Alman Kredi Riski Tahmini - Öznitelik Mühendisliği Projesi

Samet Baturay - baturay.samet@ogr.iu.edu.tr

Veri Kaynağı : https://www.kaggle.com/datasets/kabure/german-credit-data-with-risk/

1. **Projenin AMACI**

   Bu projenin temel amacı, **Alman Kredi Veri Seti (German Credit Data)** kullanılarak müşterilerin kredi ödeme risklerini (Good/Bad) yüksek doğrulukla tahmin eden bir makine öğrenmesi modeli geliştirmektir.

   Ancak projenin teknik odağı, sadece tahminleme yapmak değil, ham veri üzerinde **Öznitelik Mühendisliği (Feature Engineering)** ve **Öznitelik Seçimi (Feature Selection)** tekniklerinin model performansı üzerindeki etkisini incelemektir.

2. **Veri Setinin Tanımlanması**

- Veri seti toplam 1000 satır ve 10 sütundan oluşmaktadır.

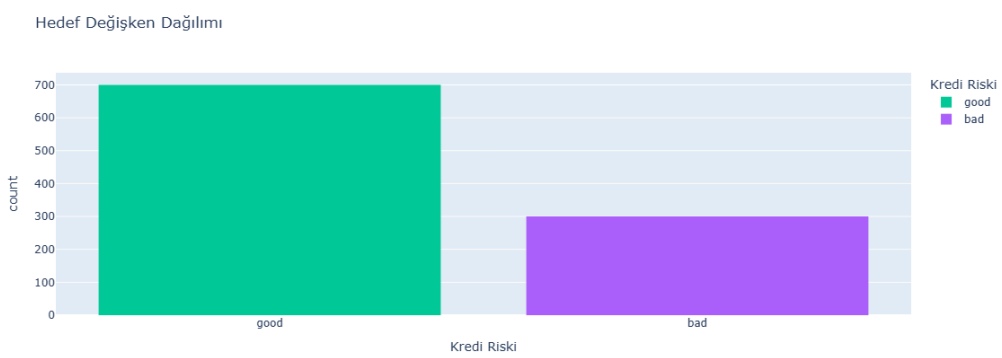
▼ Veri setinin ilk 5 satırı aşağıdaki gibidir

	Age	Sex	Job	Housing	Saving accounts	Checking account	Credit amount	Duration	Purpose	Risk
0	67	male	2	own	NaN	little	1169	6	radio/TV	good
1	22	female	2	own	little	moderate	5951	48	radio/TV	bad
2	49	male	1	own	little	NaN	2096	12	education	good
3	45	male	2	free	little	little	7882	42	furniture/equipment	good
4	53	male	2	free	little	little	4870	24	car	bad

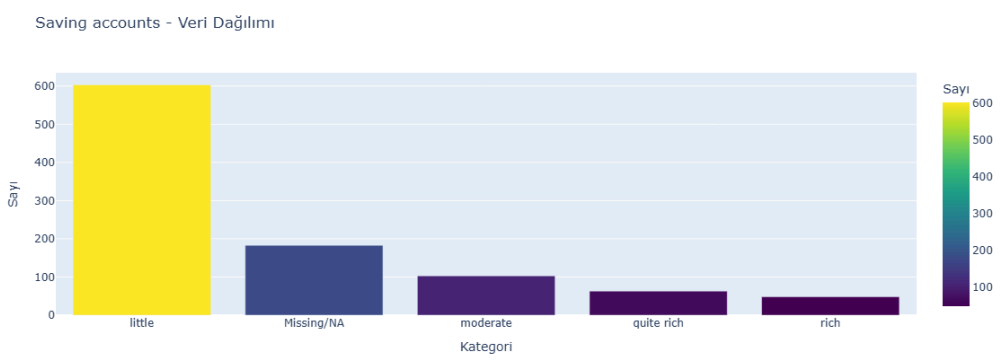
▼ Eksik veriler ve veri türleri aşağıdaki gibidir.

