İSTANBUL TİCARET
ÜNİVERSİTESİ

**Yapay Zeka Uygulamaları Dönem Ödevi**

Ömer BATUK
200030617

Öğretim Görevlisi
Ali Mertcan KÖSE

## İçindekiler

## *1.1 Projede İstenenler*

Bu projede, belirli bir veri seti kullanılarak çeşitli performans metriklerinin hesaplanması ve görselleştirilmesi istenmiştir. Bu metrikler arasında AUC (Area Under the Curve) değeri, ROC (Receiver Operating Characteristic) eğrisi, confusion matrix, precision ve recall (sensitivity) bulunmaktadır. Bu metrikler, bir sınıflandırma modelinin performansını değerlendirmek için yaygın olarak kullanılır ve her biri farklı bir performans yönünü temsil eder.

## *1.2 Metrikler Hakkında Bilgiler*

**AUC Değeri**

AUC, ROC eğrisinin altındaki alanı temsil eder. ROC eğrisi, modelin doğru pozitif oranı (True Positive Rate, TPR) ile yanlış pozitif oranı (False Positive Rate, FPR) arasındaki ilişkiyi gösterir. AUC, modelin sınıflandırma yeteneğinin genel bir ölçüsüdür. 1'e ne kadar yakınsa, modelin performansı o kadar iyidir. AUC değeri, modelin rastgele bir pozitif örneği rastgele bir negatif örnekten daha yüksek bir skorla sınıflandırma olasılığını gösterir.

**ROC Eğrisi**

ROC eğrisi, farklı eşik değerlerinde modelin TPR ve FPR oranlarını grafiksel olarak gösterir. Bu eğri, modelin çeşitli eşiklerdeki performansını değerlendirmeye yardımcı olur. Eğrinin sol üst köşesine yakın olması, modelin yüksek bir TPR ve düşük bir FPR oranına sahip olduğunu ve dolayısıyla iyi performans gösterdiğini gösterir.

**Confusion Matrix**

Confusion matrix, modelin tahmin sonuçlarını ve gerçek sınıfları karşılaştıran bir matristir. Matris dört temel bileşenden oluşur:

- True Positive (TP): Doğru bir şekilde pozitif olarak sınıflandırılan örnekler.
- False Positive (FP): Yanlış bir şekilde pozitif olarak sınıflandırılan örnekler.
- True Negative (TN): Doğru bir şekilde negatif olarak sınıflandırılan örnekler.
- False Negative (FN): Yanlış bir şekilde negatif olarak sınıflandırılan örnekler.

  Confusion matrix, modelin hatalarını ve doğru tahminlerini ayrıntılı olarak göstermeye yardımcı olur.

**Precision (Kesinlik)**

Precision, doğru pozitif tahminlerin toplam pozitif tahminlere oranını gösterir. Yüksek precision değeri, modelin pozitif tahminlerinde az sayıda yanlış pozitif bulunduğunu gösterir.

**Recall (Sensitivity, Duyarlılık)**

Recall veya sensitivity, doğru pozitif tahminlerin toplam gerçek pozitiflere oranını gösterir. Yüksek recall değeri, modelin pozitif örneklerin büyük bir kısmını doğru bir şekilde tahmin ettiğini gösterir.

### *1.3 Projenin Konusu*

Bu projenin konusunu 16 Kasım 1994'de UC Irvine Machine Learning Repository'ye bağışlanmış "German Credit Data" veri setiyle dönemin Almanya'sındaki kredi sahiplerini analiz ederek yeni krediye başvuran adayları iyi / kötü kredi skoru olarak tahminde bulunur.

## *2.1 Kullanılan Veri Seti*

Bu projede UCI tarafından paylaşılan “German Credit Data” verisi kullanılmıştır.

## *2.2 Veri Seti Hakkında Ön Bilgiler*

Veri Kümesi Özellikleri : Çok Değişkenli
Konu Alanı : Sosyal Bilimler
Araştırma Zaman / Mekanı : 1994, Almanya
İlgili Görevler : Sınıflandırma
Özellik Türleri : Kategorik, Tamsayı
Örnekler : 1000
Özellikler : 20
Eksik değer bulunmamaktadır.

