Models (/github/ramizallahverdiyev/Models/tree/main)
/ Logistic_Regression.ipynb (/github/ramizallahverdiyev/Models/tree/main/Logistic_Regression.ipynb)

📘 1. ƏSAS ANLAYIŞLAR

🔹 1.1 Reqressiya və Klassifikasiya

- **Reqressiya (Regression)** modelləri, kəsilməz (numerical) bir dəyişəni proqnozlaşdırır.

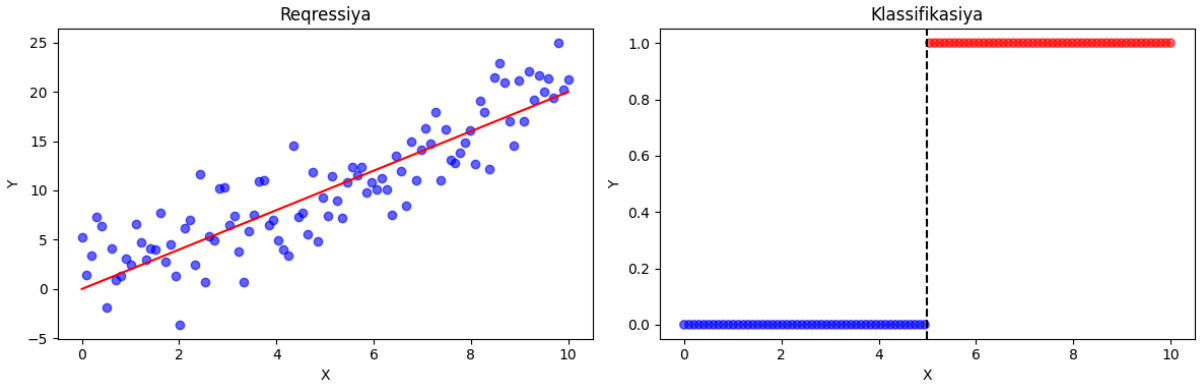
> Məsələn: Evin qiyməti, temperatur, maaş və s.

- **Klassifikasiya (Classification)** modelləri, kateqorik (categorical) bir nəticəni proqnozlaşdırır.

> Məsələn: E-poçt spamdır, yoxsa yox? Müştəri bir malı alacaq, yoxsa yox?

➡️ Loqistik **reqressiya**, adına baxmayaraq, bir **klassifikasiya alqoritmidir.**

In [ ]:



🔹 1.2 Niyə Proqnoz Yox, Ehtimal?

- Xətti reqressiya 0 ilə 1-dən kənar dəyərləri proqnozlaşdıra bilər → klassifikasiyada isə bu **məntiqsiz** olar.
- Loqistik reqressiya bu problemi **ehtimal proqnozu** ilə həll edir:

Əvvəlcə ehtimal hesablanır:

$$p = \frac{1}{1 + e^{-(\beta_0 + \beta_1 x_1 + \cdots + \beta_n x_n)}}$$

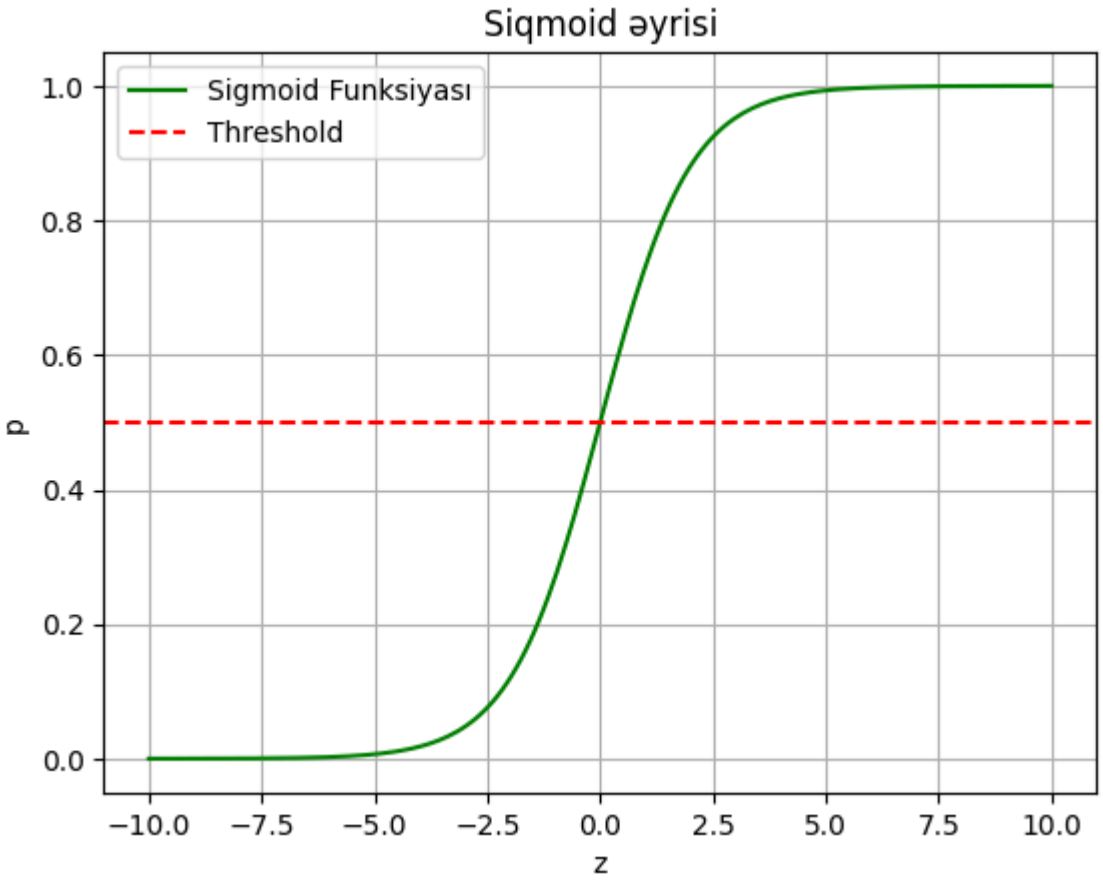
Sonra sinifə qərar verilir (məsələn, hədd 0.5):

- $$p \geq 0.5 \Rightarrow \text{sinif} = 1$$
- $$p < 0.5 \Rightarrow \text{sinif} = 0$$

Üstünlüyü:

- Ehtimal sayəsində modelin **nə qədər əmin** olduğunu öyrənə bilərik.
- Qərar həddini (threshold) problemin təbiətinə görə tənzimləyə bilərik.

In [ ]:



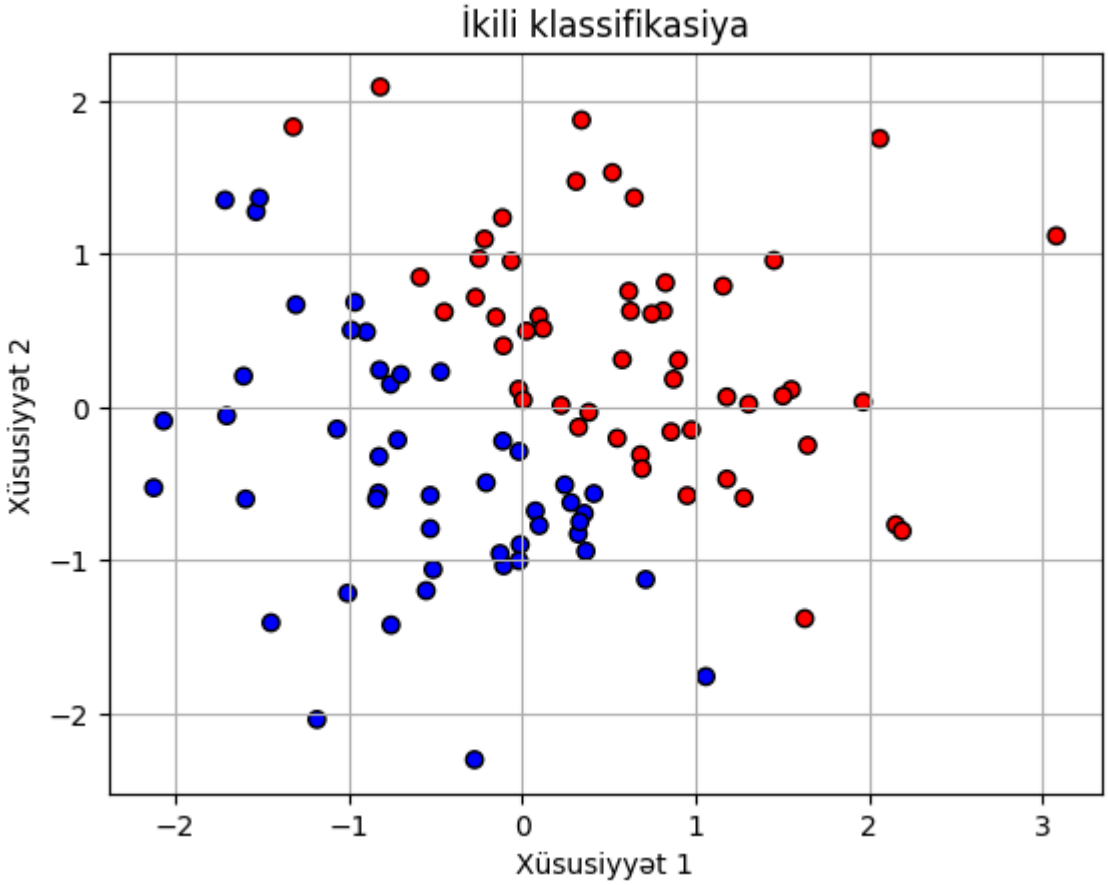
🔹 1.3 Klassifikasiyanın Növləri

✅ 1.3.1 İkili Klassifikasiya (Binary Classification)

- İki sinif var: 0 və ya 1

> Məsələn: Şəxs Xəstədirmi? (Bəli/Xeyr)

In [ ]:

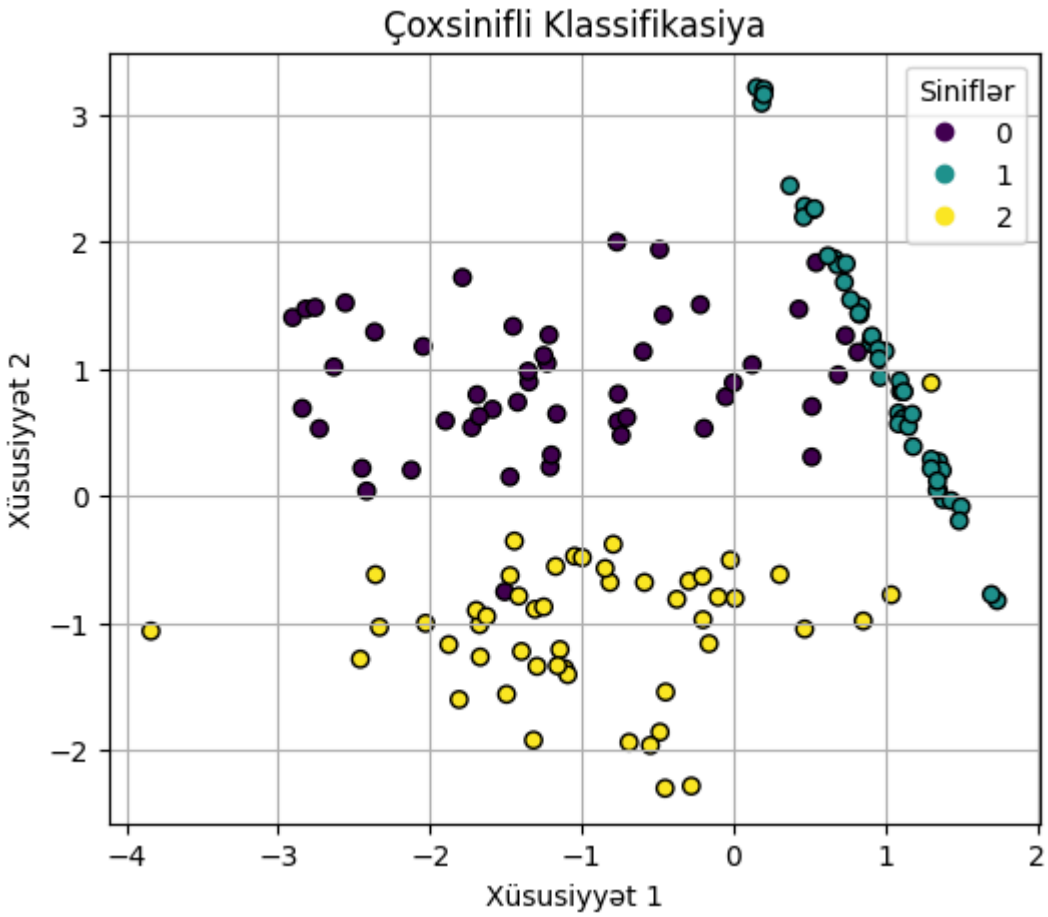


✅ 1.3.2 Çoxsinifli Klassifikasiya (Multiclass Classification)

- Birdən çox sinif var, lakin **hər müşahidə yalnız bir sinifə** aiddir.

> Məsələn: Müştəri A, B və ya C paketlərindən birini seçir.

In [ ]:

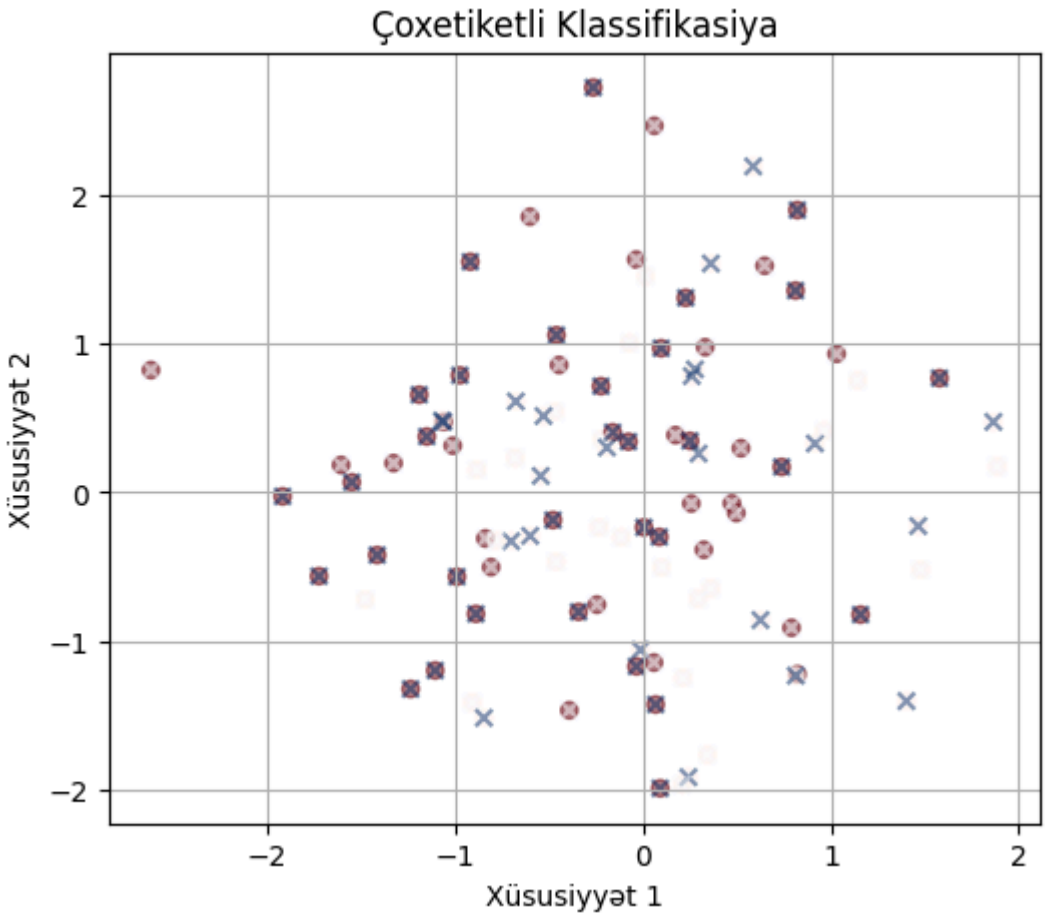


✅ 1.3.3 Çoxetiketli Klassifikasiya (Multilabel Classification)

- Bir müşahidə **eyni zamanda birdən çox sinifə** aid ola bilər.

> Məsələn: Bir xəbər həm “idman”, həm də “siyasət” kateqoriyasına aid ola bilər.

In [ ]:

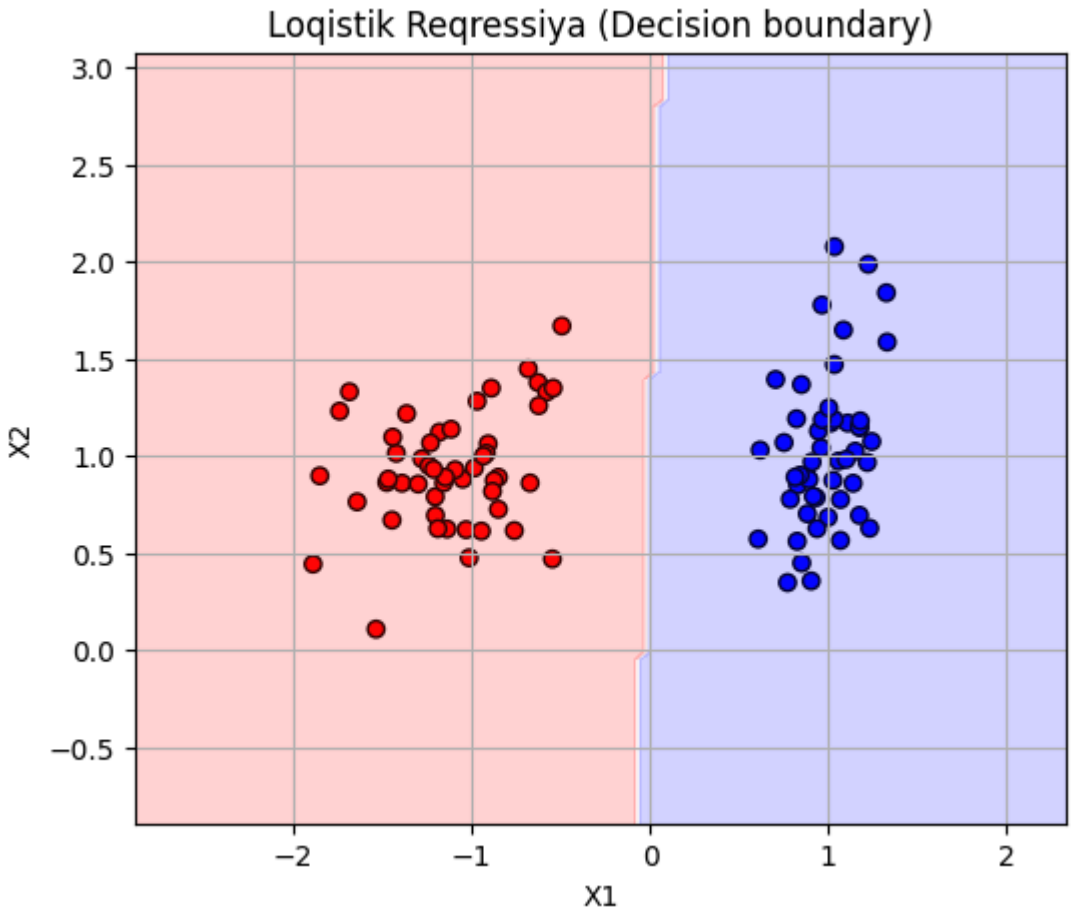


🔹 1.4 Loqistik Reqressiya Nə Zaman İstifadə Edilir?