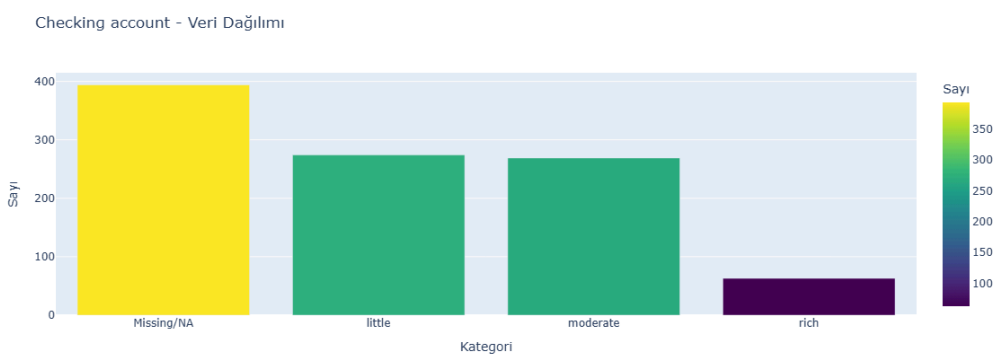
```
Data columns (total 10 columns):
 #   Column            Non-Null Count  Dtype
---  ------            --------------  -----
 0   Age               1000 non-null   int64
 1   Sex               1000 non-null   object
 2   Job               1000 non-null   int64
 3   Housing           1000 non-null   object
 4   Saving accounts   817 non-null    object
 5   Checking account  606 non-null    object
 6   Credit amount     1000 non-null   int64
 7   Duration          1000 non-null   int64
 8   Purpose           1000 non-null   object
 9   Risk              1000 non-null   object
```

▼ Hedef değişken dağılımı aşağıdaki gibidir.

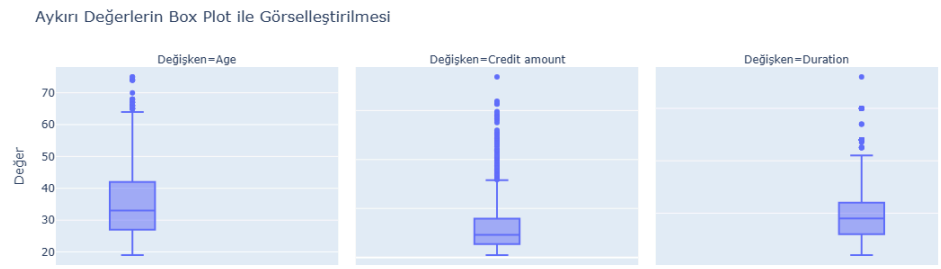


▼ Eksik veri bulunan “Saving account” ve “Checking Account” kolonlarının dağılımı aşağıdaki gibidir.





▼ Aykırı değer bulunan kolonlar aşağıda Box Plot ile görselleştirilmiştir.



3. Yöntem

▼ Preprocessing

▼ Eksik Veri Yönetimi

“Saving account” ve “Checking Account” kolonlarında eksikler vardır.
Bir müşterinin tasarruf hesabı (Saving account) veya çek hesabı (Checking Account) olmaması, **kredi riski açısından çok değerli bir bilgidir**. Bu yüzden eksik verileri silmek yerine yeni bir kategori olarak ("No Account") tutulmuştur.

▼ Aykırı Değerlerin Yönetimi

- Finansal verilerde aykırı değerler gerçek durumları yansıtabilir. Bu yüzden aykırı değerler için sadece ölçekleme sırasında aykırı değerlere çok dayanıklı olan RobustScaler kullanılmıştır.

▼ Encoding

- Hedef değişken olan “Risk” , 1 ve 0’lara dönüştürüldü (good =1 , bad =0)
- “Saving account” ve “Checking Account” kolonları, ordinal yapıda olduğundan ordinal encoding yapılmıştır.
  - Sıralama: No Account < little < moderate < quite
- “Sex” , “Job”, “Housing” , “Purpose” kolonlarına one hot encoding yapıldı.
  - Burada akıllara “Job” zaten integer neden one hot encoding yaptık diye bir soru gelebilir. “Job” kolonundaki sayısal değerler (0,1,2,3) sadece birer etiket, matematiksel bir üstünlük anlamı yok.
- Encoding işlemleri sonrası öznitelik sayısı 18 olmuştur.