## *2.3 Değişken Bilgileri*

---

***1.Değişken**: Mevcut çek hesabının durumu.*

A11: … < 0 DM*

A12: 0 <= … < 200 DM

A13: … > = 200 DM veya en az 1 yıl maaş ödemesi.

A14: Çek hesabı yok.

* **DM**: Almanya'da Euro'ya geçinceye kadar kullanılmış para birimidir. ( 1948 - 1990 Batı Almanya / 1990 - 2002 Doğu ve Batı Almanya)

---

***2.Değişken**: Ay cinsinden çek süresi.*

---

***3.Değişken**: Kredi Geçmişi*

A30: Hiç kredi alınmadı veya tüm krediler usulüne uygun olarak ödendi.

A31: Bu bankadaki tüm krediler usulüne uygun olarak ödendi.

A32: Şu ana kadar usulüne uygun olarak olarak geri ödenen mevcut kredi sahibi.

A33: Geçmişte çek ödemede gecikmiş.

A34: Kritik hesap / Başka bankalarda mevcut kredi sahibi.

---

***4.Değişken:*** *Kredi amacı*

| | | |
|---|---|---|
| A40: Araba(Yeni) | A41: Araba(Kullanılmış) | A42: Mobilya/Ekipman |
| A43: Radyo/Televizyon | A44: Ev Aletleri | A45: Onarım/Tamirat |
| A46: Eğitim | A47: Tatil | A48: Mesleki Eğitim |
| A49: İş | A410: Diğer | |

---

***5.Değişken:*** *Sayısal biçimde kredi miktarı.*

---

***6.Değişken****: Müşteri tasarruf hesabı hakkında bilgi.*

A61: … < 100 DM

A62: 100 <= … < 500 DM

A63: 500 <= .. < 1000 DM

A64: … >= 1000 DM

A65: Bilinmiyor veya tasarruf hesabı yok.

---

***7.Değişken:*** *Mevcut çalışma süresi.*

A71: İşsiz

A72: … < 1 yıl

A73: 1 <= … < 4 yıl

A74: 4 <= … < 7 yıl

A75: … >= 7 yıl

---

***8.Değişken:*** Gelirin kullanılabilir kısmının yüzdesi olarak taksit oranı.

---

***9.Değişken:*** *Cinsiyet ve Medeni Durumu*

A91 : Erkek : Boşanmış/Ayrılmış

A92 : Kadın: Boşanmış/Ayrı düşmüş/Evli

A93 : Erkek : Bekar

A94 : Erkek : Evli/Dul

A95 : Kadın : Bekar

---

***10.Değişken:*** *Diğer borçlular veya garantörler*

A101 : Yok

A102 : Eş-başvuru sahibi

A103 : Kefil

---

***11.Değişken:*** *Sayısal biçimde şu andan itibaren mevcut ikamet.*

---

***12.Değişken:*** *Mülkleri*

A121 : emlak

A122 : A121 Değilse: yapı kooperatifi tasarruf sözleşmesi/hayat sigortası

A123 : A121 veya A122 değilse: Araba veya başka, Değişken 6'da olmayan

A124 : Bilinmiyor / Yok

---

***13.Değişken:*** *Yıl cinsinden yaş*

---

***14.Değişken:*** *Diğer mevcut taksit planları*

A141: Banka

A142: Mağaza

A143: Yok

---

***15.Değişken:*** *Konut bilgisi*

A151: Kirada

A152: Kendisinin

A153: Ücretsiz

---

***16.Değişken:*** *Bu bankadaki mevcut kredi sayısı.*

---

***17.Değişken:*** *Meslek*

A171: İşsiz / Vasıfsız - Yerleşik Değil

A172: Vasıfsız - Yerleşik

A173: Vasıflı çalışan - Yetkili

A174: Yönetim / Serbest meslek sahibi / Yüksek vasıflı çalışan / Memur

---

***18.Değişken:*** *Bakımını sağlamakla yükümlü olduğu kişi sayısı.*

---

***19.Değişken:*** *Telefon*

A191: Yok

A192: Evet, müşteri üzerine kayıtlı.

