColor', 'r', 'MarkerSize', 7);
10
11 hold off;
12
13 end
```

**predict.m**

```
1 function p = predict(theta, X)
2
3 m = size(X, 1);
4 p = zeros(m, 1);
5
6 p = sigmoid(X * theta);
7
8 for i = 1:m
9     if (p(i) >= 0.5)
10         p(i) = 1;
11     else
12         p(i) = 0;
13     end
14 end
15
```

**sigmoid.m**

```
1 function g = sigmoid(z)
2
3 g = zeros(size(z));
4 [m, n] = size(z);
5
6 for i = 1:m
7     for j = 1:n
8         g(i, j) = 1/(1+ exp(-z(i, j)));
9     end
10 end
11
12 end
13
14
```

**costFunction.m**

```
1 function [J, grad] = costFunction(theta, X, y)
2
3 m = length(y);
4 J = 0;
5 grad = zeros(size(theta));
6
7 h = sigmoid(X * theta);
8 J = (-1/m)*sum(y.*log(h) +
9
10 for j = 1:size(grad)
11     grad(j) = (1/m) * sum((
12
13 end
14
15 end
```

**Untitled.m**

```
1 %veri setini tükleyelim.
2 data = load('C:\Users\ASUS\Desktop\DataSetOdv\dataset.txt');
3 X = data(:, [1,2]);
4 %örnek puanı.
5 y = data(:, 3);
6 %örnek sıfır veya bir.
7
8 plotData(X,y);
9
10 xlabel('Soru skoru 1');
11 ylabel('Soru skoru 2');
12
13 legend('Kabul', 'Ret');
14
15 [m, n] = size(X);
16 X = [ones(m, 1) X];
17 %x'e müdahale terimi ekler.
18
19 %uydurma parametrelerini başlatmak için.
20 initial_theta = zeros(n+1, 1);
21
22 %maliyet ve gradyan hesaplamak için.
23 [cost, grad] = costFunction(initial_theta, X, y);
24
25 fprintf('İlk tetadaki maliyet (sıfırlar): %f\n', cost);
26 disp('İlk tetadaki gradyan (sıfırlar):'); disp(grad);
27
28 %fminunc için seçeneği ayarlamak için.
29 options = options(@fminunc, 'Algorithm', 'Quasi-Newton', 'GradOnj', 'on', 'MaxIter', 400);
30
31 %optimal teta'yi elde etmek için fminunc çalıştırmak için.
32 [theta,cost] = fminunc(@(t)(costFunction(t, X, y)), initial_theta, options);
33
34 plotDecisionBoundary(theta, X, y);
35 hold on;
36 xlabel('Soru skoru 1');
37 ylabel('Soru skoru 2');
38 legend('Kabul','Ret');
39 hold off;
40
41 %45 ve 85 puan alan bir öğrencinin olasılığını tahmin etmek için.
42 prob = sigmoid([1 45 85] * theta);
43 fprintf('45 ve 85 puanları olan bir öğrenci için kabul olasılığını tahmin ediyoruz %f\n\n',
44
45 %eğitim setimizde hesaplamak için.
46 p = predict(theta, X);
47 fprintf('Eğitim Doğruluğu: %f\n', mean(double(p == y)) * 100);
```