

# Human-AI Code Detector Test Case Tabloları

**Tablo 1:** Test Case 1: Model Dosyasının Yüklenmesi

<b>Risk Level</b>	Düşük
<b>Purpose</b>	Logistic Regression model dosyasının sistem tarafından başarıyla yüklenip yüklenmediğini test etmek.
<b>Inputs</b>	Model dosya yolu: LogisticRegression.pkl
<b>Expected Outputs</b>	Model dosyası hata oluşturmadan yüklenir.
<b>Pass Criterias</b>	Model nesnesi başarıyla oluşturulursa.
<b>Fail Criterias</b>	Model dosyası bulunamazsa veya yükleme sırasında hata oluşursa.
<b>Test Procedure</b>	1. Test kodu çalıştırılır. 2. Model dosyası yüklenmeye çalışılır. 3. Model nesnesinin oluşturulduğu doğrulanır.
<b>Test Status</b>	Passed

**Tablo 2:** Test Case 2: Büyük Boyutlu Kod Girdisinin İşlenmesi

<b>Risk Level</b>	Yüksek
<b>Purpose</b>	Sistemin çok satırlı ve uzun kod girdilerini performans kaybı veya hata oluşturmadan işleyebildiğini doğrulamak.
<b>Inputs</b>	100+ satırdan oluşan uzun bir kod bloğu
<b>Expected Outputs</b>	Sistem kod girdisini başarıyla işler ve tahmin sonucunu üretir.
<b>Pass Criterias</b>	Tahmin işlemi makul sürede tamamlanır ve uygulama çökmez.
<b>Fail Criterias</b>	Uygulama yanıt vermez hale gelirse veya hata oluşursa.
<b>Test Procedure</b>	1. Uygulama başlatılır. 2. Uzun ve çok satırlı bir kod bloğu giriş alanına yapıştırılır. 3. “Tahmin Et” butonuna basılır. 4. Uygulamanın yanıt süresi ve durumu gözlemlenir.
<b>Test Status</b>	Passed

**Tablo 3:** Test Case 3: Aynı Girdi İçin Tekrarlı Tahmin Tutarlılığı

<b>Risk Level</b>	Orta
<b>Purpose</b>	Aynı kod girdisi ve aynı model kullanıldığında, sistemin her çalıştırmada tutarlı sonuçlar ürettiğini doğrulamak.
<b>Inputs</b>	Örnek kod girdisi: <code>def test_function(): pass</code>
<b>Expected Outputs</b>	Aynı kod girdisi için her çalıştırmada benzer veya aynı tahmin sonuçları üretilir.
<b>Pass Criterias</b>	Tekrarlı çalıştırmalarda elde edilen tahmin sonuçları arasında anlamlı bir fark yoksa.
<b>Fail Criterias</b>	Aynı girdi için her çalıştırmada ciddi farklılıklar oluşursa.
<b>Test Procedure</b>	<ol style="list-style-type: none"><li>1. Uygulama başlatılır ve bir model seçilir.</li><li>2. Aynı kod girdisi girilerek “Tahmin Et” butonuna basılır.</li><li>3. İşlem en az 3 kez tekrarlanır.</li><li>4. Elde edilen sonuçlar karşılaştırılır.</li></ol>
<b>Test Status</b>	Passed

**Tablo 4:** Test Case 4: Boş Kod Girişi Kontrolü

<b>Risk Level</b>	Düşük
<b>Purpose</b>	Kullanıcı boş kod girdiğinde sistemin uyarı mesajı vermesini doğrulamak.
<b>Inputs</b>	Boş metin
<b>Expected Outputs</b>	“Lütfen kod giriniz” uyarı mesajı gösterilir.
<b>Pass Criterias</b>	Uyarı mesajı ekranda görünürse.
<b>Fail Criterias</b>	Uyarı mesajı gösterilmezse.
<b>Test Procedure</b>	<ol style="list-style-type: none"><li>1. Uygulama başlatılır.</li><li>2. Kod alanı boş bırakılır.</li><li>3. “Tahmin Et” butonuna basılır.</li></ol>
<b>Test Status</b>	Passed

**Tablo 5:** Test Case 5: Çoklu Model Seçimi

<b>Risk Level</b>	Orta
<b>Purpose</b>	Kullanıcının farklı makine öğrenmesi modelleri ile tahmin alabilmesini doğrulamak.
<b>Inputs</b>	Aynı kod girdisi + farklı model seçimi
<b>Expected Outputs</b>	Her model için yüzdelik AI/İnsan tahmini gösterilir.
<b>Pass Criterias</b>	Seçilen her model için sonuç üretirse.
<b>Fail Criterias</b>	Model seçimine rağmen sonuç üretilmezse.
<b>Test Procedure</b>	1. Uygulama başlatılır. 2. Kod girdisi yapılır. 3. Farklı modeller seçilerek tahmin alınır.
<b>Test Status</b>	Passed

**Tablo 6:** Test Case 6: Eksik Model Dosyası Hata Yönetimi

<b>Risk Level</b>	Yüksek
<b>Purpose</b>	Model dosyası eksik olduğunda sistemin çökmeden kullanıcıya uygun bir hata mesajı gösterebildiğini doğrulamak.
<b>Inputs</b>	Model dosyası silinmiş/taşınmış durumda: <code>LogisticRegression.pkl</code> yok
<b>Expected Outputs</b>	Uygulama çökmez, kullanıcıya “Model dosyası bulunamadı” benzeri bir hata mesajı gösterilir.
<b>Pass Criterias</b>	Uygulama kapanmadan hata mesajını gösterirse ve çalışmaya devam ederse.
<b>Fail Criterias</b>	Uygulama kapanırsa, donarsa veya kullanıcıya mesaj göstermeden hata verirse.
<b>Test Procedure</b>	1. <code>app/LogisticRegression.pkl</code> dosyasını geçici olarak yeniden adlandır (örn. <code>LogisticRegression_backup.pkl</code> ). 2. Uygulamayı başlat. 3. Tahmin işlemi başlatılmaya çalışılır. 4. Sistem davranışı ve hata mesajı gözlemlenir. 5. Testten sonra dosya adını eski haline getir.
<b>Test Status</b>	Passed