---

***20.Değişken:*** *Yabancı işçi mi ?*

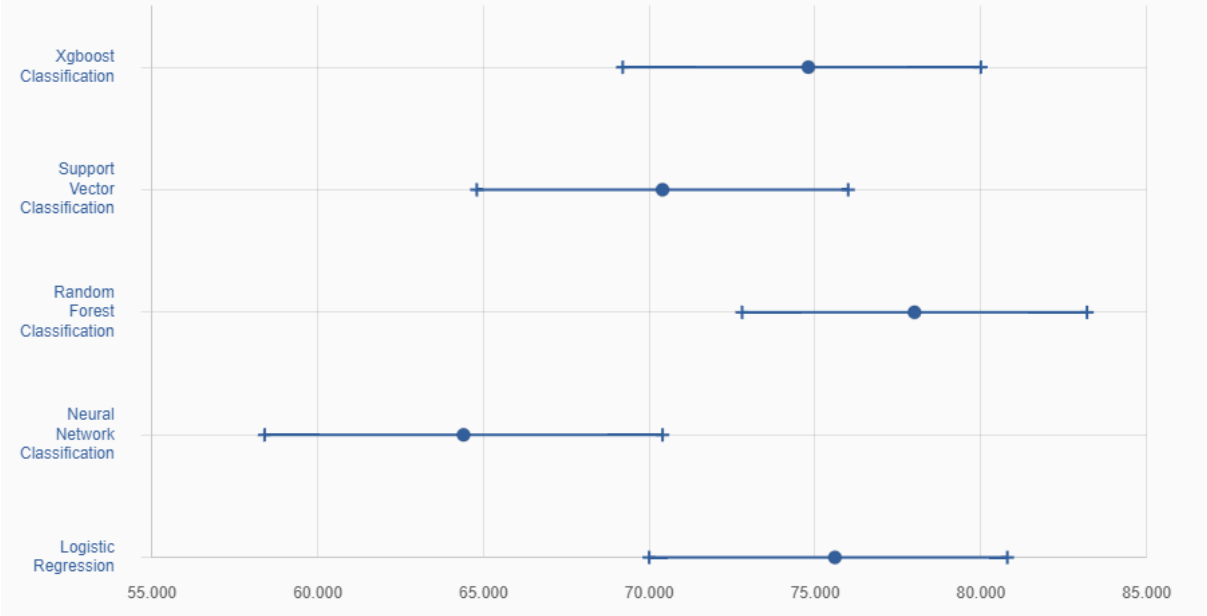
A201: Evet

A202: Hayır

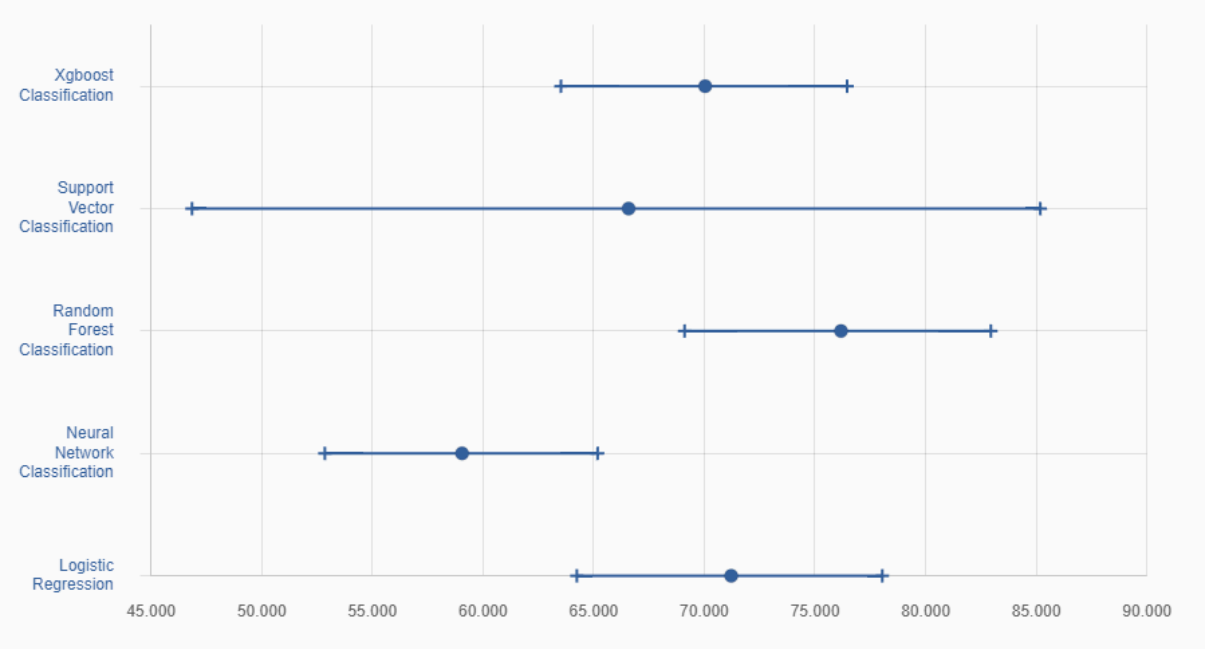
---

## *2.4 Temel Model Performansı*

***Accuracy(Doğruluk)***



### *Precision(Kesinlik)*




## *2.5 Algoritma Seçimi*

### *Doğruluk Bakımından*

Grafiklere bakıldığında, Random Forest algoritmasının doğruluk değeri %77 ile en yüksek değeri sağlıyor. Diğer algoritmalar %70 ila %72 aralığında doğruluk değerlerine sahip. Bu, Random Forest algoritmasının doğru ve yanlış sınıflandırmalarını hesaba katan genel performans açısından üstün olduğunu gösteriyor.

### *Kesinlik Bakımından*

Kesinlik grafiğinde de Random Forest algoritması %75 ile en yüksek değeri alıyor. Diğer algoritmaların kesinlik değerleri %63 ila %70 arasında değişiyor. Kesinlik, pozitif olarak tahmin edilen sonuçların ne kadarının gerçekten pozitif olduğunu gösterir. Bu, özellikle kredi riskinin belirlenmesi gibi kritik alanlarda yanlış pozitif oranını düşük tutmanın önemli olduğu durumlarda değerlidir.

Bu nedenle, bu projede Random Forest algoritmasını seçmek mantıklıdır. Hem doğru tahmin oranının yüksekliği hem de pozitif tahminlerin doğruluğu açısından Random Forest, German Credit Data veri setinde en verimli algoritma olarak öne çıkmaktadır