Loqistik reqressiya aşağıdakı hallarda üstünlük verilir:

- ✅ Hədəf dəyişən **kateqorikdirsə** (adətən 0/1),
- ✅ Sərbəst dəyişənlər **ədədi və ya kateqorik (dummy) olaraq kodlanıbsa**,
- ✅ Proqnoz yox, **ehtimal üzərindən qərar** verilmək istənilirsə,
- ✅ Modelin **şərh edilə bilməsi** vacibdirsə (məsələn, səhiyyə, hüquq, iqtisadiyyat),

In [ ]:



📘 2. MODELİN RİYAZİ ƏSASLARI

(Loqistik Reqressiyanın Quruluşu, Düsturu və Ehtimal Hesablanması)

🔹 2.1 Niyə Xətti Model Yox?

Xətti reqressiya klassifikasiyada istifadə edilsəydi, nəticə belə olardı:

$$\hat{y} = \beta_0 + \beta_1 x_1 + \cdots + \beta_n x_n$$

Lakin bu modelin nəticəsi istənilən ədəd ola bilər: -1.2, 0.8, 3.4, və s.

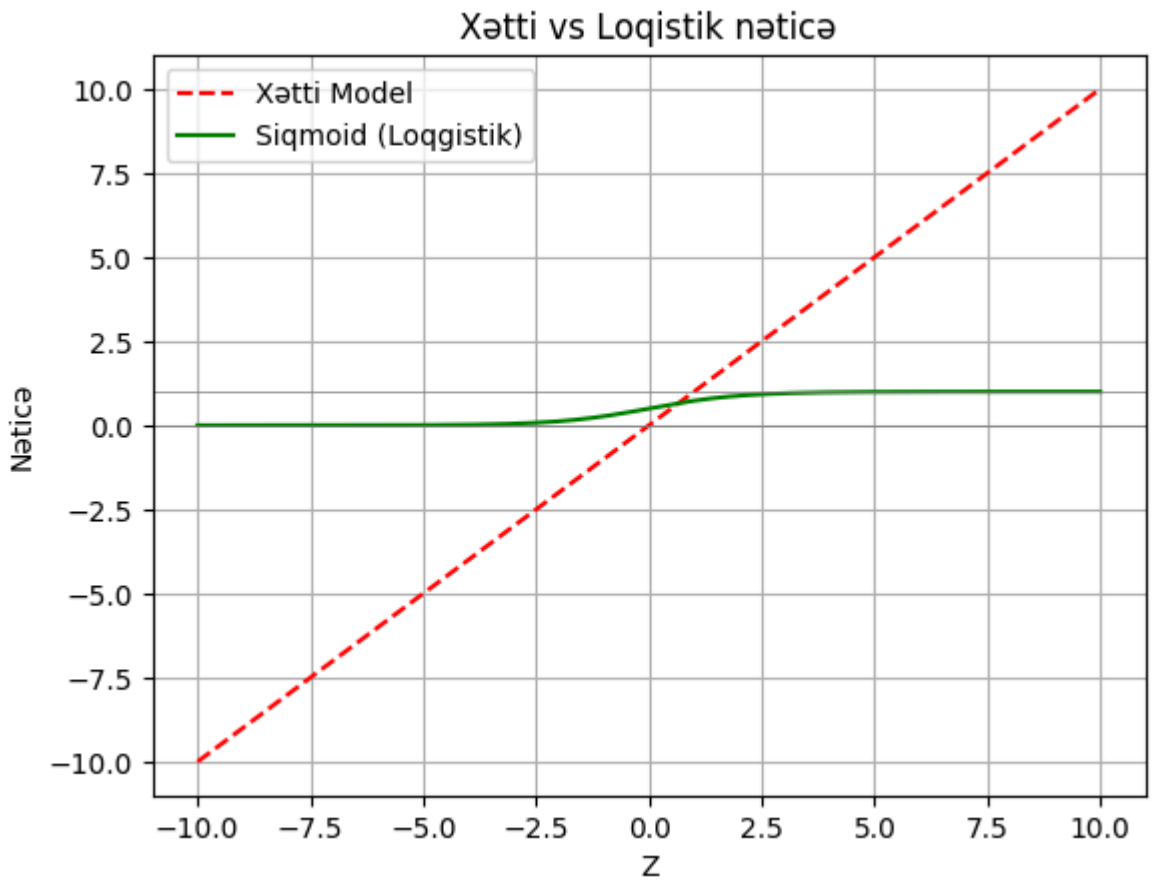
Lakin klassifikasiya problemlərində:

- Nəticə

  $$[0, 1]$$

  aralığında **bir ehtimal** olmalıdır.
- Buna görə də nəticəni bu aralığa **sıxmaq** lazımdır.

In [ ]:



🔹 2.2 Logit Funksiyası və Əmsal Anlayışı

Bir hadisənin baş vermə ehtimalı p olarsa, baş verməmə ehtimalı 1 - p olar.
Bu iki dəyər nisbətləndikdə **əmsal (odds) (ehtimal nisbəti)** əldə edilir:

$$\text{odds} = \frac{p}{1-p}$$

Odds şərhi:

- $\text{odds} = 1$
  baş vermə və baş verməmə ehtimalı bərabərdir
- > 1
  baş vermə ehtimalı daha yüksəkdir
- < 1
  baş verməmə ehtimalı daha yüksəkdir

Loqistik reqressiyada odds'un **loqarifmi** (log-odds və ya **logit**) götürülür:

📌 Logit funksiyası:

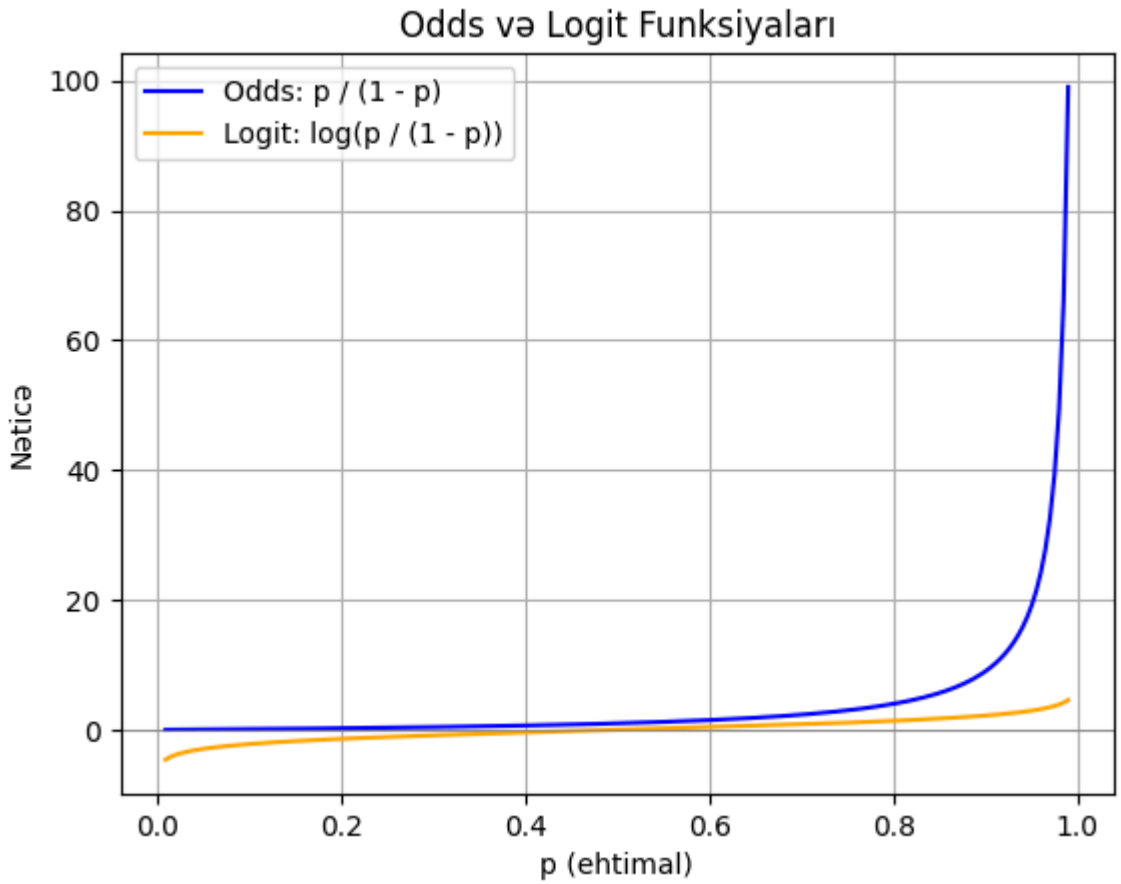
$$\text{logit}(p) = \log\left(\frac{p}{1-p}\right)$$

Bu transformasiya sayəsində nəticə

$$(-\infty, +\infty)$$

aralığına yayılır.

In [ ]:



🔹 2.3 Loqistik Reqressiya Modelinin Tənliyi

Loqistik reqressiyada logit(p), xətti reqressiya şəklində ifadə edilir:

$$\log\left(\frac{p}{1-p}\right) = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \cdots + \beta_n x_n$$

Sol tərəf loqit (log-odds), sağ tərəf xətti modeldir.

🔹 2.4 Ehtimalın Proqnozlaşdırılması (Siqmoid Funksiyası)

Bu tənlik

$$p$$

ilə yenidən yazılsa:

$$p = \frac{1}{1 + e^{-(\beta_0 + \beta_1 x_1 + \cdots + \beta_n x_n)}}$$

Bu düstur **siqmoid funksiyası** kimi tanınır.

✅ Xüsusiyyətləri:

- Çıxış aralığı:

$$(0, 1)$$

- Z böyüdükcə

$$p \to 1$$

- Z kiçildikcə

- $$p \to 0$$
  $$z = 0$$
  üçün
  $$p = 0.5$$
  → Qərar sərhədi (Threshold)

🔹 2.5 Siqmoid Funksiyası

Funksiyanın riyazi şəkli:

$$\sigma(z) = \frac{1}{1 + e^{-z}}$$

- Z oxu: modelin hesabladığı xətti nəticə
- Y oxu: proqnozlaşdırılan **ehtimal**

✅ 2.6 Xülasə – Loqistik Reqressiya Prosesi

1. Xətti skor hesablanır:

$$z = \beta_0 + \beta_1 x_1 + \cdots + \beta_n x_n$$

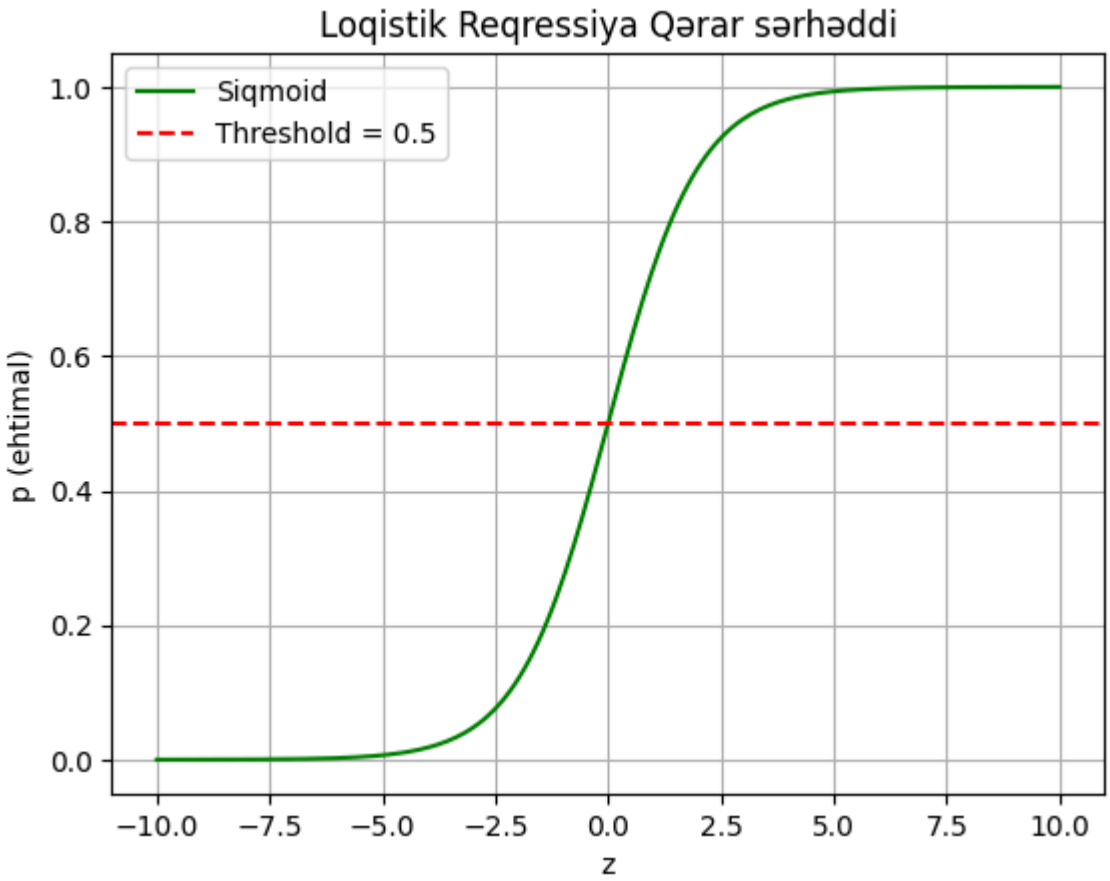
2. Siqmoid ilə ehtimala çevrilir:

$$p = \frac{1}{1 + e^{-z}}$$

3. Sinif müəyyən edilir:

- $$p \geq 0.5 \Rightarrow \text{sinif} = 1$$
- $$p < 0.5 \Rightarrow \text{sinif} = 0$$

In [ ]:



📘 3. FƏRZİYYƏLƏR

(Loqistik Reqressiyanın Etibarlı Olması Üçün Tələb Olunan Şərtlər)

🔹 3.1 Asılı Dəyişənin Kateqorik Olması

Loqistik reqressiyanın əsas məqsədi **klassifikasiya** etməkdir.

Ən geniş yayılmış hal: **ikili (binary) klassifikasiya**

Asılı dəyişən:

$$y \in \{0, 1\}$$

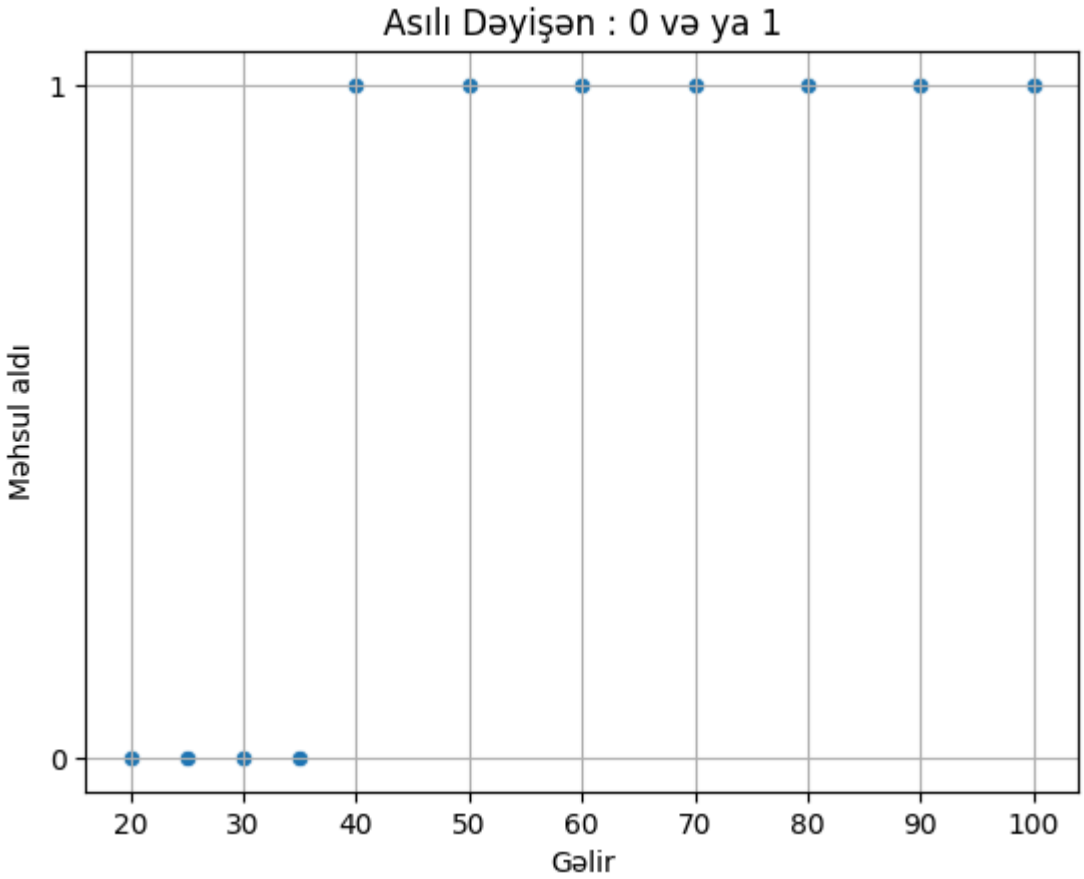
Nümunə: Xəstədir? Bəli (1) / Xeyr (0)

Loqistik reqressiya çoxsinifli (multiclass) və çoxetiketli (multilabel) strukturlara da genişləndirilə bilər.

Lakin hər halda hədəf dəyişən **kateqorik** olmalıdır.

> 📌 Əgər kəsilməz (numerical) bir dəyər proqnozlaşdırılırsa, loqistik reqressiya deyil, xətti reqressiya istifadə olunur.

In [ ]:



🔹 3.2 Müşahidələrin Bir-birindən Asılı Olmaması

Modelin əsas fərziyyələrindən biri hər müşahidənin müstəqil olmasıdır.

- Bir sətirdəki müşahidə nəticəsi, başqa bir sətrin nəticəsi ilə əlaqəli olmamalıdır.
- Zaman asılılığı (time series) və ya eyni xüsusiyyətdən təkrarlanan ölçmələr varsa, bu fərziyyə pozulur.

> 📌 Belə hallar üçün zaman seriyası (time series) və ya mixed-effects modelləri uyğundur.

🔹 3.3 Inputlar (X) ilə logit(p) Arasında Xətti Əlaqə Olması

Bu fərziyyə, loqistik reqressiyanın ən kritik strukturlarından biridir.