▼ Ölçeklendirme

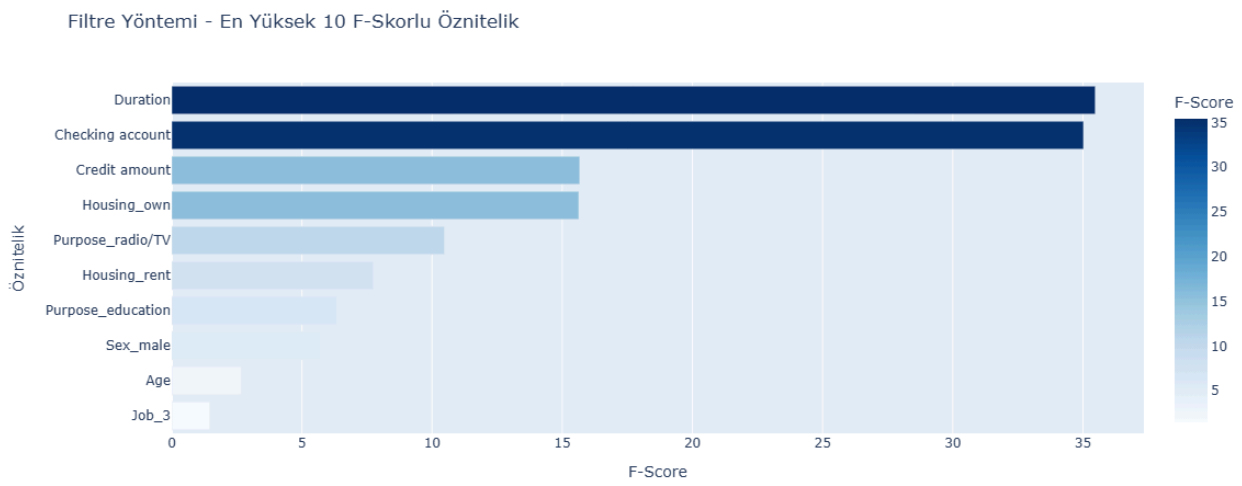
- Aykırı değerlere karşı dirençli olan RobustScaler yöntemi kullanılmıştır.

▼ Öznitelik Seçimi

Burada bizden filtre, sarmalama ve gömülü olmak üzere 3 farklı yöntem kullanmamız bekleniyordu.

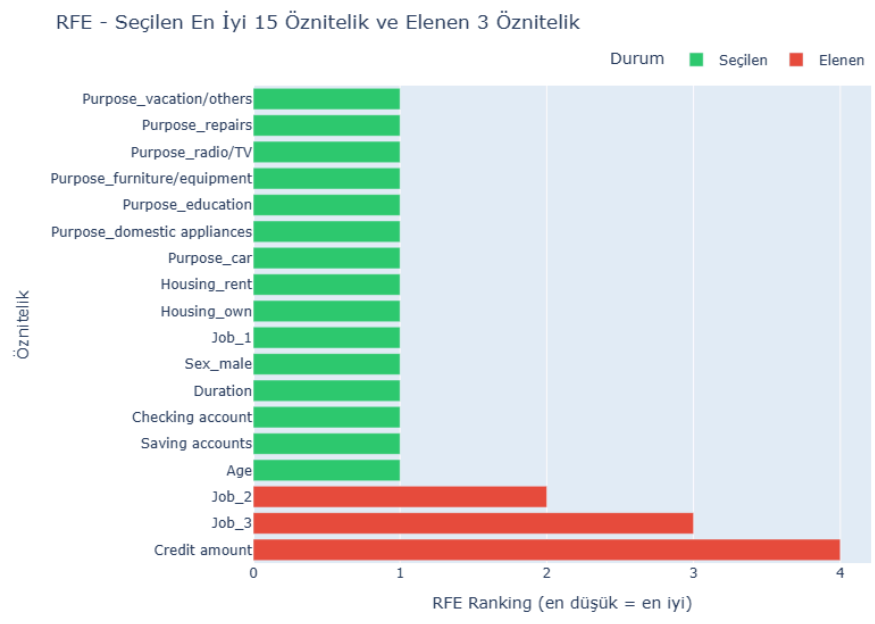
▼ SelectKBest - Filtre Yöntemi

- Öncelikle 5, 10, 15 arasından en iyi k değeri bulundu. Bu değer hesaplanırken random forest ile roc auc skorlarına bakıldı ve k = 10 olarak bulundu.
  Daha sonra k = 10 için en yüksek F-skora sahip 10 öznitelik seçildi.