## Bölüm 3 - Kod

### *3.1 Kullanılan Araç ve Kütüphaneler*

### *Araçlar:*

*PyCharm*: Python projeleri geliştirmek için kullanılan entegre geliştirme ortamı (IDE). Bu projede kod yazımı, düzenlemesi ve çalıştırılması için PyCharm kullanılmıştır.

---

### *Kütüphaneler:*

**pandas**: Veri manipülasyonu ve analizi için kullanılan kütüphane. Bu projede veri yükleme, temizleme ve ön işleme için kullanılmıştır.

**numpy**: Sayısal hesaplamalar için kullanılan kütüphane. Veri manipülasyonlarında ve çeşitli hesaplamalarda destek amaçlı kullanılmıştır.

**scikit-learn**: Makine öğrenimi modelleri oluşturmak ve değerlendirmek için kullanılan kütüphane. Bu projede veri setinin eğitim ve test olarak ayrılması, model eğitimi, performans değerlendirmesi ve metrik hesaplamaları için kullanılmıştır.

**imblearn**: Sınıf dengesizliği problemleri ile başa çıkmak için kullanılan kütüphane. Bu projede SMOTE (Synthetic Minority Over-sampling Technique) yöntemi ile eğitim veri setinde dengeleme yapılmıştır. (*bknz: s.10 - 4.kod bloğu)*

**matplotlib:** Grafik ve görselleştirme için kullanılan kütüphane. Model performansının görselleştirilmesi için ROC eğrisi çiziminde kullanılmıştır.

**seaborn**: İstatistiksel verilerin görselleştirilmesi için kullanılan kütüphane. Confusion matrix'in ısı haritası olarak görselleştirilmesinde kullanılmıştır.