Loqistik reqressiya bunu fərz edir:

$$\text{logit}(p) = \log\left(\frac{p}{1-p}\right) = \beta_0 + \beta_1 x_1 + \cdots + \beta_n x_n$$

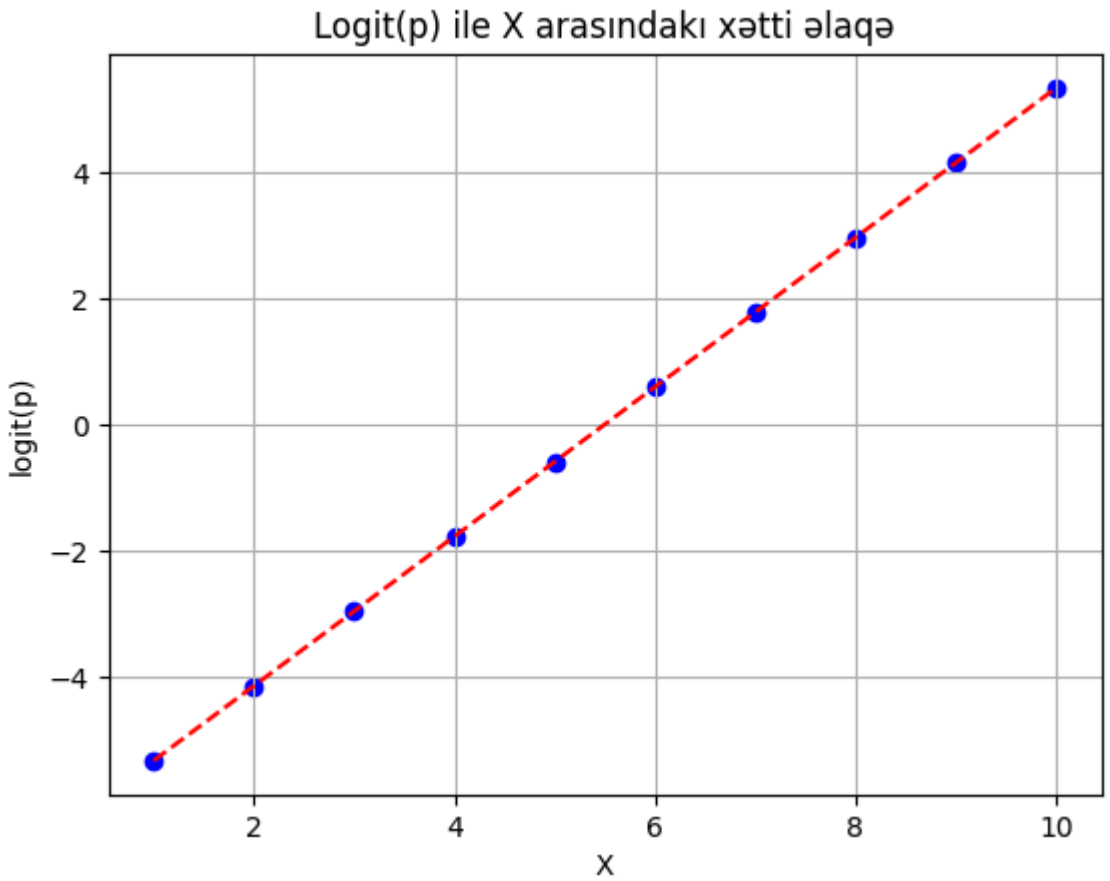
Yəni **logit(p)** ilə müstəqil dəyişənlər arasında **xətti bir əlaqə** olmalıdır.

Əgər bu əlaqə xətti deyilsə:

- Box-Tidwell testi və ya
- Qismən qalıq (Partial residual) qrafikləri ilə yoxlama aparılmalıdır.

📌 Lazım gələrsə dəyişənlərə transformasiya (log, root, polynomial) tətbiq edilməlidir.

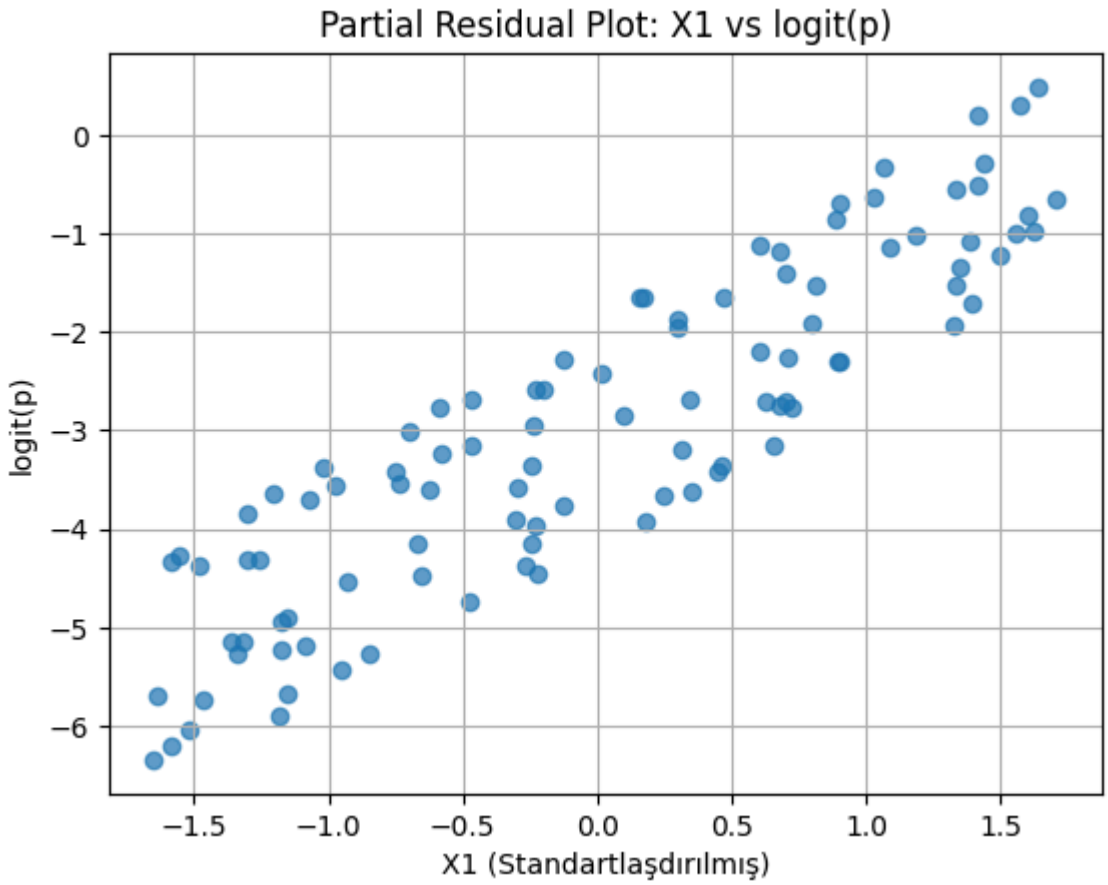
In [ ]:



In [ ]:

```
Optimization terminated successfully.
         Current function value: 0.229009
         Iterations 9
                           Logit Regression Results
==============================================================================
Dep. Variable:                      y   No. Observations:                  100
Model:                          Logit   Df Residuals:                       95
Method:                           MLE   Df Model:                            4
Date:                Mon, 04 Aug 2025   Pseudo R-squ.:                  0.3391
Time:                        09:16:33   Log-Likelihood:                -22.901
converged:                       True   LL-Null:                       -34.652
Covariance Type:            nonrobust   LLR p-value:                 0.0001005
==============================================================================
                 coef    std err          z      P>|z|      [0.025      0.975]
------------------------------------------------------------------------------
const         -4.6614      6.262     -0.744      0.457     -16.935       7.612
X1            -0.4314      2.739     -0.157      0.875      -5.799       4.937
X1log          0.4050      0.964      0.420      0.674      -1.485       2.295
X2             0.7235      1.695      0.427      0.670      -2.599       4.046
X2log         -0.4398      0.704     -0.624      0.532      -1.821       0.941
==============================================================================
```

In [ ]:



🔹 3.4 Çoxlu Xəttilik (Multicollinearity) Olmamalı

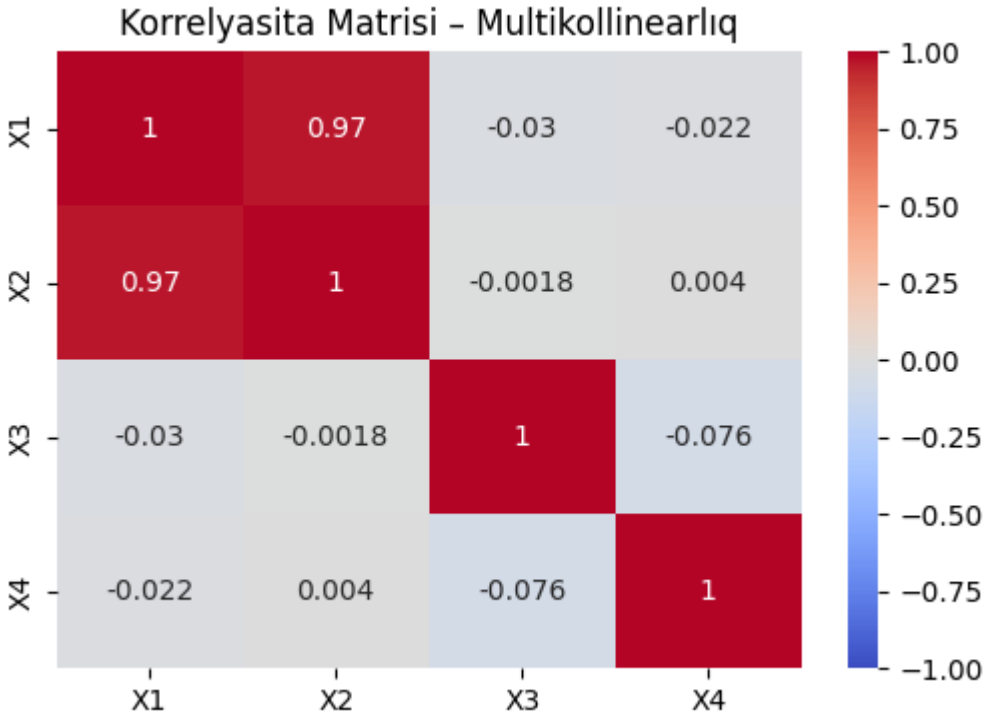
Giriş dəyişənləri bir-biri ilə yüksək korrelyasiyaya malikdirsə:

- Əmsalların etibarlılığı azalır
- Standart xətalar böyüyür

Həll: **VIF (Variance Inflation Factor)** ilə yoxlanılır.

> 📌 Adətən VIF > 5 xəbərdarlıqdır, VIF > 10 ciddi problem sayılır.

In [ ]:



🔹 3.5 Kifayət Qədər Müşahidə

Loqistik reqressiya maksimum ehtimal (maximum likelihood) metoduna əsaslanır. Bu da böyük dataset tələb edir.

Ümumi təklif:

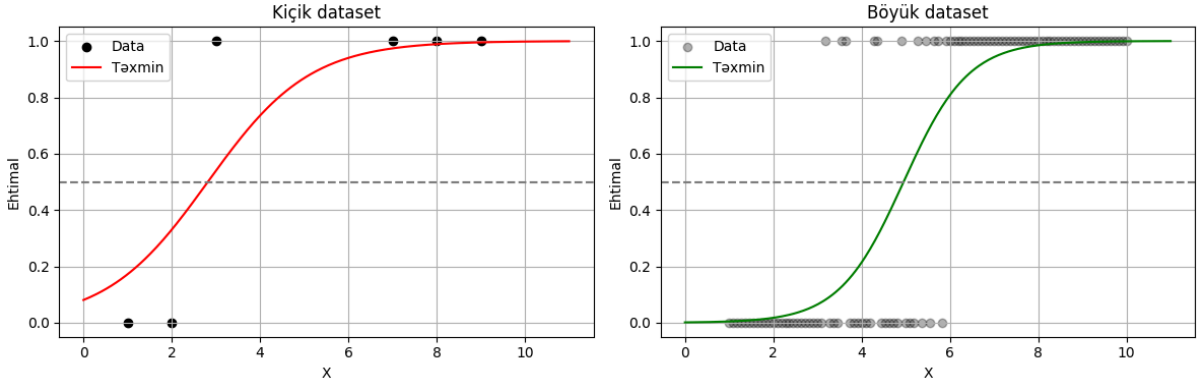
- Hər müstəqil dəyişən üçün **ən azı 10** müşahidə
- Mümkündürsə **20+** müşahidə (xüsusilə aşağı tezlikli siniflər varsa)

Az müşahidə ilə:

- Model qeyri-sabitləşir
- Əmsallar şişir
- Konvergensiya problemləri yarana bilər

> 📌 Bu halda regularization, bootstrap və ya Bayesian modellər nəzərdən keçirilə bilər.

In [ ]:



📘 4. Modelin Qiymətləndirilməsi

Loqistik reqressiya modelləri klassifikasiya etdiyi üçün klassik reqressiya göstəricilərindən fərqli olaraq, klassifikasiyanın doğruluğunu ölçən xüsusi göstəricilər istifadə olunur.

In [ ]:

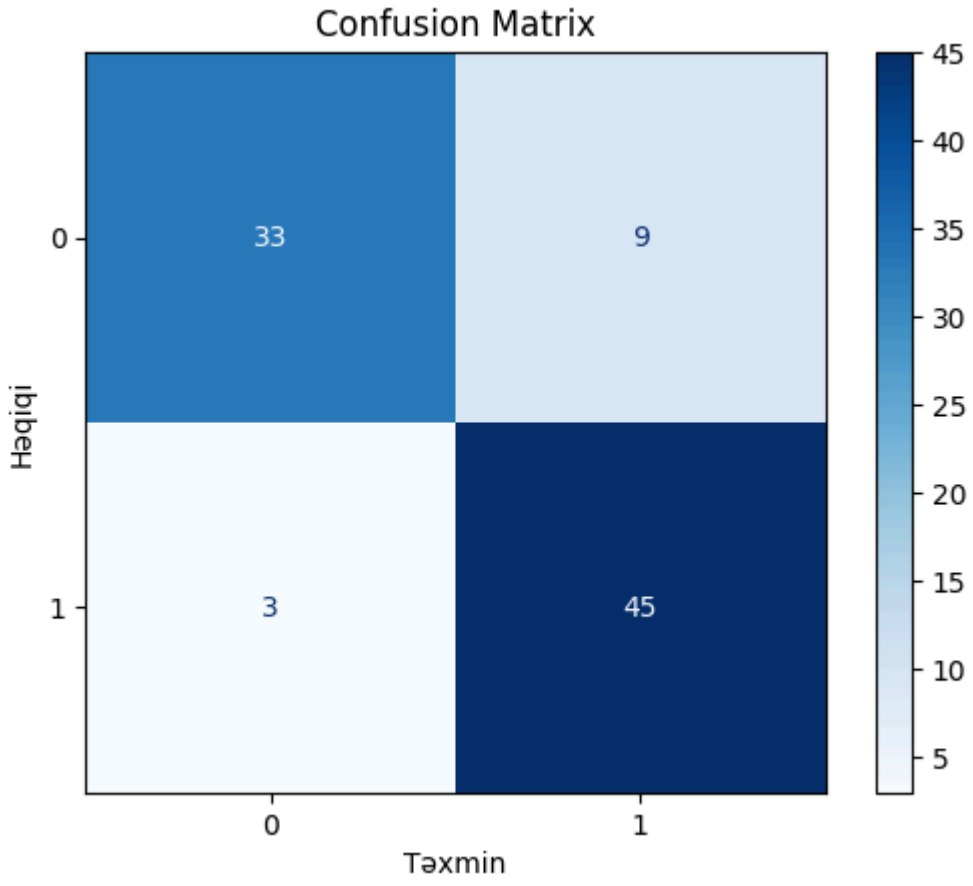
🔢 4.1. Qarışıqlıq Matrisi (Confusion Matrix)

	Actual = 1	**Actual = 0**
Predicted = 1	TP (True Positive)	FP (False Positive)
Predicted = 0	FN (False Negative)	TN (True Negative)

İzahı:

- TP: Xəstə olan birinə “xəstə” demək
- FP: Sağlam birinə “xəstə” demək (yanlış həyəcan)
- FN: Xəstə birinə “sağlam” demək (çox təhlükəli)
- TN: Sağlam birinə “sağlam” demək

In [ ]:



```
              precision    recall  f1-score   support

           0       0.92      0.79      0.85        42
           1       0.83      0.94      0.88        48

    accuracy                           0.87        90
   macro avg       0.88      0.86      0.86        90
weighted avg       0.87      0.87      0.87        90
```

📏 4.2. Dəqiqlik (Accuracy)

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN}$$

- Ümumi müvəffəqiyyət dərəcəsidir.
- Lakin balanssız siniflərdə yanıldıcı ola bilər.
- Xəstəxana məsələsində, modelin xəstələrdən və sağlamlardan neçəsini doğru tapdığını göstərir.

In [ ]:

```
Accuracy: 0.867
```

🎯 4.3. Kəskinlik (Precision)

$$\text{Precision} = \frac{TP}{TP + FP}$$

- Modelin “pozitiv” dediyi nümunələrin nə qədər doğru olduğunu ölçür.
- Yanlış pozitiv hallarda vacibdir.
- Xəstəxana məsələsində, modelin xəstə dediklərinin həqiqətən nə qədərinin xəstə olduğunu göstərir.

In [ ]:

```
Precision: 0.833
```

💡 Həssaslıq (Recall)

$$\text{Recall} = \frac{TP}{TP + FN}$$

- Həqiqi pozitivləri tapma dərəcəsidir.
- Yanlış neqativlərin (buraxılmış pozitivlərin) vacib olduğu hallarda əhəmiyyətlidir.
- Xəstəxana məsələsində, həqiqətən xəstə olanlardan neçəsinin doğru tapılmasıdır.

In [ ]:

```
Recall: 0.938
```

⚖️ 4.5. F1-Skor (Harmonik Orta)

$$\text{F1} = 2 \cdot \frac{\text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}}$$

- Kəskinlik (Precision) və Həssaslıq (Recall) arasında balansı ölçür.

- Xüsusilə balanssız (imbalanced) datasetlərdə faydalıdır.