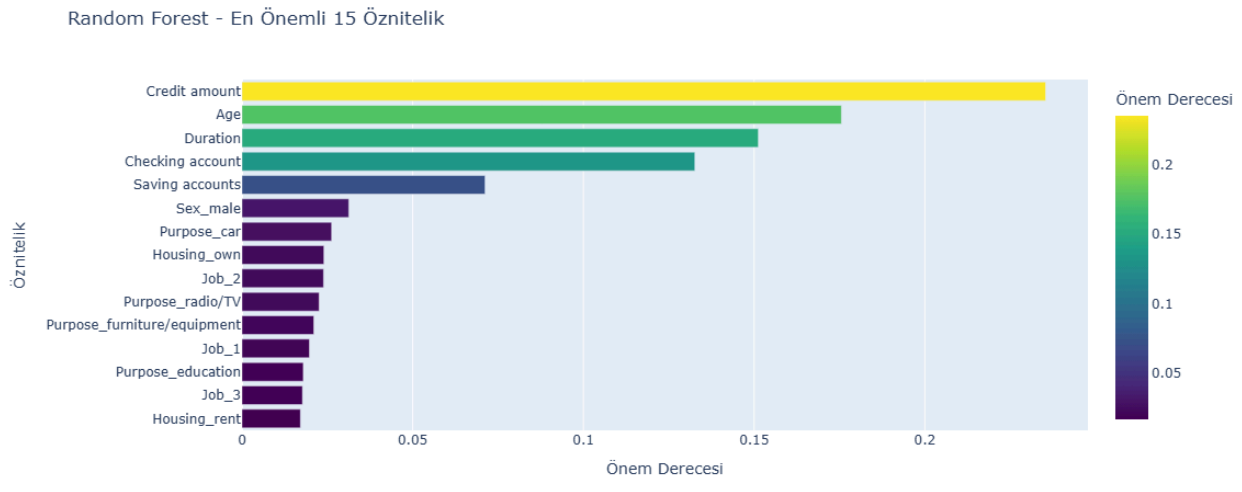
▼ RFE - Sarmalama Yöntemi

Burada öznitelik seçiminde estimator olarak logistic regresyon kullanıldı çünkü RFE iteratif bir süreçtir ve logistic regresyon bu durumda hızlı ve etkilidir. Fakat seçilen özniteliklerin performans değerlendirilmesi için yine random forest kullanılmıştır.



▼ Gömülü Yöntem

Burada feature_importances değeri random forest üzerinden hesaplanmıştır.



Çıkan sonuca göre ilk 5 özniteliği seçmeye karar veriyorum.

▼ **Modelleme**

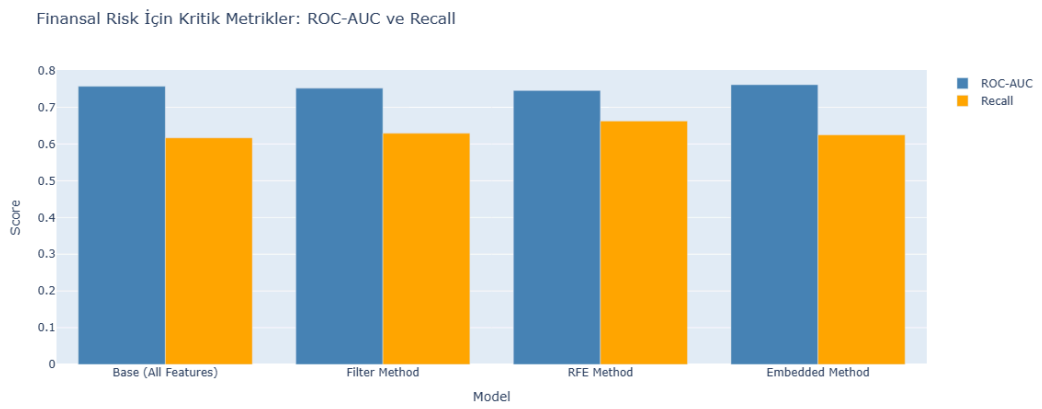
Modellerin performansını karşılaştırmak için dört farklı öznitelik seti oluşturulmuş ve her biri **5-Katlı Çapraz Doğrulama (5-Fold Cross Validation)** ile test edilmiştir.

1. **Baz Model (Base Model):** Herhangi bir öznitelik seçimi uygulanmadan, veri setindeki tüm değişkenlerin (18 öznitelik) kullanıldığı referans modeldir.
2. **Filtre Yöntemi (Filter Method):** SelectKBest algoritması kullanılarak, istatistiksel skorlarına göre en iyi performansı veren **10 öznitelik** ile kurulmuştur.
3. **Sarmalama Yöntemi (Wrapper Method - RFE):** Geriye Doğru Öznitelik Eleme (RFE) tekniği ile en anlamlı bulunan **15 öznitelik** seçilmiştir.
4. **Gömülü Yöntem (Embedded Method):** Modelin öğrenme sürecinde belirlediği önem derecelerine göre (Feature Importance) en kritik **5 öznitelik** kullanılmıştır.