### *3.2 Veri Setinin Aktarımı*

Veri setini projemize aktarırken yerel bir dosya olarak yüklemek yerine, UCI'ın bize sağladığı link üzerinden aktarımı sağlayıp sütunlara böldük.

```
        url =
"https://archive.ics.uci.edu/ml/machine-learning-databases/statlog/german/german.data"
        columns = ["Status", "Duration", "CreditHistory", "Purpose", "CreditAmount",
"Savings", "Employment",
                   "InstallmentRate", "PersonalStatus", "OtherDebtors", "ResidenceSince",
"Property", "Age",
                   "OtherInstallmentPlans", "Housing", "ExistingCredits", "Job",
"LiablePeople", "Telephone",
                   "ForeignWorker", "CreditRisk"]
        df = pd.read_csv(url, delimiter=' ', header=None, names=columns)
```

```
df['CreditRisk'] = df['CreditRisk'].map({1: 1, 2: 0})
```

### 3.3 Veri Ayarları ve Modeli Eğitme

Veriyi özellikler ve hedef değişken olarak ayırma

```
X = df.drop("CreditRisk", axis=1)
y = df["CreditRisk"]
X = pd.get_dummies(X, drop_first=True)
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)
```

Veriyi standartlaştırma

```
scaler = StandardScaler()
X_train = scaler.fit_transform(X_train)
X_test = scaler.transform(X_test)
```

SMOTE ile sınıf dengesizliğini giderme.

```
sm = SMOTE(random_state=42)
X_train_res, y_train_res = sm.fit_resample(X_train, y_train)
```

Modeli Eğitme

```
model = RandomForestClassifier(random_state=42)
model.fit(X_train_res, y_train_res)
```

### 3.4 Kullanıcıdan Veri Alma

Kullanıcıdan uygulama kısmı için input alma

```
def user_input_features():
    print("Müşteri Bilgilerini Girin:")

    Status = input("Durum (A11, A12, A13, A14): ")
    Duration = int(input("Süre (ay): "))
    CreditHistory = input("Kredi Geçmişi (A30, A31, A32, A33, A34): ")
    Purpose = input("Kredi Amacı (A40, A41, A42, A43, A44, A45, A46, A47, A48, A49, A410):
    ")
    CreditAmount = int(input("Kredi Miktarı: "))
    Savings = input("Tasarruflar (A61, A62, A63, A64, A65): ")
    Employment = input("İstihdam Süresi (A71, A72, A73, A74, A75): ")
    InstallmentRate = int(input("Taksit Oranı (%): "))
    PersonalStatus = input("Kişisel Durum ve Cinsiyet (A91, A92, A93, A94): ")
    OtherDebtors = input("Diğer Borçlular (A101, A102, A103): ")
    ResidenceSince = int(input("İkamet Süresi: "))
    Property = input("Mülkiyet (A121, A122, A123, A124): ")
    Age = int(input("Yaş: "))
    OtherInstallmentPlans = input("Diğer Taksit Planları (A141, A142, A143): ")
    Housing = input("Konut (A151, A152, A153): ")
    ExistingCredits = int(input("Mevcut Krediler: "))
    Job = input("İş (A171, A172, A173, A174): ")
    LiablePeople = int(input("Bakmakla Yükümlü Kişiler: "))
    Telephone = input("Telefon (A191, A192): ")
    ForeignWorker = input("Yabancı İşçi (A201, A202): ")
```

Kullanıcıdan alınan veriyi atama

```
    data = {
        'Status': Status,
        'Duration': Duration,
        'CreditHistory': CreditHistory,
        'Purpose': Purpose,
        'CreditAmount': CreditAmount,
        'Savings': Savings,
        'Employment': Employment,
        'InstallmentRate': InstallmentRate,
        'PersonalStatus': PersonalStatus,
        'OtherDebtors': OtherDebtors,
        'ResidenceSince': ResidenceSince,
        'Property': Property,
        'Age': Age,
        'OtherInstallmentPlans': OtherInstallmentPlans,
```

```
                                                    'Housing': Housing,
                                                'ExistingCredits': ExistingCredits,
                                                        'Job': Job,
                                                'LiablePeople': LiablePeople,
                                                    'Telephone': Telephone,
                                                'ForeignWorker': ForeignWorker
                                                    }
                                            features = pd.DataFrame(data, index=[0])
                                                    return features


                                        input_df = user_input_features()
```