In [ ]:

```
F1 Score: 0.882
```

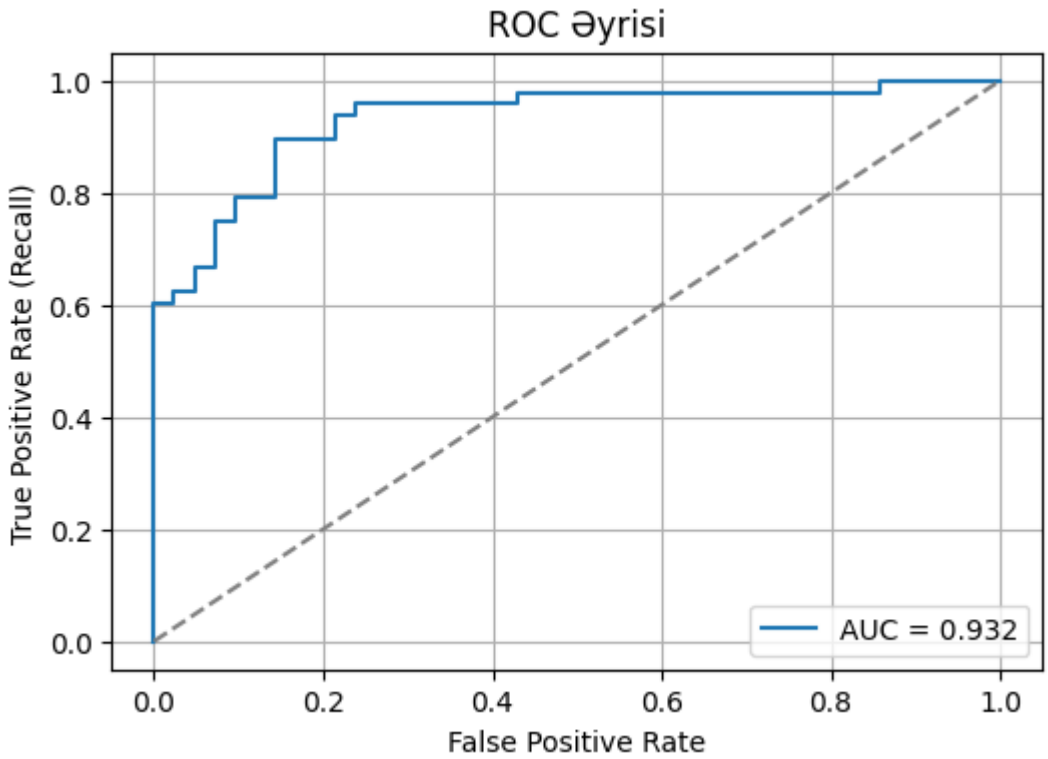
📉 4.6. ROC Əyrisi və AUC

- **ROC əyrisi:** TPR (True Positive Rate) vs FPR (False Positive Rate)
- **AUC:** ROC əyrisinin altında qalan sahə

Şərh:

- AUC = 1.0 → mükəmməl model
- AUC = 0.5 → tamamilə təsadüfi
- AUC ≥ 0.8 → güclü model

In [ ]:



📦 4.7. Log-loss (Loqarifmik Xəta)

$$\text{LogLoss} = -\frac{1}{n}\sum_{i=1}^{n}[y_i \cdot \log(p_i) + (1-y_i) \cdot \log(1-p_i)]$$

- Modelin ehtimal proqnozlarının nə qədər "dəqiq" olduğunu ölçür.
- Yanlış klassifikasiyada aşağı ehtimal → böyük cəza

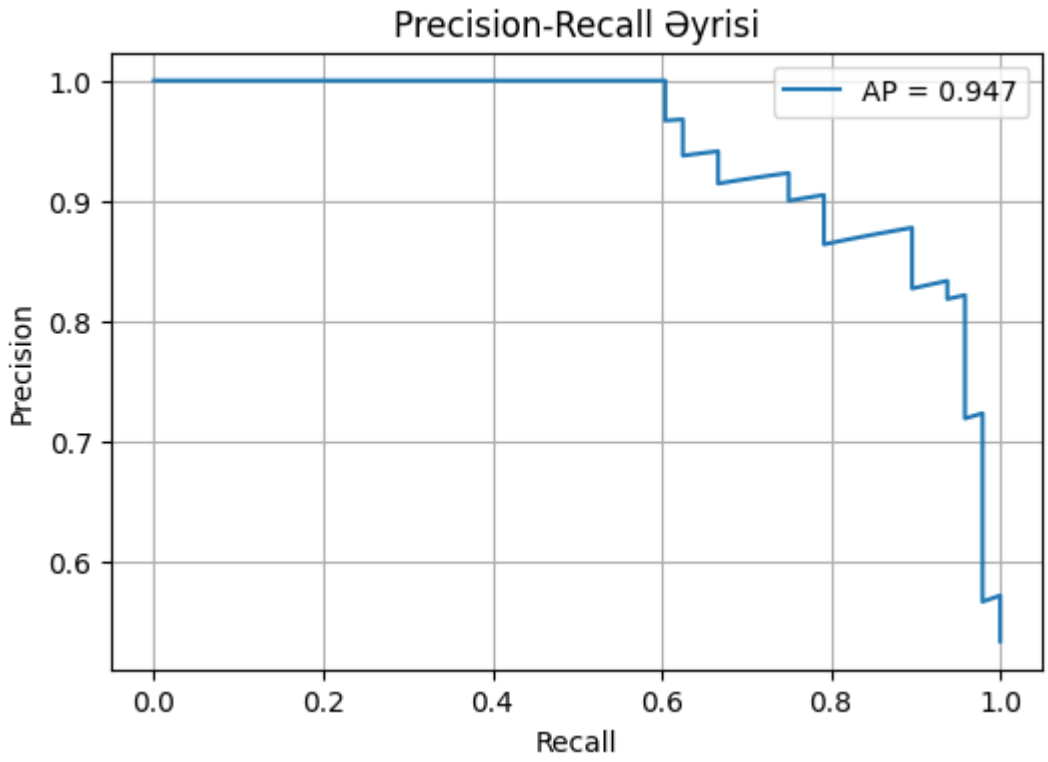
In [ ]:

```
Log-loss: 0.3250
```

📊 4.8. Kəskinlik-Həssaslıq Əyrisi (Precision-Recall Curve)

- ROC-a alternativ olaraq Kəskinlik və Həssaslıq dəyərlərinin dəyişməsini göstərir.
- Xüsusilə **balanssız sinif** hallarında daha aydınlaşdırıcıdır.

In [ ]:



🔚 Müqayisə Cədvəli

Metrik	İzahı	Nə zaman istifadə olunur?
Accuracy	Ümumi dəqiqlik	Siniflər balanslıdırsa
Precision	Pozitiv proqnoz dəqiqliyi	Yanlış pozitivlər kritikdirsə (FP)
Recall	Həqiqi pozitivləri tapma	Yanlış neqativlər kritikdirsə (FN)
F1-Score	Precision + Recall balansı	Balanssız sinif varsa
ROC-AUC	Təsnifatın müvəffəqiyyəti	Adətən müqayisə üçün
Log-loss	Ehtimal keyfiyyəti	Proqnoz etibarlılığı tələb olunursa
PR Curve	Dəqiqlik vs Geri çağırma əyrisi	Pozitiv sinif azdırsa üstünlük verilir

📘 5. Model Diaqnostikası (Ətraflı Təhlil)

Modeli öyrətdik, lakin yetərl olmaya bilər! Həqiqətən yaxşı işləyirmi? Bütün dəyişənlər bərabər dərəcədə təsir edirmi? Hipotezlər təmin olunurmu?

Model Diaqnostikası ilə bunları aydınlaşdırmaq mümkündür.

📉 5.1. Qalıqların Analizi (Deviance Residuals)

Qalıq (Residual) modelin proqnozlaşdırdığı ehtimal ilə həqiqi sinif arasındakı fərqdir.

Loqistik reqressiyada klassik qalıqlar(RSS) əvəzinə **Deviance Residuals** istifadə olunur.

Deviance (Ümumi Uyğunluq İtkisi) düsturu belədir:

$$D = -2 \sum_{i=1}^{n} [y_i \log(\hat{p}_i) + (1 - y_i) \log(1 - \hat{p}_i)]$$

Burada:

$$\hat{p}_i$$

modelin proqnozlaşdırdığı pozitiv sinif ehtimalıdır.

$$y_i$$

həqiqi sinif (0 və ya 1) dəyəridir.

Deviance Residual düsturu isə:

$$r_i = \text{sign}(y_i - \hat{p}_i) \cdot \sqrt{-2 \left[y_i \log(\hat{p}_i) + (1 - y_i) \log(1 - \hat{p}_i) \right]}$$

Misal:

Həqiqi nəticə:

$$y_i = 1$$

Proqnoz:

$$\hat{p}_i = 0.9$$

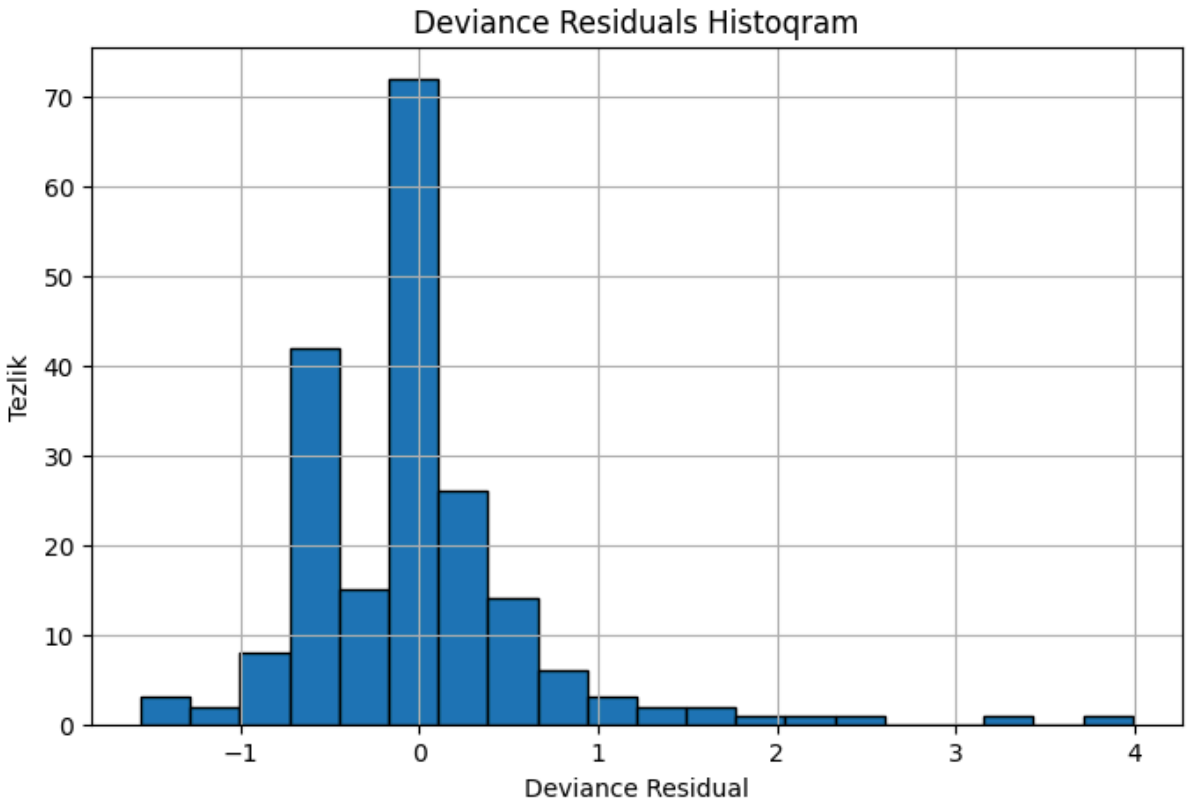
Bu halda deviance residual:

$$r_i = +1 \times \sqrt{-2 \times (1 \times \log(0.9) + 0 \times \log(0.1))} = \sqrt{-2 \times (-0.105)} \approx 0.458$$

Model müşahidəni yaxşı proqnozlaşdırıb (kiçik qalıq).

In [ ]:

```
Optimization terminated successfully.
         Current function value: 0.240485
         Iterations 18
```



🧨 5.2. Təsirli Müşahidələr (Influential Observations)

Bəzi müşahidələr modelə həddindən artıq təsir edə bilər. Bunlar adətən kənar dəyərlər, uc nöqtələrdəki müşahidələr və ya yüksək leverage dəyərinə malik nöqtələr olur.

Cook's Distance düsturu:

$$D_i = \frac{\sum_{j=1}^{n} (\hat{y}_j - \hat{y}_{j(i)})^2}{p \cdot MSE}$$

Burada:

$$\hat{y}_{j(i)}$$

(i) - inci müşahidə olmadan edilən proqnoz,

$$p$$

parametr sayı,

$$MSE$$

orta kvadrat xətası.

Qayda:

$$D_i > \frac{4}{n}$$

olarsa, diqqətlə araşdırılmalıdır

Leverage (Ling) düsturu:

$$h_i = x_i^T (X^T X)^{-1} x_i$$

Burada:

$$h_i > \frac{2p}{n}$$

olarsa, yüksək leverage hesab olunur.

Misal:

Parametr sayı:

$$p = 5$$

Müşahidə sayı:

$$n = 100$$

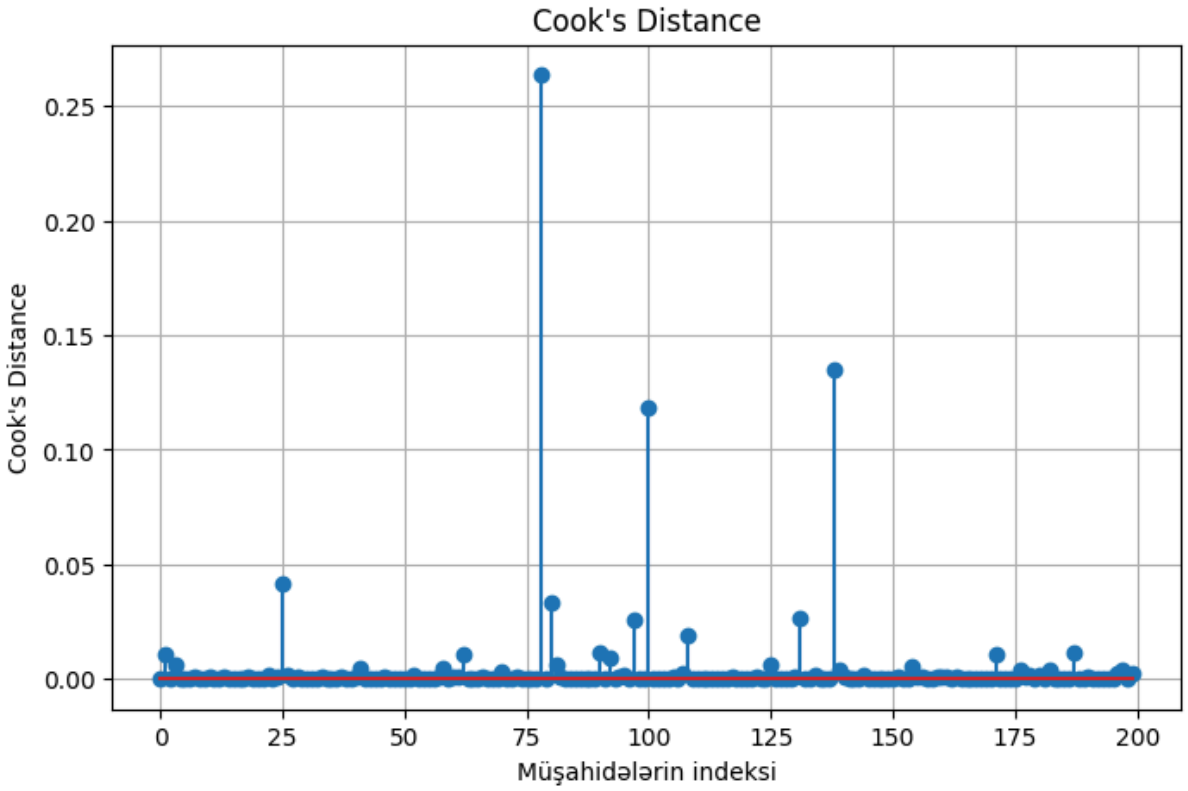
Təsirli müşahidə üçün hədd:

$$D_i > 0.04$$

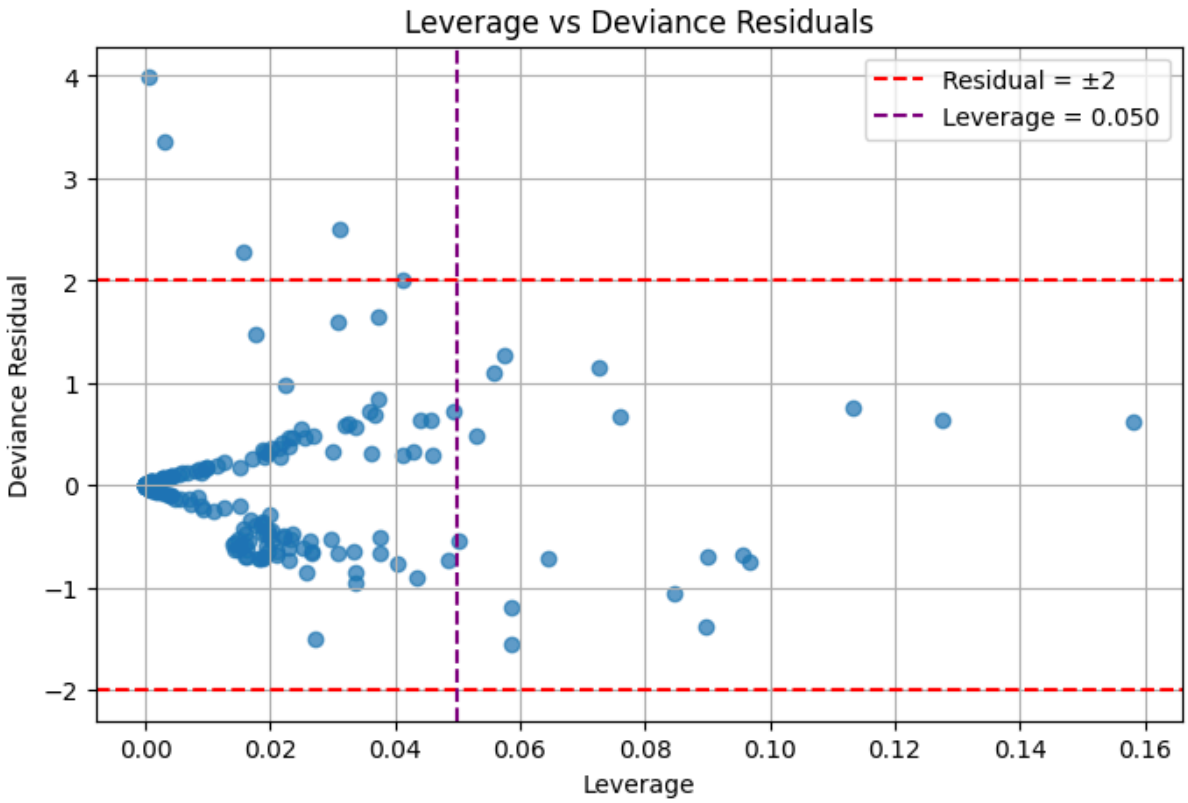
Yüksək leverage üçün hədd:

$$h_i > 0.1$$

In [ ]:



In [ ]:



📊 5.3. Hosmer–Lemeshow Testi (Yaxşı Uyğunluq Testi)

Modelin proqnozlaşdırdığı ehtimallarla həqiqi nəticələr nə qədər uyğundur?

Test statistikası:

$$C = \sum_{j=1}^{g} \frac{(O_j - E_j)^2}{E_j(1 - \hat{p}_j)}$$

Burada:

$$O_j$$

müşahidə olunan (observed) pozitiv hadisə sayı,

$$E_j$$

gözlənilən (expected) pozitiv hadisə sayı,

$$\hat{p}_j$$

qrup orta proqnoz ehtimalı,

$$g$$

qrup sayı (adətən 10).

Nəticə:

(p > 0.05) olarsa, model uyğun hesab olunur ✅

(p < 0.05) olarsa, uyğunluq kifayət deyil ❌

In [ ]:

```
Hosmer-Lemeshow Statistikası: 127.91694407839114
p-value: 7.642870493220003e-24
```

🔁 5.4. Çoxlu Xətti Asılılıq (Multicollinearity) – VIF ilə Nəzarət

Müstəqil dəyişənlər arasında yüksək korrelyasiya varsa, əmsallar qeyri-sabit olur və modelin şərh edilməsi pozulur.