4. **Bulgular ve Model Performansları**

Bu bölümde, öznitelik seçimi uygulanmayan "Base Model" ile üç farklı öznitelik seçimi yöntemi (Filter, Wrapper, Embedded) uygulanmış modellerin performansları karşılaştırılmıştır.

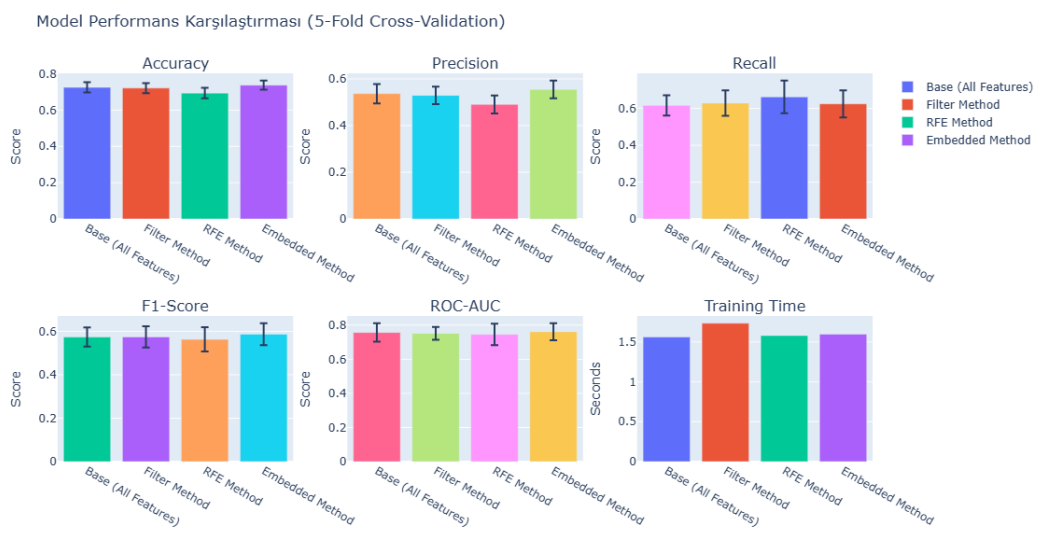
▼ **ROC-AUC Skorları**



Modellerin ayırt edicilik gücünü gösteren ROC-AUC skorlarına göre performans sıralaması şu şekildedir:

1. **Embedded Method (Gömülü Yöntem): 0.7612** (En Yüksek Başarı)
   - Kullanılan Öznitelik Sayısı : 5
2. **Base Model (Tüm Öznitelikler):** 0.7571
   - Kullanılan Öznitelik Sayısı : 18
3. **Filter Method (Filtre Yöntemi):** 0.7518
   - Kullanılan Öznitelik Sayısı : 10
4. **RFE Method (Sarmalama Yöntemi):** 0.7457
   - Kullanılan Öznitelik Sayısı : 15

▼ **Genel Analiz**



- **Verimlilik Artışı:** Base model 18 değişkenle 0.7571 skoruna ulaşırken, Embedded yöntem sadece 5 değişkenle bu skoru geçerek 0.7612 seviyesine taşımıştır. Bu durum, veri setindeki bazı değişkenlerin modelin öğrenmesini zorlaştırdığını kanıtlamaktadır.
- **Recall (Duyarlılık) Değeri:** Kredi riski tahmininde "Bad" (Riskli) müşteriyi kaçırmamak hayati önem taşır. Embedded yöntem, ROC-AUC başarısının yanı sıra **0.6333 Recall** değeri ile riskli müşterileri tespit etme konusunda da dengeli bir performans sergilemiştir.
- **RFE Yönteminin Başarısızlığı:** RFE yöntemi 15 değişken seçmesine rağmen en düşük performansı (0.7457) göstermemiştir. Bu durum, değişkenler arasındaki çoklu bağlantıların sarmalama yöntemini olumsuz etkilediğini göstermektedir.

▼ **Sonuç Değerlendirmesi**

Yapılan analizler neticesinde; Embedded Method sayesinde gereksiz 13 öznitelik elenerek model karmaşıklığı ciddi oranda azaltılmış, veri setindeki gürültü temizlenmiş ve sadece en kritik 5 değişken ile (daha az işlem maliyetiyle), tüm değişkenlerin kullanıldığı modele kıyasla daha yüksek bir tahmin başarısı elde edilmiştir.