## *3.5 Veri Düzenleme ve Metrik İşlemler*

*-gun hale getirme.

```
input_df = pd.get_dummies(input_df)
input_df = input_df.reindex(columns=X.columns, fill_value=0)
input_df = scaler.transform(input_df)
```

Tahmin yapma ve sonucu gösterme

```
prediction = model.predict(input_df)
prediction_proba = model.predict_proba(input_df)
print("\nTahmin Sonucu")
print("İyi Kredi" if prediction[0] == 1 else "Kötü Kredi")

print("\nİyi Kredi Olasılığı")
print(prediction_proba[0][1])
```

Performans metriklerini hesaplama ve gösterme

```
print("\nPerformans Metrikleri")
y_pred = model.predict(X_test)
print("ROC AUC Skoru:", roc_auc_score(y_test, model.predict_proba(X_test)[:, 1]))
print("Confusion Matrix:")
print(confusion_matrix(y_test, y_pred))
print("Precision Skoru:", precision_score(y_test, y_pred))
print("Recall Skoru (Sensitivity):", recall_score(y_test, y_pred))
```

ROC eğrisini çizme

```
fpr, tpr, _ = roc_curve(y_test, model.predict_proba(X_test)[:, 1])
plt.figure()
plt.plot(fpr, tpr, color='blue', lw=2,
        label='ROC curve (area = %0.2f)' % roc_auc_score(y_test,
model.predict_proba(X_test)[:, 1]))
plt.plot([0, 1], [0, 1], color='gray', lw=2, linestyle='--')
plt.xlim([0.0, 1.0])
plt.ylim([0.0, 1.05])
plt.xlabel('False Positive Rate')
plt.ylabel('True Positive Rate')
plt.title('ROC Eğrisi')
plt.legend(loc="lower right")
plt.show()
```

Confusion Matrix ısı haritasını çizme

```
cm = confusion_matrix(y_test, y_pred)
sns.heatmap(cm, annot=True, fmt='d')
plt.xlabel('Tahmin Edilen')
plt.ylabel('Gerçek')
plt.title('Confusion Matrix Isı Haritası')
plt.show()
```

## *4.1 Uygulama Çıktısı Hakkında Bilgi*

Projede kullanılan Random Forest modeli, eğitim ve test aşamalarından sonra belirli performans metriklerine göre değerlendirildi. Uygulama çıktısı, kullanıcının belirttiği müşteri bilgilerine dayanarak kredi riskini tahmin eder ve ilgili performans metriklerini sağlar.

### *Çıktılar*

### *Tahmin Sonucu:*

Kullanıcıdan alınan bilgiler doğrultusunda, müşterinin kredi riskini “İyi Kredi” veya “Kötü Kredi” olarak sınıflandırılmaktadır.

---

### *İyi Kredi Olasılığı:*

Müşterinin iyi krediye sahip olma olasılığı hesaplanır ve yüzdelik olarak sunulur.

---

### *Performans Metrikleri:*

ROC AUC Skoru, Confusion Matrix, Precision (Kesinlik) Skoru, Recall (Duyarlılık) Skoru, ROC Eğrisi çizimi, Confusion Matrix Isı Haritası çizimi.

---

## *4.2 Örnekler*

Bu örnekte model, müşterinin kredi riskini düşük olarak sınıflandırır ve yüksek bir iyi kredi olasılığı gösterir. Bunun gerçekleşmesi için uygulamayı çalıştırıp gerekli bilgileri dolduralım.