VIF (Variance Inflation Factor) düsturu:

$$VIF_j = \frac{1}{1 - R_j^2}$$

Burada:

$$R_j^2$$

(j)-inci dəyişənin digər müstəqil dəyişənlərə reqressiyasından alınan R-kvadrat dəyəridir.

Şərh:

- (VIF < 5) → Problem yoxdur
- (5 < VIF < 10) → Diqqət edilməlidir
- (VIF > 10) → Çoxlu xətti asılılıq var

In [ ]:

```
  Feature        VIF
0   const  25.945055
1      x1   1.890110
2      x2   1.890110
```

🧪 Müqayisə

Nəzarət nöqtələri və mənaları:

Dəyərlər	İstifadə Olunan Ölçü/Düstur	Mənası / Şərhi
Deviance Residuals	$r_i = \text{sign}(y_i - \hat{p}_i) \cdot \sqrt{-2\left[y_i \log(\hat{p}_i) + (1 - y_i)\log(1 - \hat{p}_i)\right]}$	Modelin müşahidə əsaslı uyğunluğunu ölçür
Cook's Distance	$D_i = \frac{\sum(\hat{y} - \hat{y}_{(i)})^2}{p \cdot MSE}$	Müşahidənin modelə təsiri
Leverage	$h_i = x_i^T (X^T X)^{-1} x_i$	Müşahidənin outlierinin olub olmadığını göstərir
Hosmer-Lemeshow Testi	$C = \sum \frac{(O_j - E_j)^2}{E_j(1 - \hat{p}_j)}$	Model uyğunluğunun statistik testi
VIF	$VIF_j = \frac{1}{1 - R_j^2}$	Dəyişənlər arası çoxlu xətti asılılıq

📘 6.Feature engineering and Preprocessing

Maşın öyrənməsi modellərində verilənləri birbaşa istifadə etmək adətən kifayət olmur. Modelin uğurlu olması üçün **verilənlərin düzgün hazırlanması, transformasiyası və uyğun featurelərin yaradılması** lazım gəlir.

6.1 Dummy Dəyişənlər (One-Hot Encoding)

- **Nədir?** Kateqorik dəyişənlərin saylı modelə daxil edilməsi üçün ədədi(numeric) formaya salınması.
- Misal: "Rəng" dəyişəni: {Qırmızı, Mavi, Yaşıl}
- Saylı modelə birbaşa daxil edilə bilməyən kateqorik dəyişənləri, hər kateqoriya üçün ayrıca bir sütun (dummy dəyişən) yaradırıq.

Necə?

Hər kateqoriya üçün 0 və ya 1 dəyərləri alan yeni dəyişənlər yaradılır.

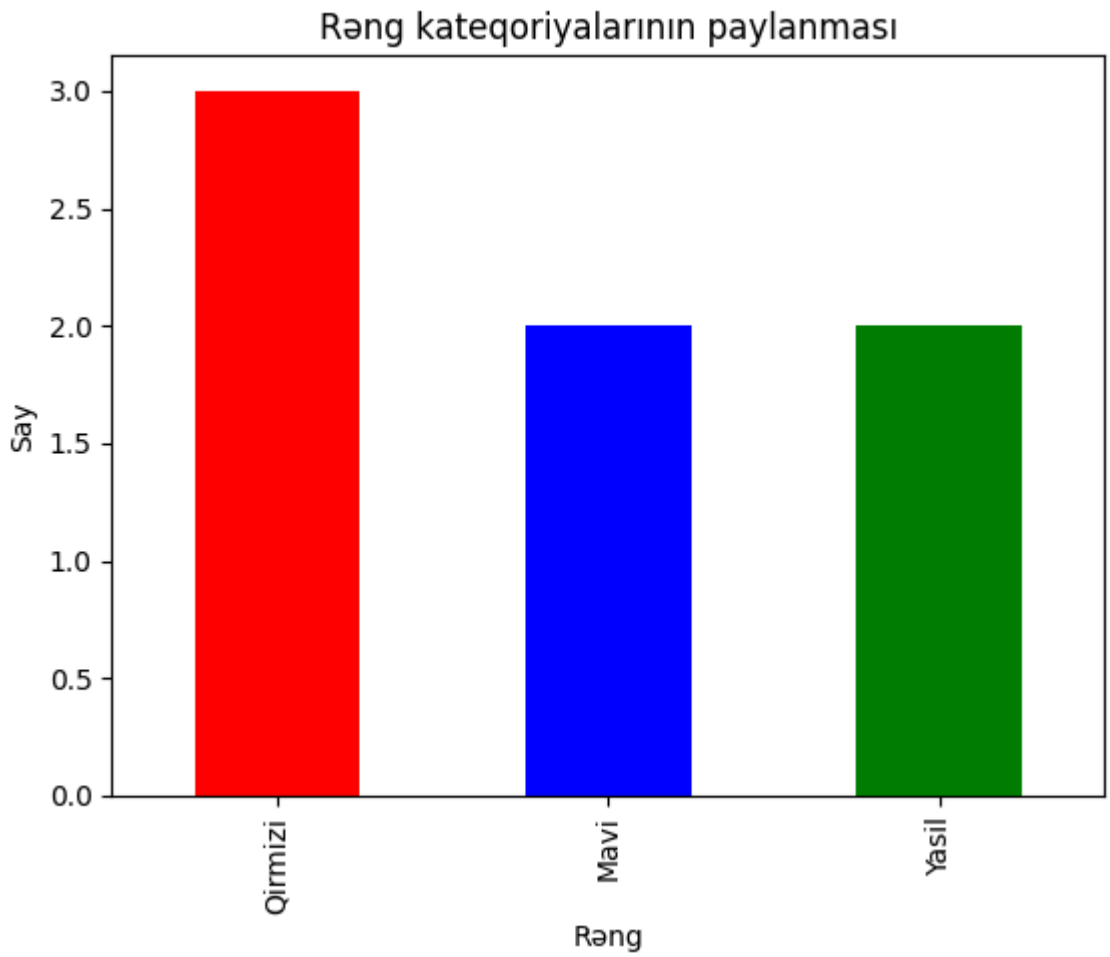
Məsələn:

Rəng	Rəng_Qırmızı	Rəng_Mavi	Rəng_Yaşıl
Qırmızı	1	0	0
Mavi	0	1	0
Yaşıl	0	0	1

Niyə önəmlidir?

- Modellər saylı dəyərlərlə işləyir, bu çevrilmə modeli qidalandırır.
- Kateqorik dəyişənlər düzgün təmsil olunmazsa, modelin performansı zəif olar.
- **Qeyd:** Dummy dəyişənlərdən biri adətən çıxarılır(drop_first=True) (dummy variable trap qarşısını almaq üçün) — digərləri o dəyişənin təmsilini təmin edir.

In [ ]:



6.2 Feature Scaling

- **Nədir?** Özəlliklər fərqli miqyaslardadırsa (məsələn, yaş 0-100, gəlir 1000-1000000), model xüsusilə məsafə əsaslı (məsələn, KNN, SVM) və ya qradient əsaslı alqoritmlərdə çətinlik çəkə bilər.
- **Məqsəd:** Bütün özəllikləri ortaq bir miqyasa gətirmək.

Ən çox istifadə edilən üsullar:

- **Min-Max Scaling:** Özəllik dəyərlərini 0 ilə 1 arasına sıxışdırır.

$$x_{scaled} = \frac{x - x_{min}}{x_{max} - x_{min}}$$

- **Standardlaşdırma (Z-Score):** Özəlliyin ortalaması 0, standart kənaraçıxması 1 olacaq şəkildə çevirir.

$$x_{scaled} = \frac{x - \mu}{\sigma}$$

Burada

$$\mu$$

: ortalama,

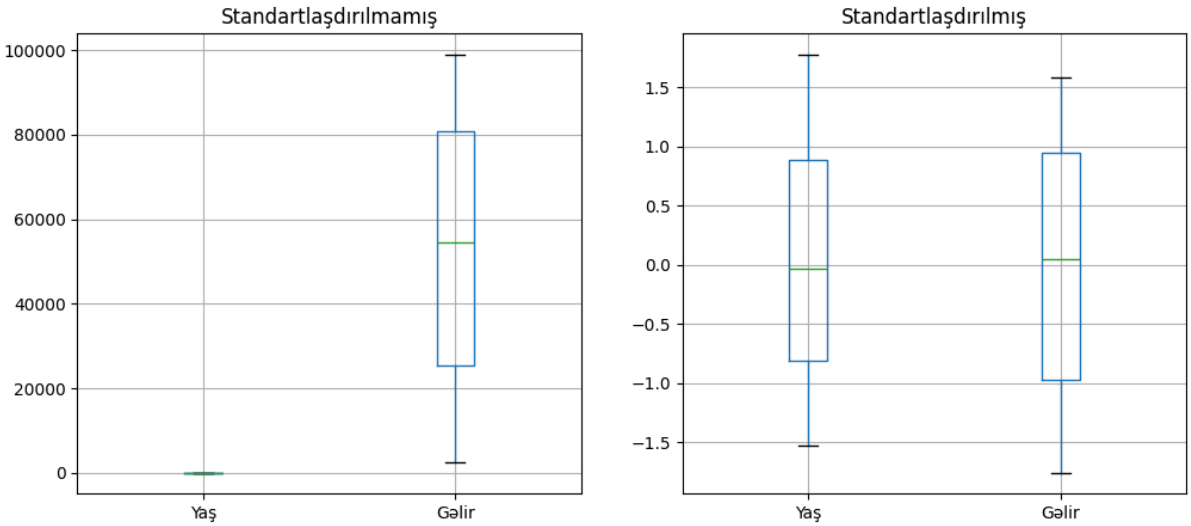
$$\sigma$$

: dispersiya(std).

Nə vaxt istifadə edilir?

- Məsafəyə əsaslanan modellərdə (KNN, SVM, k-means kimi)
- Xüsusilə fərqli vahidlərə malik özəlliklərdə.

In [ ]:



6.3 Outlierlər

- **Nədir?** Verilən setində əksəriyyətdən çox fərqli, kənar nöqtələrdə olan dəyərlər.
- Misal: Bir insanın yaşı 150-dirsə, bu, böyük ehtimalla outlierdir.

Niyə önəmlidir?

- Modeli yanlış yönləndirə bilər, xüsusilə ortalamaya əsaslanan üsullarda səhvləri böyüdür.
- Modelin ümumi uğuruna və ümumiləşdirməsinə mənfi təsir göstərir.

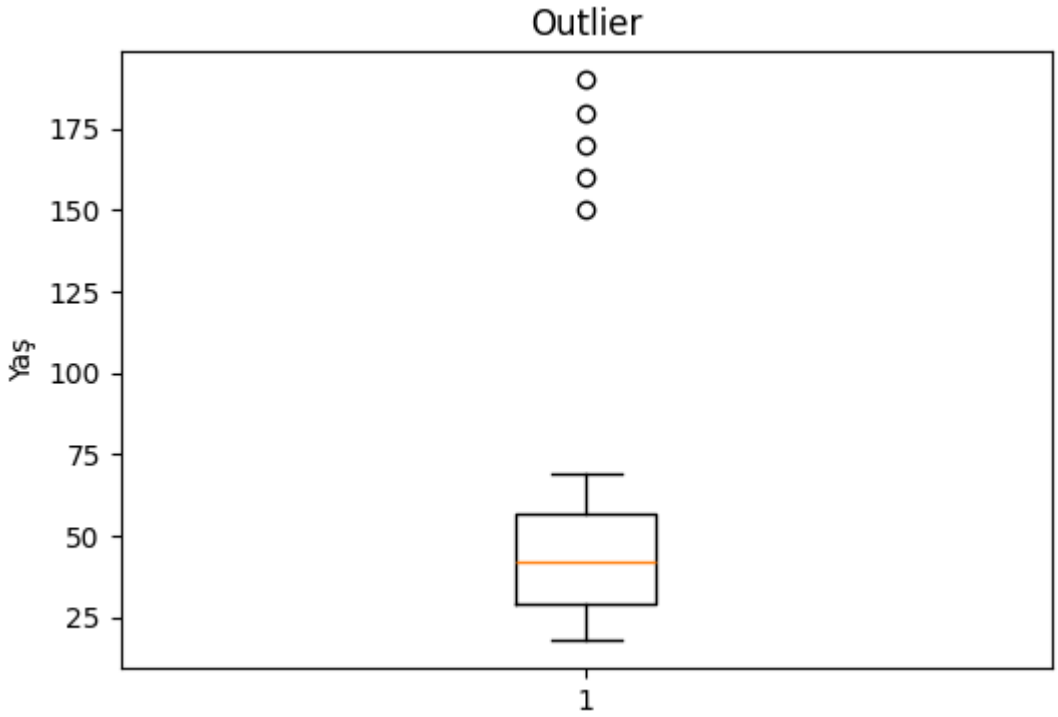
Outlierlərin aşkarlanması:

- **Statistik üsullar:** Z-score, IQR (Kvantillər) üsulu
- **Qrafik üsullar:** Boxplot, scatter plot

Outlierlərə tətbiq olunanlar:

- Silmək (əgər səhvdirsə)
- Transformasiya (loqarifm, kök kimi)
- Winsorizing

In [ ]:



6.4 Interaction Terms

- **Nədir?** İki və ya daha çox dəyişənin təsirlərinin birləşməsi.
- Misal: Yaş və gəlir tək başına təsirli ola bilər, lakin yaş $\times$ gəlir çarpımı model üçün əhəmiyyətli ola bilər.

Düstur:

İki dəyişən

$$x_1$$

və

$$x_2$$

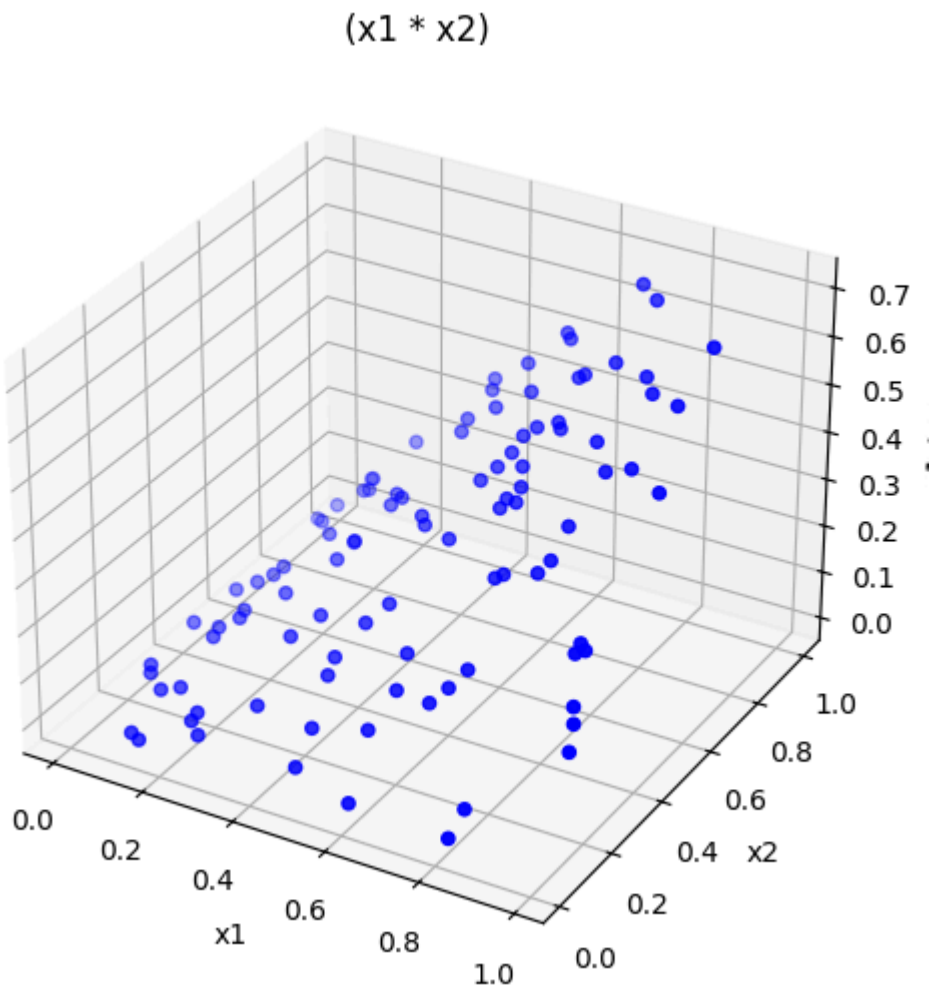
üçün qarşılıqlı təsir şərti:

$$x_3 = x_1 \times x_2$$

Niyə?

- Özəlliklər arasındakı mürəkkəb əlaqələri modelə əlavə edir.
- Modelin dəqiqliyini artıra bilər.

In [ ]:



6.5 Polynomial Features

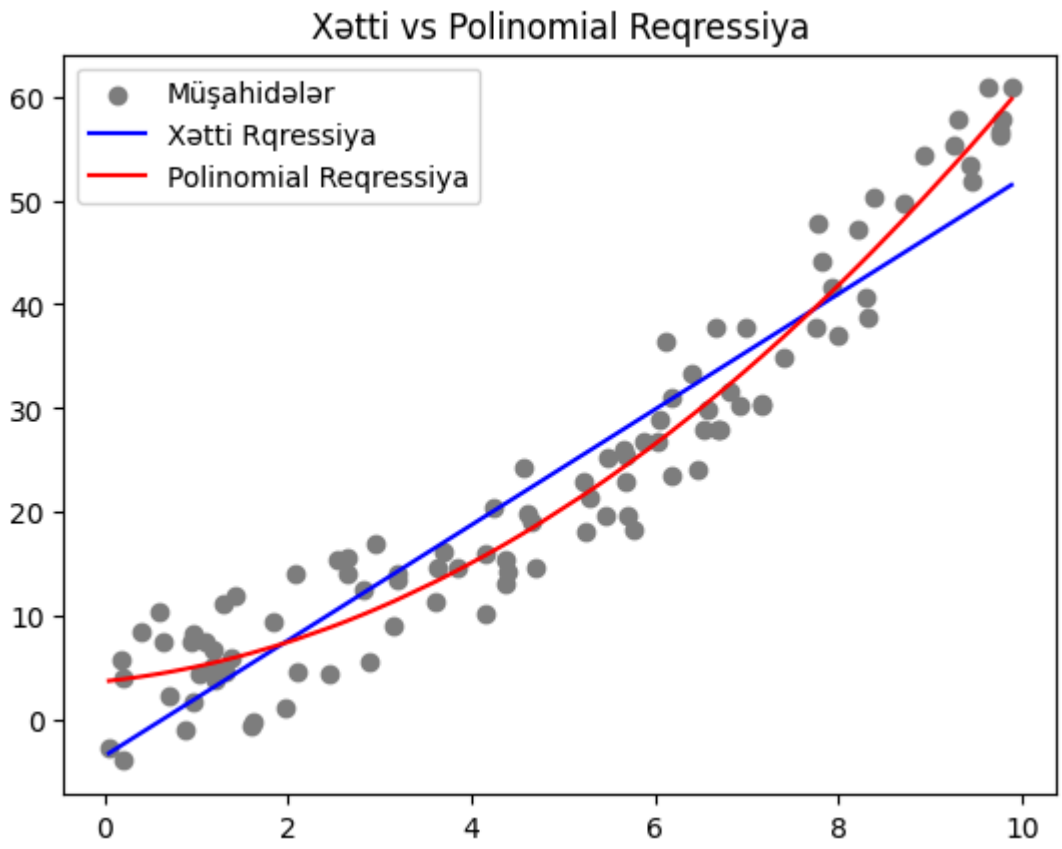
- **Nədir?** Mövcud özəlliklərin kvadratı, kubu kimi yüksək dərəcəli şərtlərini yaratmaq.
- Misal:

$$x_1^2, x_1^3, x_1 x_2$$

Məqsəd:

- Modelin qeyri-xətti əlaqələri öyrənməsini təmin edir.
- Xətti modellərdə elastikliyi artırır.