### *4.2.1 İyi Kredi Sonucu Veren Örnek:*

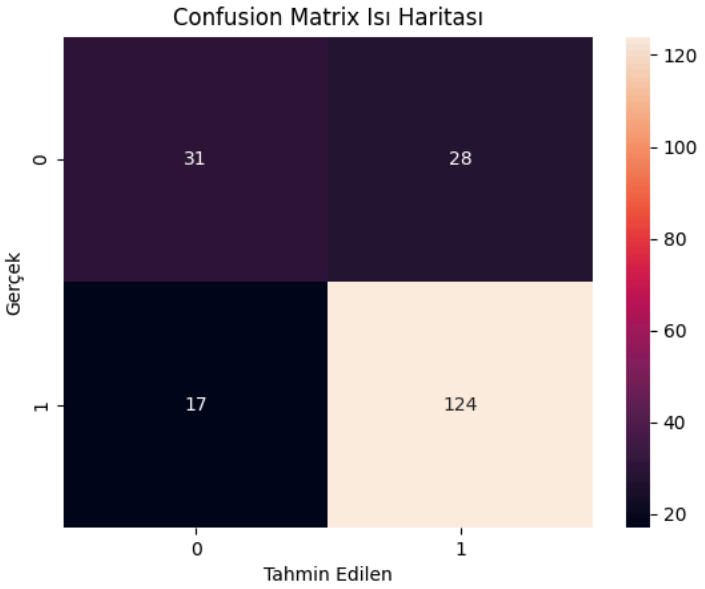
Örnek için girdiğimiz bilgiler.
A13, 12, A30, A40, 2750, A64, A75, 10, A94, A101, 5, A121, 30, A143, A152, 0, A174, 1, A192, A202

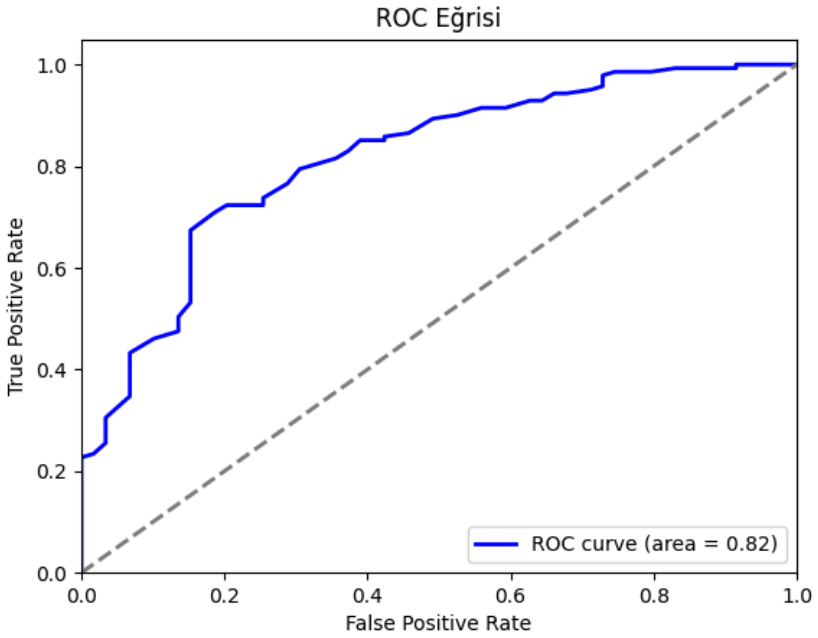
**Yukarıdaki veriler genel olarak iyi bir kredi sonucu verilmesi hedefiyle girilmiştir ve sonucu aşağıdadır !**

```
Tahmin Sonucu
İyi Kredi

İyi Kredi Olasılığı
0.69
```

```
Performans Metrikleri
ROC AUC Skoru: 0.815482630123813
Confusion Matrix:
[[ 31  28]
 [ 17 124]]
Precision Skoru: 0.8157894736842105
Recall Skoru (Sensitivity): 0.8794326241134752
```