In [ ]:



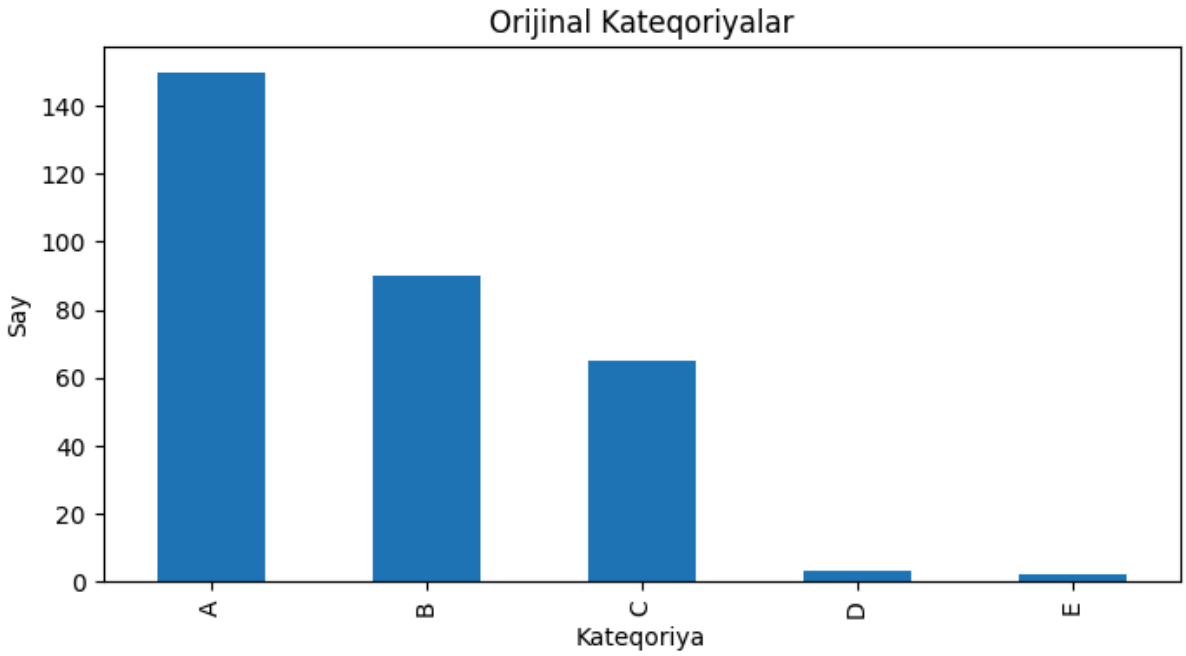
6.6 Rare Label Encoding

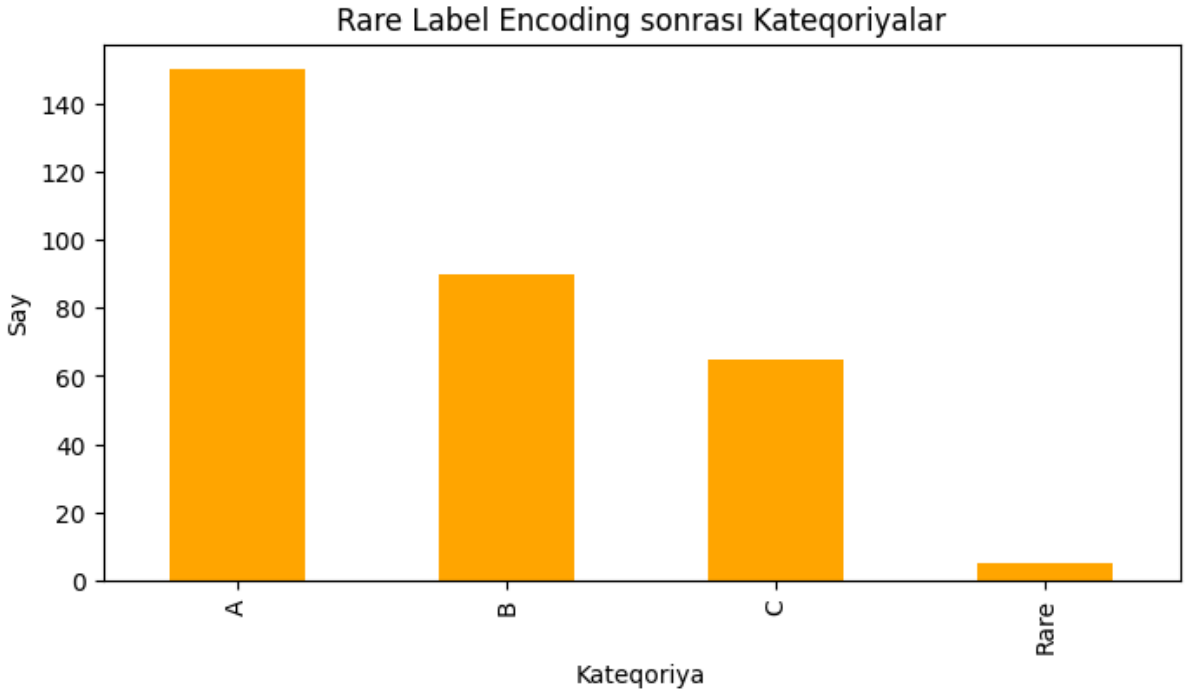
- **Nədir?** Kateqorik dəyişənlərdə çox az rast gəlinən siniflər (məsələn, %1-dən az verilənlən təşkil olunmuş) nadir kateqoriya adlandırılır.
- Bu nadir kateqoriyalar modelə daxil edildikdə problem yarada bilər (overfitting, qeyri-sabitlik).

Həll Üsulları:

- Nadir kateqoriyaları tək bir "Rare" etiketi altında birləşdirmək.
- Bu şəkildə model daha stabil və ümumiləşdirmə gücü daha yüksək olar.

In [ ]:





📘 7. Regularizasiya Metodları

Maşın öyrənməsi və statistikada, xüsusilə reqressiyada, modelin mürəkkəbliyi artdıqca **overfitting** riski yüksəlir. **Regularizasiya** metodları, modeli sadələşdirib ümumiləşdirmə qabiliyyətini artırmaq üçün istifadə olunur.

7.1 L1 Regularizasiya (Lasso)

L1 Requlrarizasiya, bəzi əmsalları tam olaraq sıfıra endirərək dəyişən seçimi edir.

$$L_{lasso}(\beta) = L(\beta) + \lambda \sum_{j=1}^{p} |\beta_j|$$

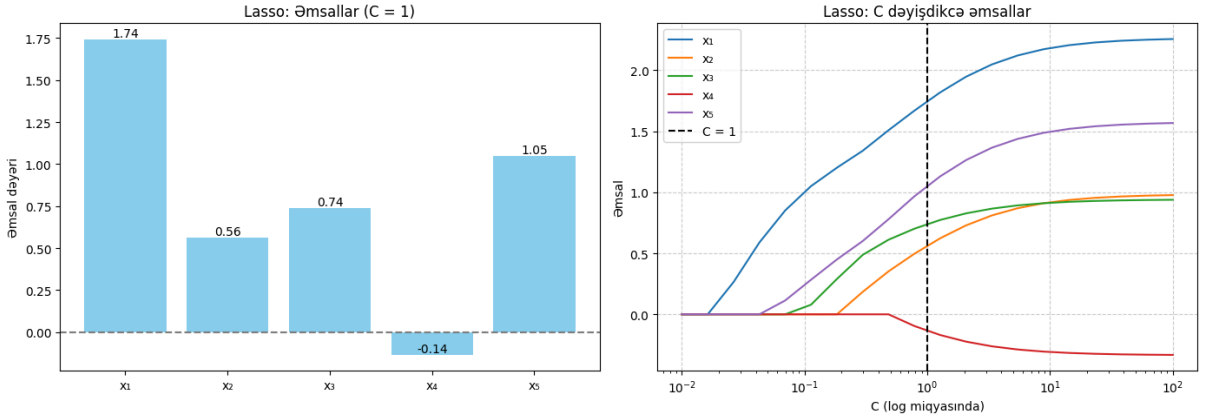
- $L(\beta)$

: əsas loss funksiyası (məsələn, loqistik reqressiyada neqativ log-likelihood)

- λ
  : tənzimləmə gücü (hiperparametr)
- β_j
  : j-inci əmsal

Nəticə: Bəzi əmsallar sıfırlanır, beləliklə modeldə əhəmiyyətli dəyişənlər seçilmiş olur.

In [ ]:



7.2 L2 Regularizasiya (Ridge)

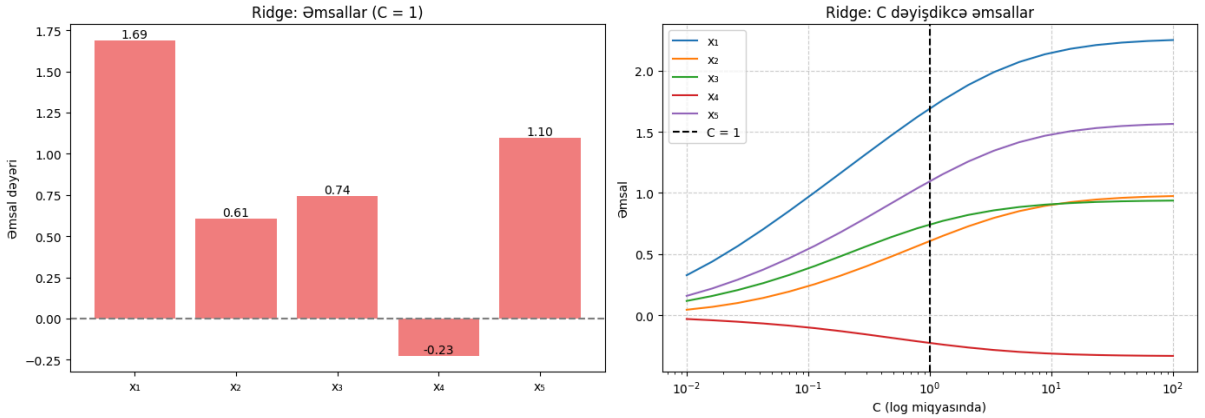
L2 tənzimləməsi, əmsalların ölçüsünü kiçildir, lakin sıfıra endirmir.

$$L_{ridge}(\beta) = L(\beta) + \lambda \sum_{j=1}^{p} \beta_j^2$$

- $L(\beta)$
  : əsas itki funksiyası
- λ
  : tənzimləmə gücü
- β_j
  : j-inci əmsal

Nəticə: Əmsallar kiçilir, model daha sabit olur.

In [ ]:



7.3 ElasticNet – Lasso + Ridge

L1 və L2 tənzimləmələrinin birləşməsidir.

$$L_{elasticnet}(\beta) = L(\beta) + \lambda \left[\alpha \sum_{j=1}^{p} |\beta_j| + (1-\alpha) \sum_{j=1}^{p} \beta_j^2 \right]$$

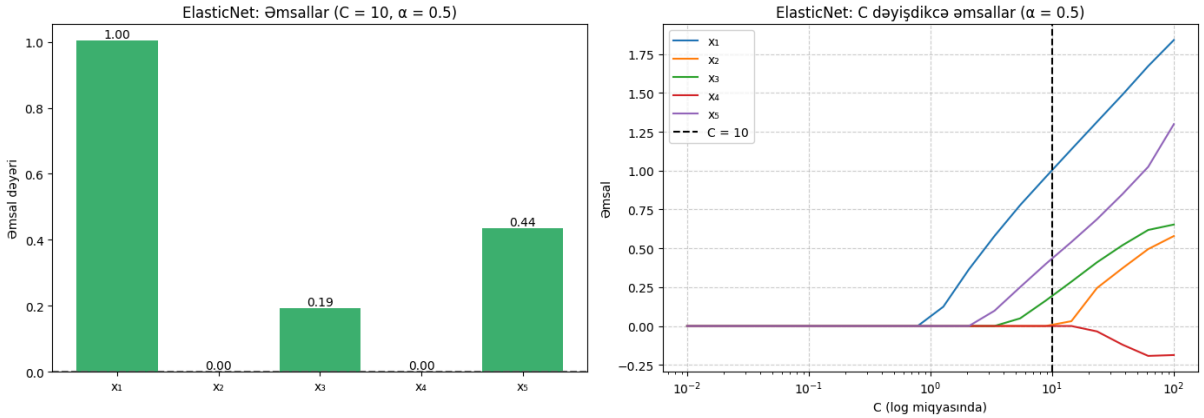
- $\alpha \in [0, 1]$
- $\alpha = 1$

  → tam Lasso
- $\alpha = 0$

  → tam Ridge

Nəticə: Həm dəyişən seçimi, həm də əmsal kiçiltməsi edilir.

In [ ]:



7.4 C Parametri (Regularizasiyanın Tərsi)

Loqistik reqressiyada tənzimləmə gücünü idarə edir.

$$C = \frac{1}{\lambda}$$

- Kiçik

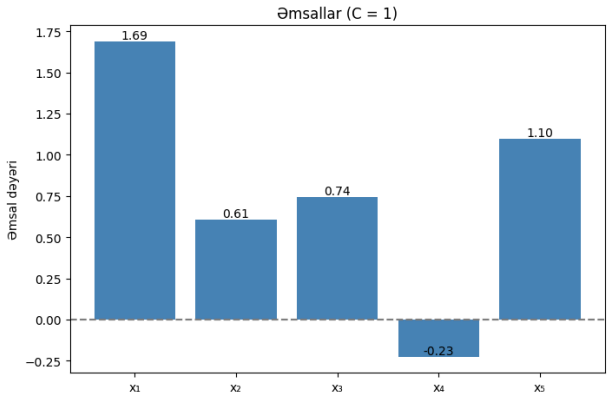
  C

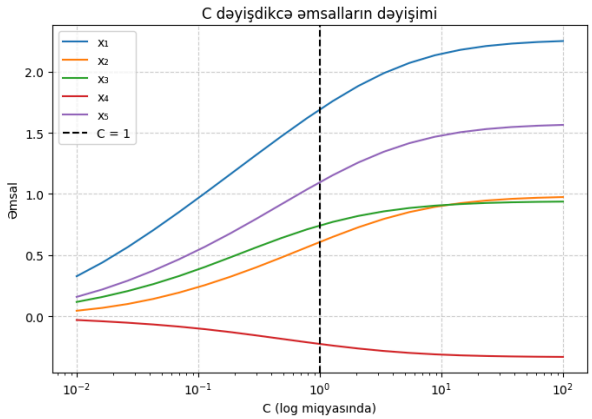
  → güclü tənzimləmə (daha çox kiçiltmə)
- Böyük

  C

  → zəif tənzimləmə (daha az kiçiltmə)

In [ ]:





📘 8. Tətbiq (Python)

8.1 Scikit-learn ilə Loqistik Reqressiya ⚙️

Təsəvvür edin ki, əlinizdə bir e-ticarət saytı var və müştərilərin bir məhsulu alıb-almayacağını təxmin etmək istəyirik. Əlimizdəki verilənlərdə müştərilərin yaşı, alış-veriş keçmişi, saytda keçirdiyi vaxt kimi məlumatlar var. Məhz burada **Scikit-learn** köməyə gəlir.

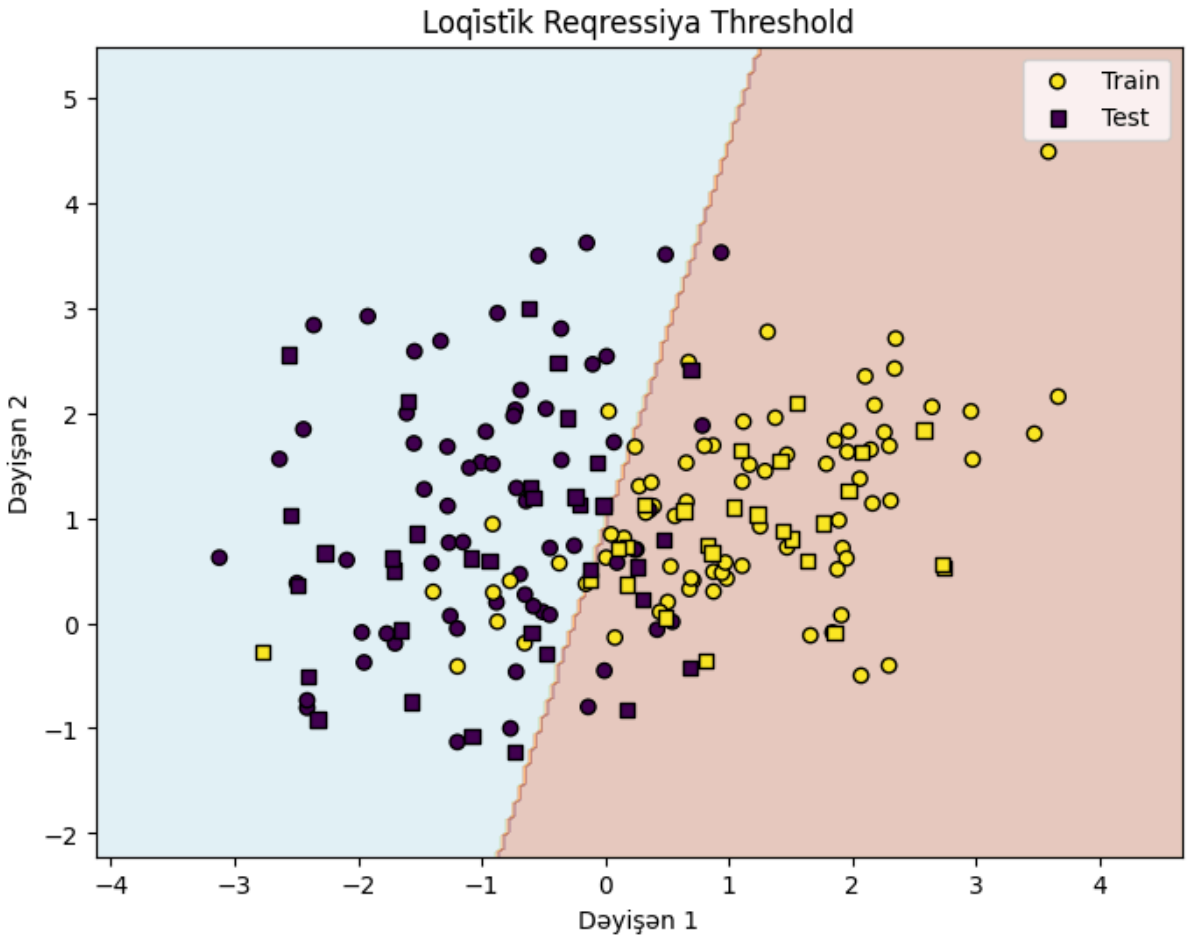
Scikit-learn ilə əlimizdəki verilənlər üzərindən bir model qurub bu müştərinin məhsulu alıb-almayacağını təxmin edə bilərik. Məsələn, belə düşünün:

- Müştəri A: 30 yaşında, saytda 5 dəqiqə keçirib, əvvəlki alış-verişləri var.
- Modelimiz bu məlumatları istifadə edərək, məhsulu alma ehtimalını hesablayır və “Bəli, alacaq” və ya “Xeyr, almayacaq” deyə qərar verir.