### *4.2.2 Kötü Kredi Sonucu Veren Örnek:*

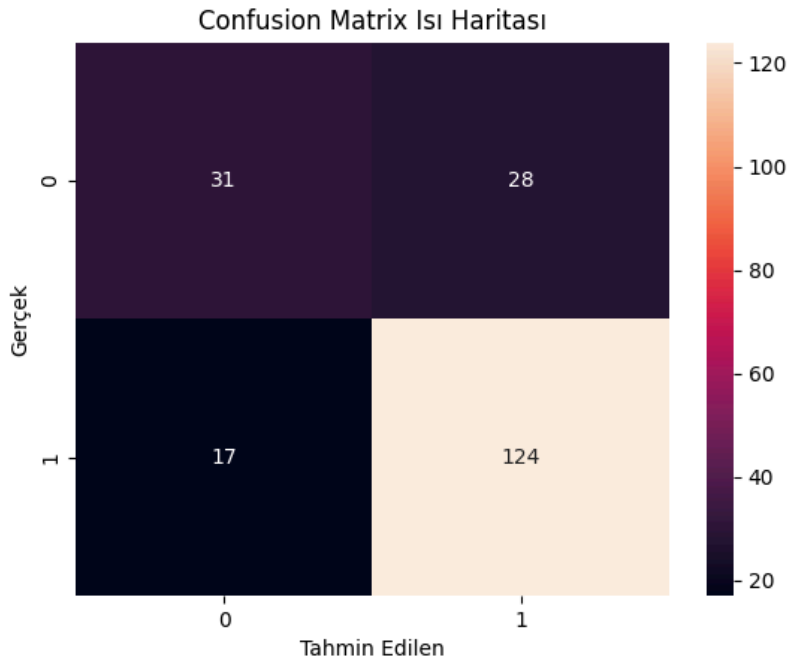
Örnek için girdiğimiz veriler:
A12, 36, A32, A46, 12750, A62, A73, 1, A93, A101, 4, A124, 47, A143, A153, 1, A173, 2, A192, A201, 2

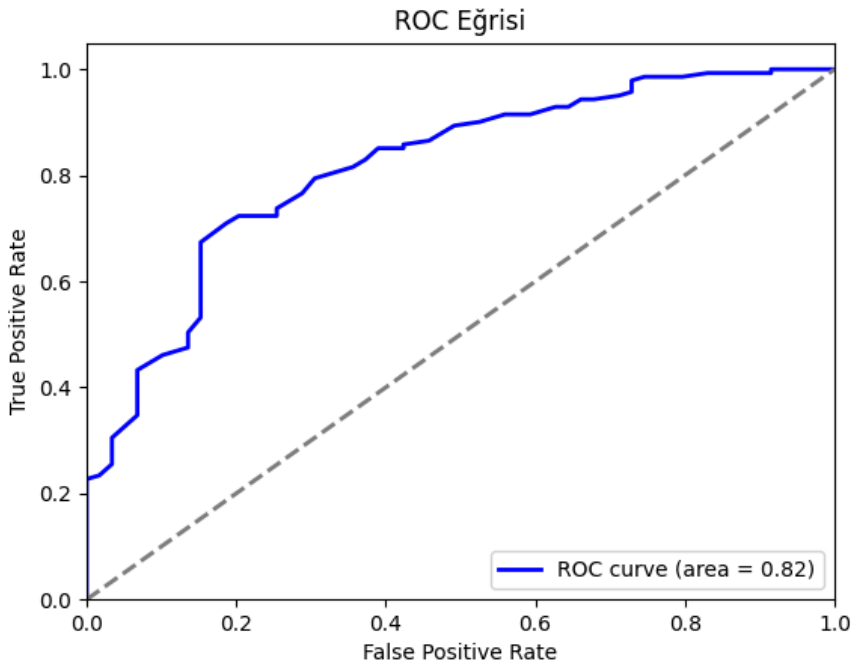
**Yukarıdaki veriler genelde kötü olarak sınıflandırabileceği düşünülüp girilmiştir!**

```
Tahmin Sonucu
Kötü Kredi

İyi Kredi Olasılığı
0.16
```

```
Performans Metrikleri
ROC AUC Skoru: 0.815482630123813
Confusion Matrix:
[[ 31  28]
 [ 17 124]]
Precision Skoru: 0.8157894736842105
Recall Skoru (Sensitivity): 0.8794326241134752
```


Confusion Matrix Isı Haritası
31
28
17
124
0
1
Gerçek
Tahmin Edilen
120
100
80
60
40
20


ROC Eğrisi
True Positive Rate
False Positive Rate
ROC curve (area = 0.82)
0.0
0.2
0.4
0.6
0.8
1.0