Bu prosesdə model öyrədilərkən `fit` funksiyasından istifadə edilir, beləliklə model verilənləki örnəkləri öyrənir. Sonra `predict` ilə yeni müştərilər üçün təxminlər edirik.

Scikit-learn, təxminləri asanlıqla verməklə yanaşı, eyni zamanda modeli qiymətləndirmək üçün dəqiqlik (accuracy), kəskinlik (precision), həssaslıq (recall) kimi göstəricilər də təqdim edir. Beləliklə modelin nə qədər yaxşı işlədiyini görə bilərik.

In [ ]:



8.2 Statsmodels ilə Loqistik Reqressiya 📊

Daha dərindən və detallı statistikaları görmək istədiyimiz vaxt, **Statsmodels** işinizə yarayır.

Deyək ki, məhsul satın alma ehtimalını sadəcə təxmin etmək deyil, hansı xüsusiyyətlərin bu qərara nə qədər təsir etdiyini də anlamaq istəyirsiniz. Statsmodels bunu təmin edir.

Məsələn, yaşın modeldəki təsiri nə qədərdir? Sayta ziyarət müddəti həqiqətən vacibdirmi? Əmsalların əhəmiyyətliliyi (p-dəyərləri) burada qarşımıza çıxır.

Statsmodelsin köməyi ilə modelinizi `fit` etdikdən sonra detallı bir report hazırlamaq mümkündür:

- Hər bir dəyişənin əmsalı (məsələn yaşın əmsalı 0.05 ola bilər),
- Bu əmsalın standart xətası,
- P-dəyəri: Bu əmsalın həqiqətən təsirli olub-olmadığını göstərir (kiçikdirsə təsirli, böyükdirsə təsirsiz).

Bu sayədə sadəcə təxmin etməklə kifayətlənmirik, modelin daxili məntiqini də başa düşmək mümkündür.

In [ ]:

```
Confusion Matrix:
[[53  2]
 [ 2 43]]

Classification Report:
              precision    recall  f1-score   support

           0       0.96      0.96      0.96        55
           1       0.96      0.96      0.96        45

    accuracy                           0.96       100
   macro avg       0.96      0.96      0.96       100
weighted avg       0.96      0.96      0.96       100
```

8.3 Model Öyrədilməsi, Təxmin və Qiymətləndirmə 🎯

Bir digər vacib addım modelimizi **öyrətmək** və sonra **yoxlamaq**dır.

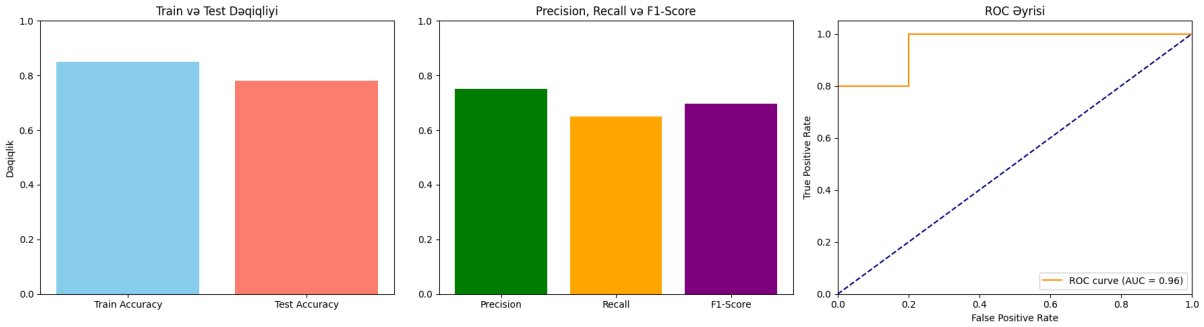
Deyək ki, əlimizdə 1000 müştəri verilənləri var. Bu verilənin 800-ünü modeli öyrətmək üçün istifadə edirik, qalan 200-ünü isə modeli test etmək üçün ayırırıq. Model test verilənindəki müştərilərin satın alma vəziyyətini təxmin edir.

Burada baxacağımız bir neçə əsas məqam var:

- **Dəqiqlik (Accuracy):** Modelin doğru təxmin etdiyi nümunələrin nisbəti. Məsələn, 200 test müştərisindən 160-nı doğru təxmin edərsə, dəqiqlik %80 olar.
- **Kəskinlik (Precision) və Həssaslıq (Recall):** Məsələn, model satın alacaq dediyində nə qədər doğrudur, satın alacaqları nə qədər qaçırmır deyə baxırıq.
- **F1-Skor (F1-Score):** Kəskinlik (Precision) ilə Həssaslığın (Recall) balanslı ortalaması, xüsusilə balanssız verilən setlərində vacibdir.
- **ROC AUC:** Modelin pozitiv və neqativləri ayırd etmə gücü.

Burada vacib məqam, yüksək dəqiqliyin hər zaman yaxşı model demək olmamasıdır. Məsələn, satın alan müştərilər azdırsa (məsələn %5), model hər kəsin satın almadığını təxmin edərək belə %95 dəqiqlik əldə edə bilər! Buna görə də digər göstəricilərə baxmaq şərtdir.

In [ ]:



8.4 ROC və Qarışıqlıq Matrisi Vizualizasiyaları 📈

Modeli sadəcə rəqəmlərlə deyil, vizual olaraq da analiz etmək çox faydalıdır.

Məsələn:

- **Qarışıqlıq Matrisi (Confusion Matrix):**\

Burada 4 xana var:

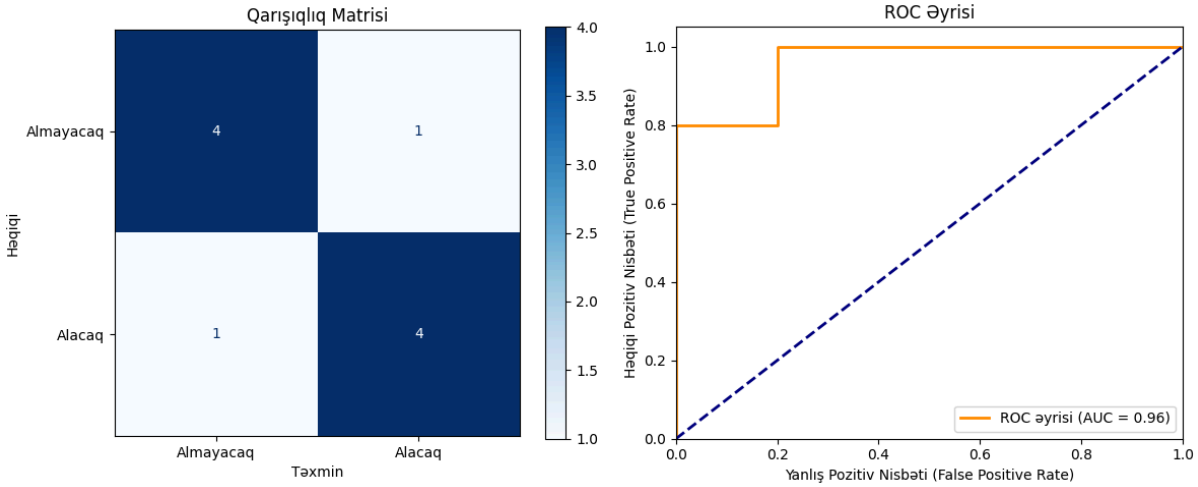
- **Həqiqi Pozitiv (True Positive):** Modelin “satın alacaq” dediyi və həqiqətən alan müştəri sayı.
- **Yanlış Pozitiv (False Positive):** Model “satın alacaq” dediyi amma almayanlar.
- **Həqiqi Neqativ (True Negative):** Model “almayacaq” dediyi və almayanlar.
- **Yanlış Neqativ (False Negative):** Model “almayacaq” dediyi amma alanlar.

Bu matrislə modelin hansı növ səhvləri daha çox etdiyini görə bilərsiniz.

- **ROC Əyrisi:**\

Modelin müxtəlif qərar həddləri üçün (məsələn satın alma ehtimalı 0.5 əvəzinə 0.3 kimi) necə performans göstərdiyini göstərir. Əyri altındakı sahə (AUC) nə qədər böyükdürsə, model o qədər yaxşıdır.

In [ ]:



8.5 GridSearchCV ilə Model Optimizasiyası 🔍

Son olaraq, modelinizi daha yaxşı hala gətirmək üçün hiperparametr tənzimləmələri etmək lazım gələ bilər.

Loqistik reqressiyada məsələn **C parametri** (requlərizasiya gücü) modelin nə qədər çevik və ya sərt olacağını müəyyən edir.

- Kiçik C → model verilənə çox uyğunlaşır, həddən artıq öyrənmə (overfitting) riski var.
- Böyük C → model həddən artıq məhdudlaşdırılır, az öyrənmə (underfitting) baş verir.

GridSearchCV ilə müxtəlif C dəyərlərini və digər parametrləri avtomatik olaraq sınaqdan keçirib, ən yaxşı nəticə verən tənzimləməni tapa bilərsiniz. Həmçinin bu proses zamanı veriləni k-fold çarpaz doğrulama (cross-validation) ilə bir neçə hissəyə bölüb test edərək, modelin ümumi uğurunu etibarlı şəkildə ölçür.

In [ ]:

```
Ən yaxşı C dəyəri: 0.01
Ən yaxşı CV dəqiqliyi: 0.8612500000000001
Test dəqiqliyi: 0.865
```

📘 9. Tez-tez Edilən Səhvlər

9.1 Sinif Balanssızlığının (Class Imbalance) Göz Ardı Edilməsi ⚖️

Nədir?

Siniflər arasında çox böyük fərqlər olduqda, məsələn, bir sinif çoxlu sayda nümunəyə sahibkən digər sinif çox az nümunəyə sahibdirsə, model adətən çoxluq sinfinə fokuslanır.

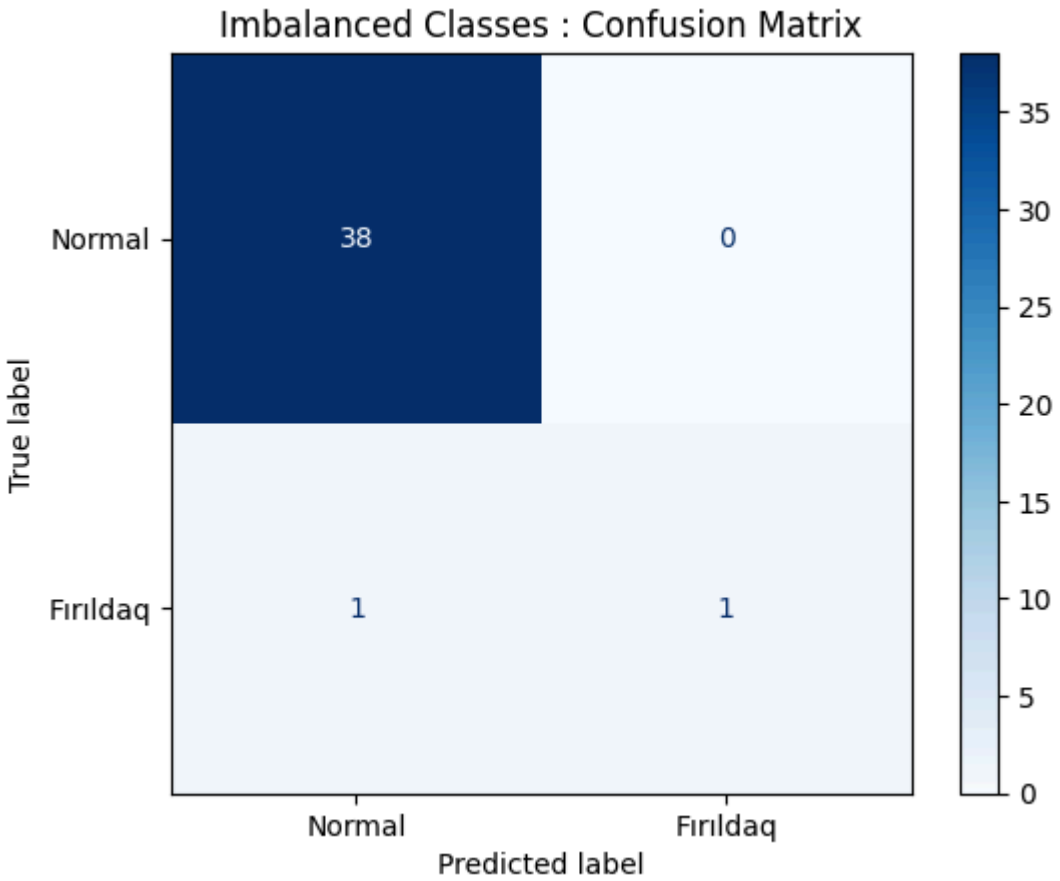
Nümunə:

Bir fırıldaqçılıq aşkarlama modelində fırıldaqçılıq halları %1, normal əməliyyatlar %99 isə model əksər vaxt "bu əməliyyat fırıldaqçılıq deyil" deyə təxmin edərək %99 dəqiqlik əldə edir. Amma əslində fırıldaqçılıq hallarını aşkar etmək əsas məqsəddir.

Niyə problemdir?

Yüksək dəqiqlik yanıldıcı olur, çünki nümunəyə az olan sinifindəki vacib hallar qaçırılır. Buna görə də **precision, recall və F1-score** kimi göstəricilərə baxmaq lazımdır.

In [ ]:



🔎 Classification Report:

	precision	recall	f1-score	support
0	0.974	1.000	0.987	38
1	1.000	0.500	0.667	2
accuracy			0.975	40
macro avg	0.987	0.750	0.827	40
weighted avg	0.976	0.975	0.971	40

9.2 Xətti Olmayan Əlaqələrdə Modelin Çətinlik Çəkməsi

Nədir?

Loqistik reqressiya xətti bir modeldir; müstəqil dəyişənlərin hədəflə xətti bir əlaqəsi olduğunu fərz edir.

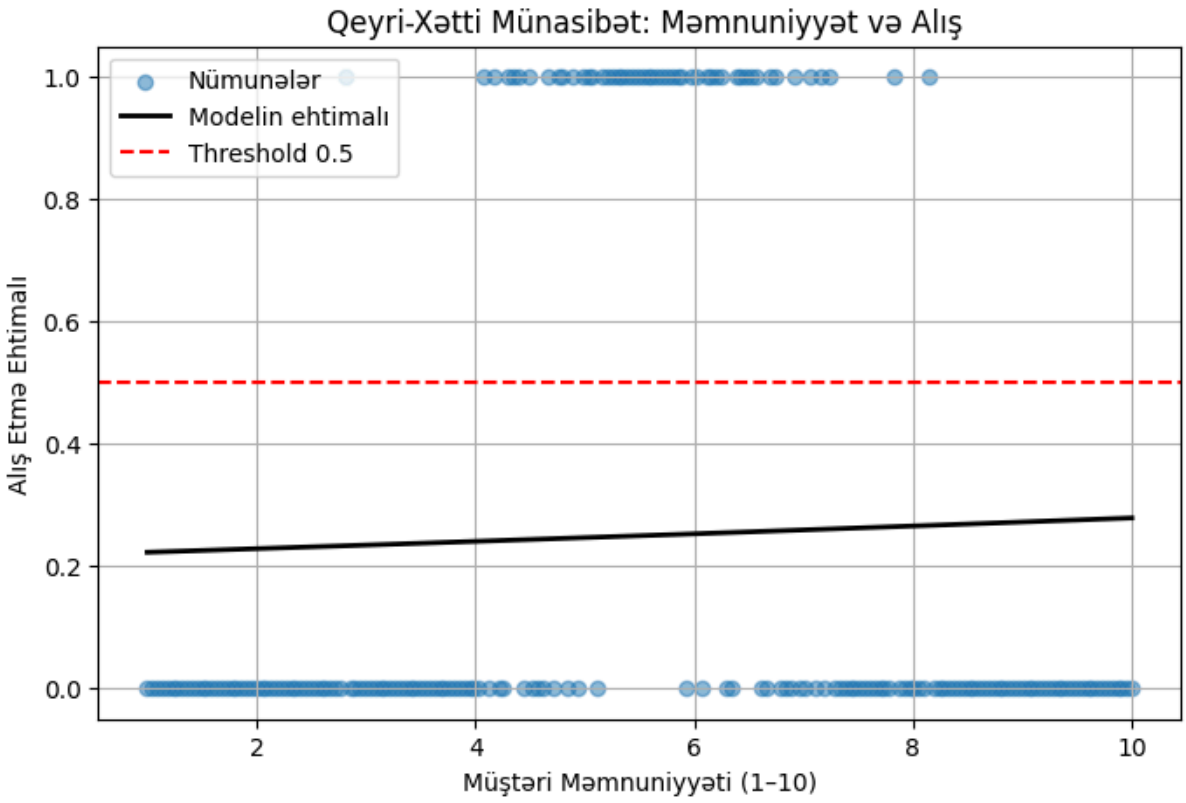
Nümunə:

Bir müştəri məmnuniyyəti balı ilə satın alma ehtimalı arasında xətti olmayan (məsələn U şəkilli) əlaqə varsa, loqistik reqressiya bu əlaqəni tam anlaya bilməz.

Niyə problemdir?

Model səhv təxminlər edir, çünki əlaqə xətti deyil. Bu halda **polinom terminlər əlavə etmək**, **transformasiyalar aparmaq** ya da **başqa modellər** (məsələn decision tree) istifadə etmək daha yaxşı olar.

In [ ]:



9.3 Dəyişən Tələsi (Dummy Variable Trap) 🚫

Nədir?

Kateqorik dəyişənləri modelə daxil edərkən, bütün kateqoriyaları dəyişən kimi istifadə etmək müstəqil dəyişənlər arasında mükəmməl çoxlu xətti asılılıq (multicollinearity) yaradır.

Nümunə:

Üç fərqli şəhər üçün "Bakı", "Gəncə və "Sumqayıt" adlı dəyişənlər yaradıb hamısını modelə əlavə etsəniz, biri digərlərinin xətti kombinasiyası olar.

Niyə problemdir?

Bu vəziyyət reqressiya əmsallarının mənasını poza bilər və modelin stabil işləməsini əngəlləyər. Həll olaraq bir kateqoriyanı əsas götürüb onu dəyişənlərdən çıxarmaq lazımdır.

In [ ]:

```
Model score: 0.90
          Feature  Coefficient
0             Yas    -0.371560
1           Maas     0.001732
2     Seher_Gəncə    -0.975841
3  Seher_Sumqayıt     0.872549
```

9.4 Həddən Artıq Öyrənmə (Overfitting) — Xüsusilə

Kiçik Dataset ilə ⚠️

Nədir?

Model `train` verilənina həddən artıq uyğunlaşır və verilən daxilindəki səs-küyü(noise) öyrənir. Nəticədə yeni verilənlərə ümumiləşdirmə edə bilmir.

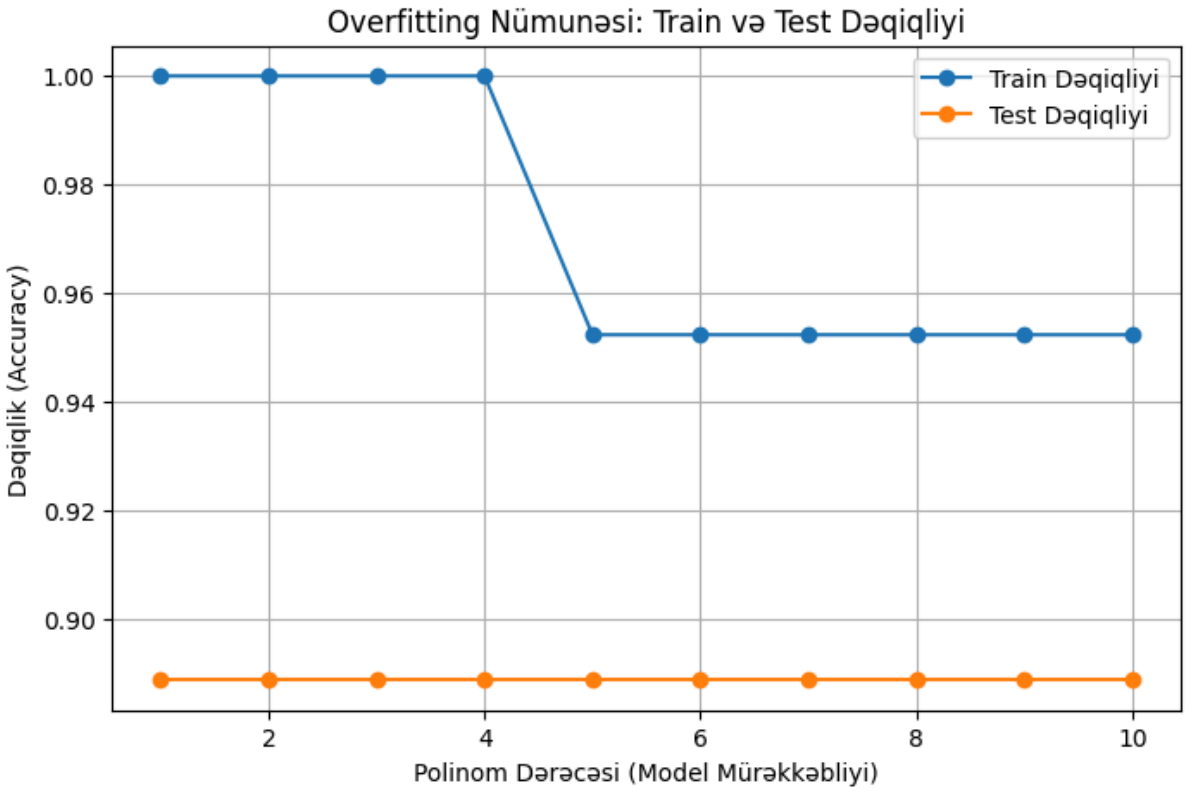
Nümunə:

Yalnız 50 nümunədən ibarət kiçik bir verilənlə çox mürəkkəb bir model təlim edilsə, `train` dəqiqliyi çox yüksək olar, amma `test` də performansı düşər.

Niyə problemdir?

Real həyatda model yaxşı işləməz, sadəcə `train` setini "əzbərləmiş" olar. Bu səbəbdən daha sadə modellər seçmək, verilən miqdarını artırmaq və ya **requlerizasiya** tətbiq etmək vacibdir.

In [ ]:



📘 10. Real Həyatda İstifadə Sahələri

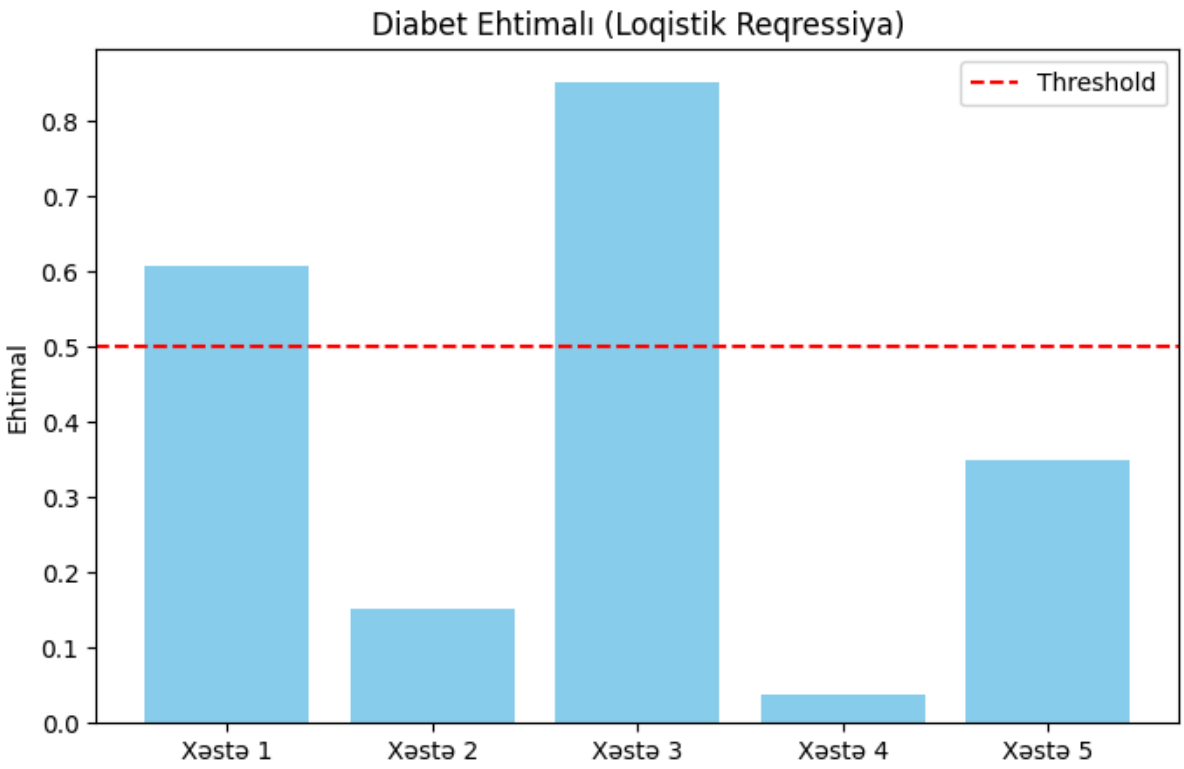
10.1 Tibb: Xəstəlik Proqnozu 🩺

Tibb sahəsində, xəstələrin müəyyən bir xəstəliyə sahib olub-olmadığını və ya xəstəliyin inkişaf edib-etməyəcəyini proqnozlaşdırmaq çox vacibdir.

- **Məqsəd:** Xəstənin simptomları, test nəticələri və demoqrafik məlumatları (yaş, cins və s.) istifadə edərək xəstəlik ehtimalını hesablamaq.

- **Nümunə:** Diabet riski proqnozu üçün; xəstənin çəki, yaş, qan şəkəri kimi məlumatları modelə daxil edilir və model “diabet var” ya da “yox” nəticəsini verir.
- **Niyə loqistik reqressiya?** Çünki nəticə iki siniflidir (xəstə/xəstə deyil) və ehtimal proqnozuna əsaslanır.
- Bu proqnozlar, erkən diaqnoz, qabaqlayıcı tədbirlər və müalicə planlaması üçün istifadə olunur.

In [ ]:

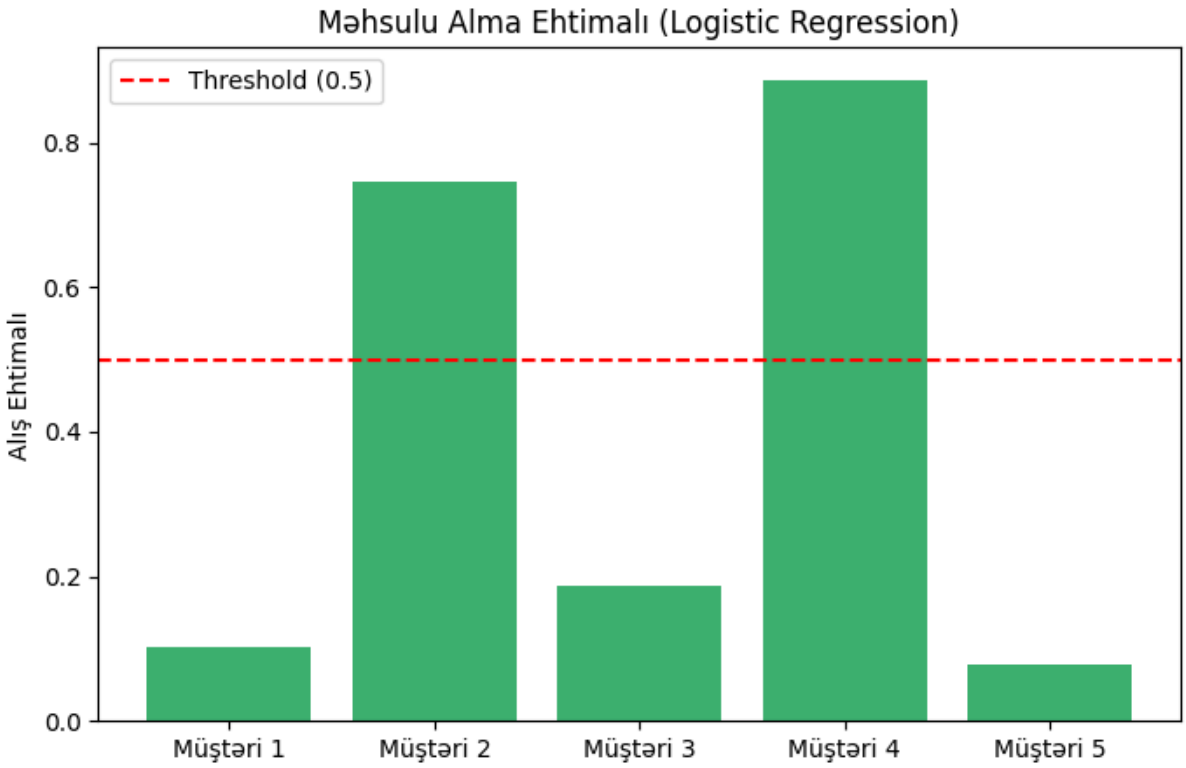


10.2 Marketinq

Marketinq sektorunda müştərilərin bir məhsulu alma ehtimalını proqnozlaşdırmaq vacib bir tapşırıqdır.

- **Məqsəd:** Müştərilərin demoqrafik məlumatları, əvvəlki satınalma davranışları və saytdakı fəaliyyətlərinə əsasən satınalma edib-etməyəcəyini proqnozlaşdırmaq.
- **Nümunə:** Bir e-ticarət saytında müştərinin yaşı, cinsi, əvvəlki alışları və veb saytı üzərindəki qarşılıqlı əlaqələri modelə daxil edilir; model “satın alacaq” ya da “satın almayacaq” nəticəsini verir.
- Bu proqnozlar kampaniya planlaması, hədəfli reklamlar və müştəri seqmentasiyası üçün istifadə olunur.

In [ ]:

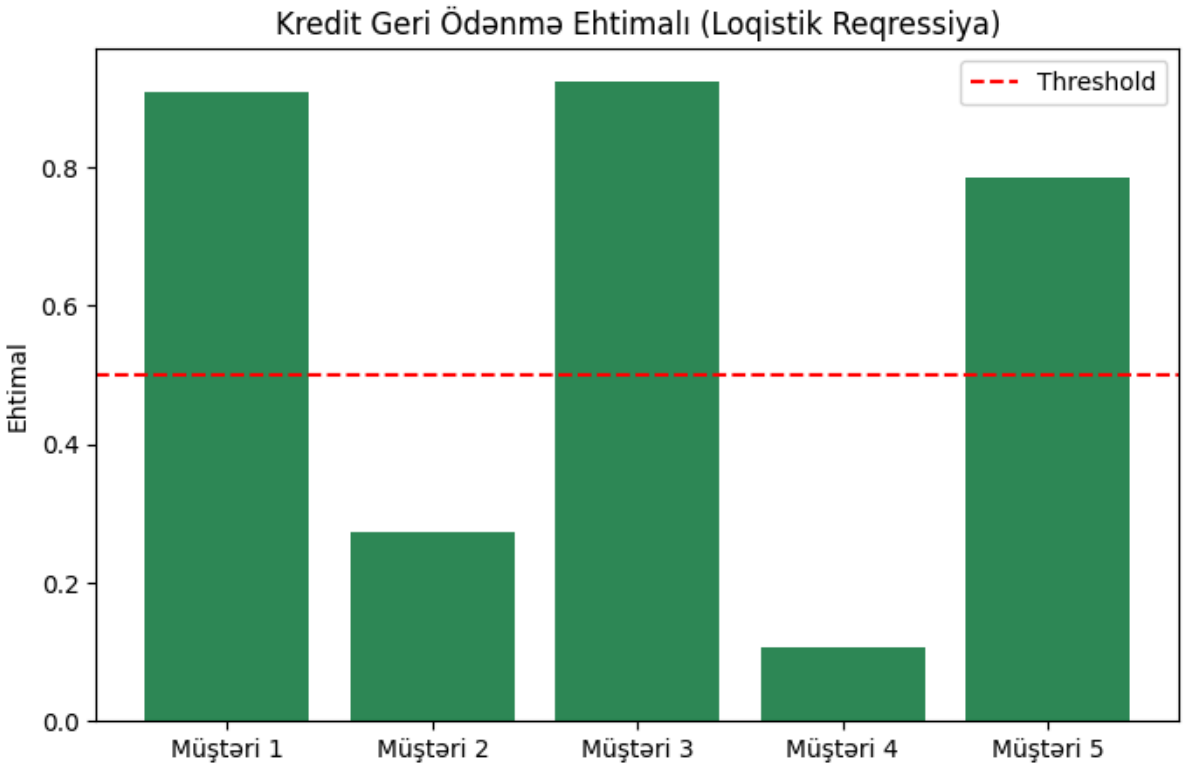


10.3 Maliyyə: Kreditin Geri Ödənilməsi Riski 💰

Maliyyə sektorunda, xüsusilə kredit verən qurumlar üçün, müştərilərin kreditlərini geri ödəmə ehtimalını proqnozlaşdırmaq kritik bir elementdir.

- **Məqsəd:** Kredit üçün müraciət edən şəxsin gəlir vəziyyəti, kredit keçmişi, mövcud borc vəziyyəti kimi məlumatlar istifadə edilərək kreditin geri ödənilməməsi riski proqnozu.
- **Nümunə:** Model bu məlumatlara əsaslanaraq “geri ödəyəcək” və ya “ödəyə bilməyəcək” şəklində nəticə verir.
- Bu məlumat bankların risk idarəçiliyini yaxşılaşdırır.

In [ ]:

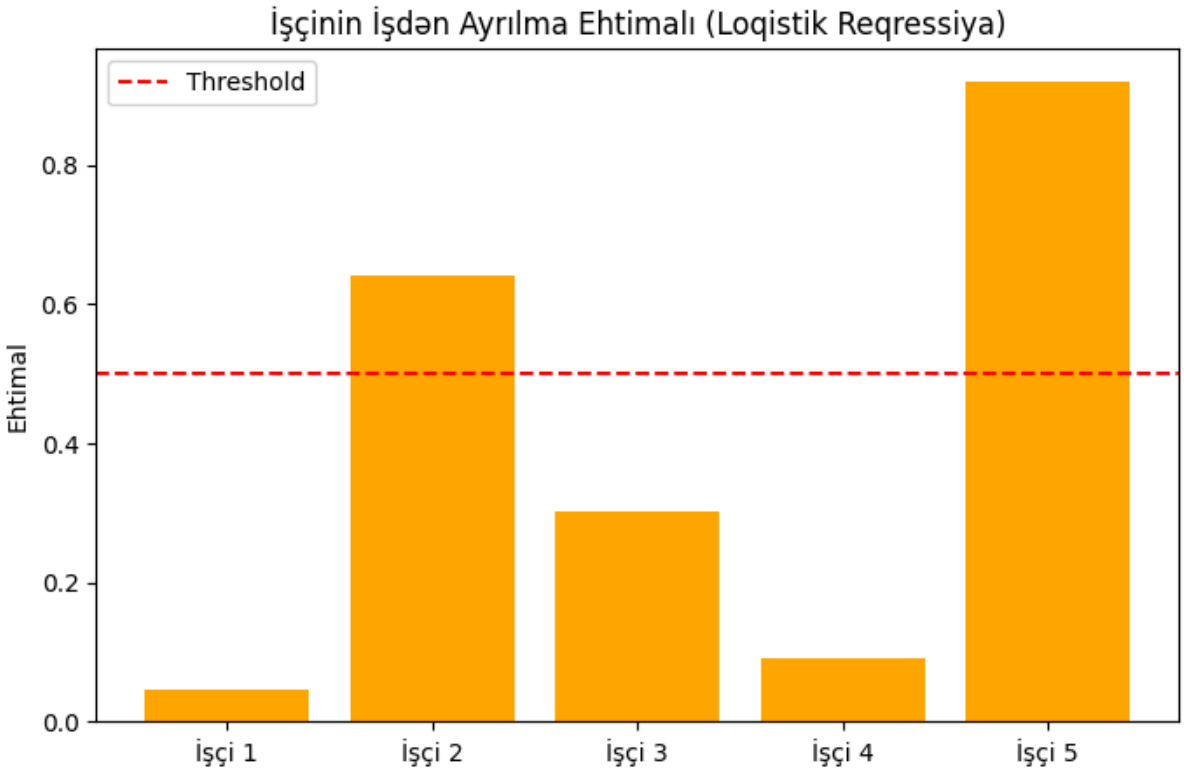


10.4 İnsan Resursları: İşçinin Ayrılma Proqnozu 👥

İnsan resursları sahəsində, işçilərin işdən ayrılma (turnover) riskini proqnozlaşdırmaq şirkətlər üçün böyük əhəmiyyət daşıyır.

- **Məqsəd:** İşçilərin işdən ayrılıb-ayrılmayacağını proqnozlaşdırmaq; beləliklə proaktiv tədbirlər görülə bilər.
- **Nümunə:** İşçinin yaşı, maaşı, iş müddəti, məmnuniyyət sorğuları kimi məlumatlar modelə daxil edilir və model “qalacaq” ya da “ayrılacaq” proqnozu verir.
- Bu sayədə işçi qüvvəsi planlaması və işçinin bağlılığı artırıla bilər.

In [ ]:



Xülasə 📌

Loqistik reqressiya kimi təsnifat modelləri, gerçək həyatda:

- Xəstəlik diaqnozunda 🩺,
- Müştərinin satınalma davranışında 🛒,
- Maliyyə risk qiymətləndirməsində 💰,
- Kadr idarəçiliyində 👥
- və.s sahələrdə istifadə olunur.

Bu sahələrdə modelin əsas vəzifəsi, bir hadisənin baş vermə ehtimalını proqnozlaşdırmaq və buna uyğun olaraq qərar proseslərinə dəstək verməkdir.

In [ ]: