

BİLGİSAYAR VE BİLİŞİM BİLİMLERİ FAKÜLTESİ

BİLGİSAYAR MÜHENDİSLİĞİ

MAKİNE ÖĞRENMESİ DERSİ PROJESİ

PROJE: Makine Öğrenmesi ile Gelir Tahmini

GRUP ÜYELERİNİN

ADI-SOYADI…....:MERVE ŞENTÜRK

NUMARASI……...:B211210087

ADI-SOYADI……..:ENNUR GÜRBÜZ

NUMARASI……....:B211210060

ADI-SOYADI….....:AYŞE BURCU AKSU

NUMARASI……....:B211210016

İÇİNDEKİLER:

1. Projenin Amacı ve Görev Dağılımı
2. Veri Seti Seçimi ve Tanıtımı
3. Veri Seti Kaynağı
4. Veri Seti Amacı
5. Veri Seti Yapısı
6. Veri Setindeki Problemlerin Analizi
7. Hedef Değişkeni
8. Veri Ön İşleme
9. o Eksik Veriler
10. Normalizasyon ve Standardizasyon
11. Feature Engineering
12. Feature Selection/Importance
13. Veri Analizi
14. Makine Öğrenmesi Modelleri
15. Sonuçların Karşılaştırılması
16. Sonuç ve Öneriler
17. Kaynakça

**1. Projenin Amacı ve Görev Dağılımı**

**Projenizin Amacı**

Projenin amacı, bireylerin demografik ve ekonomik özelliklerini analiz ederek yıllık gelirlerinin 50K'nın altında mı yoksa üzerinde mi olacağını makine öğrenmesi yöntemleri ile tahmin etmektir. Bu tür bir tahmin, şu gibi durumlarda kullanılabilir:

* Ekonomik analiz: Nüfusun gelir düzeyiyle ilgili genel eğilimleri anlamak.
* Sosyal politika geliştirme: Belirli özelliklere sahip bireylerin düşük gelirde kalma olasılıklarını azaltmak için sosyal projeler geliştirmek.
* Hedefli pazarlama: Belirli gelir seviyesine sahip insanları hedeflemek için.

Hedeflenen Sonuçlar

1. Model geliştirme: Makine öğrenmesi algoritmaları (örneğin, lojistik regresyon, karar ağaçları, ya da derin öğrenme modelleri) kullanılarak gelir tahmini yapan bir model oluşturulması.
2. Öngörüler: Hangi özelliklerin gelir seviyesini daha fazla etkilediğini belirlemek (örneğin, eğitim seviyesi, çalışma saati, ya da meslek gibi faktörlerin etkisi).
3. Raporlama: Modelin başarısının değerlendirilmesi ve bulguların görselleştirilerek sunulması.

**Takım Çalışması ve Görev Dağılımı**

Proje kapsamında, 3 kişilik ekibimiz olarak her aşamada aktif bir şekilde iş birliği içinde çalıştık. Veri ön işleme, model geliştirme, analiz ve raporlama gibi adımlarda görev dağılımı yapılmamış, her bir takım üyesi sürece eşit şekilde katkı sağlamıştır.

Bu yaklaşım sayesinde her bir takım üyesi projedeki tüm aşamaları öğrenme ve uygulama fırsatı bulmuş, kararlar birlikte alınarak takım çalışmasının avantajlarından yararlanılmıştır. Herkesin farklı bakış açıları projeye değer katmış ve daha kapsamlı bir sonuç elde edilmesini sağlamıştır.

**2. Veri Seti Seçimi ve Tanıtımı**

**a. Veri Seti Kaynağı**

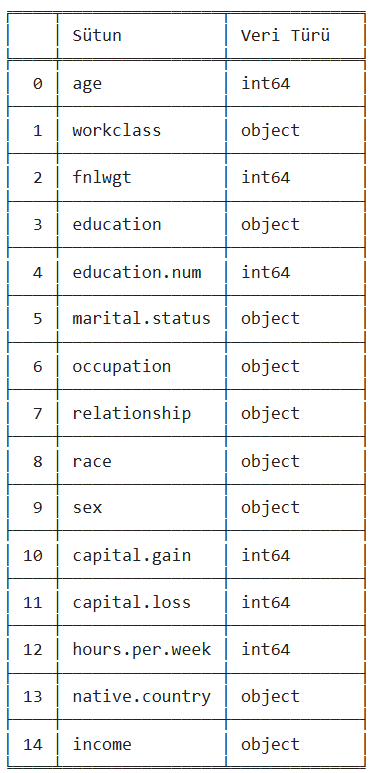
Veri seti, "Adult Income" adıyla bilinen ve <https://www.kaggle.com/datasets/uciml/adult-census-income/data> kaynağından temin edilen bir açık veri setimizdir. Bu veri seti, ABD nüfusunun demografik bilgilerini ve yıllık gelirini tahmin etmeye yönelik çeşitli özellikler içermektedir.

**b. Veri Seti Amacı**

Veri setimizin amacı, bireylerin demografik özelliklerine (yaş, eğitim, medeni durum, iş durumu vb.) dayanarak yıllık gelirlerinin 50.000 USD'nin altında mı yoksa üzerinde mi olduğunu tahmin etmektir. Veri seti, gelir tahmini yapmak için çeşitli makine öğrenmesi modelleri ile analiz edilmiştir.

**c. Veri Seti Yapısı**

Veri seti, 32.561 satır ve 15 sütun içermektedir. Her satır, bir bireyin demografik bilgilerini ve yıllık gelirini temsil etmektedir. Veri setindeki sütunlar şunlardır:



**d. Veri Setindeki Problemlerin Analizi**

Eksik Veriler: Bazı sütunlarda (özellikle workclass, occupation, native.country) eksik veriler bulunuyor. Bu verilerin yerine uygun stratejilerle (örneğin, mod değeri ile doldurma veya veri silme) müdahale edilmiştir.

Kategorik Değerler: Bazı kategorik değişkenler, "?" gibi belirsiz değerler içeriyor. Bu değerler, belirli bir kategoriye atanmış veya uygun şekilde işlenmiştir.

**e. Hedef Değişkeni**

Bu projede, income değişkeni hedef değişken olarak seçilmiştir. Bu sütun, bireyin yıllık gelirinin 50K'dan az mı yoksa fazla mı olduğunu belirtmektedir. Bu iki sınıf, "<=50K" ve ">50K" olarak etiketlenmiştir. Makine öğrenmesi modelleri, bu hedef değişkenin tahmin edilmesini amaçlamaktadır.

**3. Veri Ön İşleme**

1. **Eksik Verilerin İşlenmesi:**

Veri setinde bazı eksik değerler bulundu. Bu eksik verilerin işlenmesi için aşağıdaki adımlar uygulanmıştır:

* **Eksik Verilerin Tespiti:** Veri setindeki '?' karakterleri, eksik verilerin bir göstergesi olarak kullanılıyordu. Bu karakterler, pd.NA ile değiştirilmiş ve ardından np.nan ile eşleştirilmiştir.
* **Sayısal Verilerde Eksik Değerlerin Doldurulması:** Sayısal sütunlardaki eksik veriler, sütunların ortalama değeri ile doldurulmuştur. Bu işlem, sayısal verilerin analizini daha sağlıklı hale getirmek amacıyla gerçekleştirilmiştir. Bu süreçte, eksik değerlerin doldurulmasında SimpleImputer kütüphanesi kullanılmıştır. SimpleImputer, sayısal sütunlar için strateji olarak "mean" (ortalama) seçeneği ile yapılandırılmıştır.
* **Kategorik Verilerde Eksik Değerlerin Doldurulması:** Kategorik verilerde eksik olan değerler, sütunlardaki en sık görülen değer (mod) ile doldurulmuştur. Kategorik verilerin temsilinin bozulmamasını sağlamak için bu yöntem uygulanmıştır. Bu işlem sırasında da **SimpleImputer** kullanılmış ve strateji olarak "most\_frequent" seçeneği tercih edilmiştir.

1. **Normalizasyon ve Standardizasyon:**

Veri setindeki sayısal sütunlar normalizasyon veya standardizasyon işlemlerine tabi tutulmuştur. Bu adım, modelin daha hızlı öğrenmesini ve daha iyi performans göstermesini sağlamak için önemlidir. Normalizasyon işlemi için MinMaxScaler veya StandardScaler kullanılabilir. Ancak, bu adımda hangi scaler’ın kullanıldığı kodda görünmemektedir, ancak genellikle her sayısal sütun için bu adım uygulanır.

1. **Feature Engineering:**

Veri setindeki bazı yeni özellikler oluşturulmuştur:

* **Yaş Gruplarının Oluşturulması:** age (yaş) sütunu kullanılarak, kişilerin yaşlarına göre gruplar oluşturulmuştur. Bu gruplar şu şekilde belirlenmiştir:
  + 0-18 yaş
  + 19-25 yaş
  + 26-35 yaş
  + 36-45 yaş
  + 46-55 yaş
  + 56-65 yaş
  + 66-75 yaş
  + 76+ yaş

Bu grup, modelin yaş ile ilgili daha anlamlı öğrenmeler yapmasını sağlamaktadır.

* **Çalışma Durumu (Work Status) Özelliği:** hours.per.week (haftalık çalışma saati) sütununa dayanarak, kişilerin "full-time" (tam zamanlı) veya "part-time" (yarı zamanlı) olarak kategorize edilmelerini sağlayacak yeni bir özellik oluşturulmuştur. Bu özellik, haftalık çalışma saati 40 saatin altında ise "part-time", 40 saatin üzerinde ise "full-time" olarak adlandırılmaktadır.
* **Eğitim Durumu:** education sütunundan, kişinin eğitim seviyesini belirleyen bir özellik çıkarılmıştır. Bu özellik, kişilerin " high\_school\_or\_less" (lise veya daha yüksek durumu) olup olmadığını belirtmektedir. Eğer education sütunundaki değer, belirtilen listeye ('Bachelors', 'Masters', 'Doctorate', 'Prof-school', 'Assoc-acdm', 'Assoc-voc', 'Some-college', 'HS-grad') dahilse, 1 değeri atanır. Yeni oluşturulan sütunun adı (high\_school\_or\_less), hs\_grad\_and\_above olarak değiştirilir.
* Aksi durumda, 0 değeri atanır.
* **Kategorik Verilerin Sayısal Verilere Dönüştürülmesi:**

Veri setindeki kategorik sütunlar, makine öğrenmesi modellerinin bu verilerle doğru şekilde çalışabilmesi için **one-hot encoding** yöntemi ile sayısal verilere dönüştürülmüştür. Bu işlem, her kategorik değeri ayrı bir sütuna dönüştürerek verinin sayısal bir formatta işlenmesini sağlar.

Örnek olarak, workclass, education, occupation gibi kategorik sütunlar, her bir benzersiz kategori için bir sütun oluşturularak dönüştürülmüştür.

* **Bağımsız ve Bağımlı Değişkenlerin Ayrılması:**

Model eğitimine başlamadan önce, veri seti bağımsız (özellikler) ve bağımlı (hedef değişken) değişkenler olarak ayrılmıştır:

* **Bağımsız Değişkenler (X):** income dışında kalan tüm sütunlar bağımsız değişkenler olarak belirlenmiştir. Bu sütunlar, modelin tahmin yapacağı girdiler olarak kullanılacaktır.
* **Bağımlı Değişken (Y):** income sütunu, modelin tahmin etmeye çalışacağı hedef değişken olarak belirlenmiştir. Bu değişken, kişilerin gelir düzeyini (<=50K veya >50K) temsil etmektedir.

**d.Feature Selection/Importance:**

Lasso Regresyon modeli, özellik seçimi için etkili bir yöntem olarak kullanılır. Bu model, veri setindeki önemli değişkenleri belirler ve bazı özelliklerin katsayılarını sıfıra indirerek gereksiz değişkenleri dışlar. Parametre optimizasyonu için GridSearchCV uygulanarak en uygun alpha değeri belirlenir. Eğitim ve test setlerine ayrılmış verilerle modelin performansı değerlendirilir ve hata oranı (MSE) hesaplanır. Son olarak, modelin katsayıları incelenerek hangi özelliklerin önemli olduğu tespit edilir, sıfır olan katsayılar ise modelin dışında bırakılır. Bu süreç, en anlamlı değişkenleri seçerek modelin doğruluğunu artırır.

**Özet:**

Yukarıda belirtilen adımlar, veri setinin düzgün bir şekilde işlenmesini ve modelin doğru öğrenmesini sağlayacak temel veri ön işleme adımlarıdır. Eksik veriler doldurulmuş, kategorik veriler sayısal verilere dönüştürülmüş ve yeni özellikler oluşturulmuştur. Bu adımların ardından, model eğitimi için hazır bir veri seti oluşturulmuştur. Sonrasında, LassoRegresyon modeli kullanılarak önemli özelliklerin seçilmesi ve modelin eğitilmesi süreci başlatılmıştır. Lasso Regresyon, özellikle yüksek boyutlu veri setlerinde gereksiz özellikleri sıfıra indirerek modelin daha verimli ve doğru tahminler yapmasını sağlar. Bu adımların ardından, model eğitimine uygun hale getirilmiş bir veri seti oluşturulmuş ve modelin başarısını değerlendirebilmek için gerekli ön hazırlıklar tamamlanmıştır.

**4. Veri Analizi**

Veri setindeki sınıf dağılımı öncelikle kontrol edilmiştir. `income` sütununda, 24.720 örnekle 0 (<=50K) sınıfı ve 7.841 örnekle 1 (>50K) sınıfı bulunmaktadır. Ardından, veriler eğitim ve test setlerine ayrılmıştır. Eğitim verisi üzerinde SMOTE (Synthetic Minority Over-sampling Technique) dengeleme yöntemi uygulanarak, sınıf dengesizliği giderilmiş ve sınıflar yarı yarıya dengelenmiştir. Dengelemeden sonra, her iki sınıfın örnek sayıları eşitlenmiştir. Son olarak, `income` sütunundaki sayısal değerler, daha anlaşılır hale getirilmesi için `<=50K` ve `>50K` olarak yeniden adlandırılmıştır. Ardından, `income` dağılımı görselleştirilmiş ve sınıfların sıklığı bir bar grafiği ile gösterilmiştir. Kategorik değişkenler ile `income` arasındaki ilişkiyi daha iyi analiz edebilmek için, her bir kategorik değişken için `income` ile olan ilişkisi bar grafikleriyle gösterilmiştir. Son olarak, `income` sütunundaki metinsel değerler tekrar sayısal değerlere dönüştürülmüştür (<=50K → 0, >50K → 1). Bu adımın ardından, sayısal değişkenler arasındaki korelasyon hesaplanmış ve `income` ile diğer sayısal değişkenler arasındaki ilişkiyi görselleştirmek amacıyla bir korelasyon ısı haritası (heatmap) oluşturulmuştur. Bu adımlar, verinin daha derinlemesine analiz edilmesini ve önemli ilişkilerin görselleştirilmesini sağlamıştır.

1. **Korelasyon Analizi ve Isı Haritası**

Veri setindeki sayısal değişkenler arasındaki ilişkiyi incelemek ve gelir (income) değişkeniyle olan bağlantıyı analiz etmek amacıyla bir korelasyon matrisi oluşturulmuş ve bu matriks ısı haritası kullanılarak görselleştirilmiştir. Isı haritası, değişkenler arasındaki pozitif veya negatif yönlü ilişkilerin gücünü -1 ile 1 arasında bir ölçekle göstermektedir.

metin, ekran görüntüsü, kare, paralel içeren bir resim

Açıklama otomatik olarak oluşturuldu

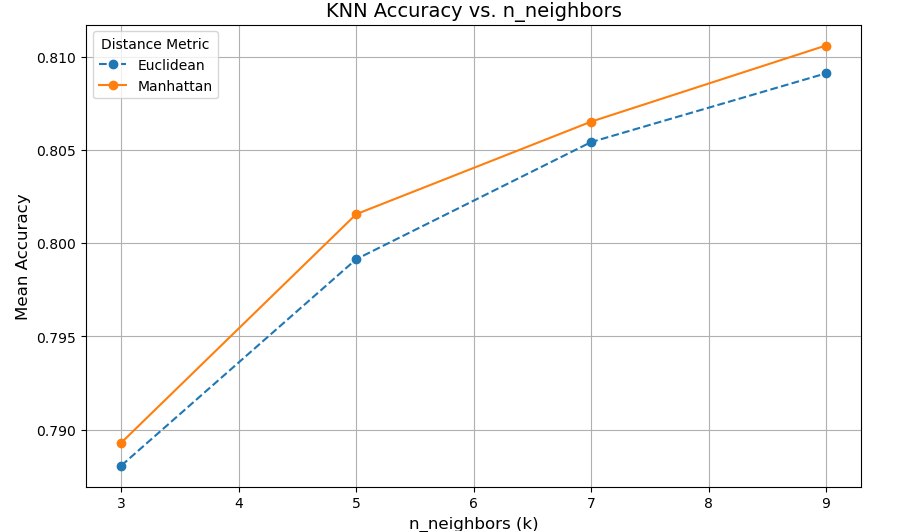
Gelir değişkeni ile diğer sayısal özellikler arasındaki korelasyonu ve veri setindeki genel ilişkileri sunmaktadır.

**5. Makine Öğrenmesi Modelleri**

**Seçilen Modeller:**

1. **Logistic Regression**: Doğrusal sınıflandırma modeli olup, özellikle ikili sınıflandırma problemlerinde başarılıdır. Genellikle gelir tahmini gibi sınıflandırma problemlerinde ilk tercih edilen modeldir. Projemizde, modelin hızı ve düşük hesaplama maliyeti sebebiyle başlangıç basamağında denendi. Model, hedef değişken ile diğer özellikler arasındaki doğrusal ilişkileri çözümlemede başarılı oldu. Ancak, doğrusal olmayan karmaşık ilişkilerde performansının sınırlı kaldığı gözlemlendi.
2. **K-Nearest Neighbors (KNN)**: Veri noktaları arasındaki mesafeye dayalı olarak sınıflandırma yapar. Çok sayıda özellik içeren veri setlerinde yüksek hesaplama gereksinimi olabilir.

Aşağıdaki grafik, farklı k değerleri ve mesafe metriklerine göre KNN algoritmasının doğruluk değerlerini göstermektedir:



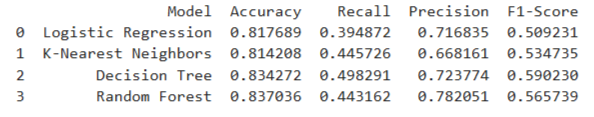
Bu grafik, KNN algoritmasında farklı komşu sayıları (k) ve mesafe metriklerinin (Manhattan) doğruluk oranı üzerindeki etkisini göstermektedir. Küçük k değerlerinde modelin daha yerel verilere odaklandığı, büyük k değerlerinde ise genelleme yeteneğinin arttığı ancak doğruluğun bir miktar düştüğü gözlemlenmiştir. En iyi sonuçlar, k=9 ve Manhattan metriği ile elde edilmiştir.

1. **Decision Tree Classifier**: Karar ağaçları, veri setindeki en iyi bölünmeleri öğrenerek sınıflandırma yapar. Projemizde gelir tahmini için demografik özelliklerin karar yollarını net bir şekilde göstermesi sayesinde hem etkili hem de görselleştirilebilir bir seçenek sundu. Model, kritik karar noktalarının (eğitim seviyesi, haftalık çalışma saatleri gibi) gelir seviyesini tahmin etmekte ne kadar etkili olduğunu ortaya koydu. Ancak, modelin aşırı uyum yapma potansiyeli, genel genelleme performansını olumsuz etkileyebilir.
2. **Random Forest Classifier**: Birden çok karar ağacının birleşiminden oluştuğundan dolayı daha genellenebilir ve güçlü bir performans sergiler. Projemizde de gelir tahmini için en önemli özelliklerin tespitinde başarılı oldu ve yanlışlı tahmin oranının düşürülmesine yardımcı oldu. Özellikle veri setindeki dengesiz sınıflar için güçlü performans göstermiştir, çünkü birçok ağaç ile dengeli bir genelleme yapabilmiştir. Ancak daha fazla hesaplama kaynağı gerektirir.

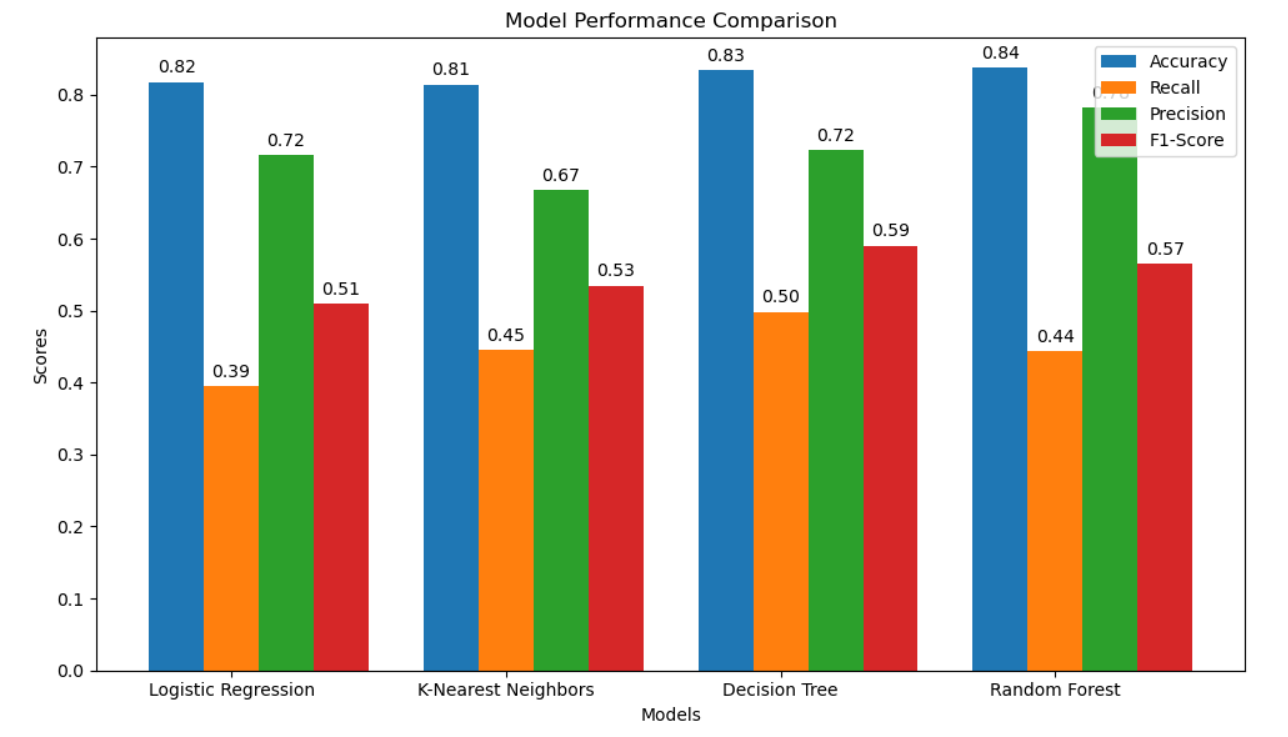
**6. Sonuçların Karşılaştırılması**

**Modellerin Sonuçları**

Aşağıda, her bir modelin test setindeki performans sonuçları verilmiştir:



**Model performans karşılaştırması:**

****

* **Modellerin Avantajları ve Dezavantajları**

1. **Logistic Regression:**

Precision ve Recall arasındaki dengesizlik, modelin sınıf dağılımındaki farklılıklardan etkilendiğini göstermektedir. Sosyal politika geliştirme gibi uygulamalarda sınıf dengesizliğinin daha iyi ele alınması gerekebilir.

**Avantajlar**

* + - Hızlı ve basit modeldir, özellikle doğrusal ilişkilerin olduğu durumlarda iyi performans gösterir.
    - Yorumlanabilirliği yüksektir.
    - Model, sınıf 0 için güçlü bir performans sergilemiş ve sınıf dengesizliğine rağmen yüksek doğruluk oranları sunmuştur.
    - Özellikle sınıf 0'ı doğru tahmin etme konusunda başarılı olması, düşük gelir grubunun belirlenmesinde etkili olabileceğini göstermektedir.

**Dezavantajlar**

* + - Karmaşık ilişkilerde düşük performans gösterebilir. Özellikle doğrusal olmayan verilerde zayıf kalabilir.
    - Sınıf 1 (yüksek gelir) üzerinde düşük bir Precision sergilemiştir, bu da pozitif sınıf tahminlerinde birçok yanlış pozitif tahmine neden olmaktadır.

1. **K-Nearest Neighbors (KNN):**

Recall değerinin görece yüksek olması, modelin gelirinin yüksek olduğunu tahmin etmesi gereken sınıflarda duyarlılığının iyi olduğunu göstermektedir. Ancak, düşük doğruluk oranı genel tahmin başarısını sınırlandırmıştır.

**Avantajlar**

* + - Basit ve esnek bir modeldir.
    - Sınıf ayırma için mesafeye dayalı bir yöntem kullandığından, karmaşık sınıflandırma problemleri için uygundur.
    - Sınıf 1 için nispeten yüksek Recall değerleri, modelin yüksek gelir sınıfını tahmin etme konusunda duyarlılığını göstermektedir.
    - Parametre optimizasyonu sonrası, k=9 ve Manhattan metriğiyle elde edilen performans, modelin gelir sınıfları arasındaki farkları anlamada faydalı olabilir.

**Dezavantajlar**

* + - Hesaplama açısından pahalıdır. Özellikle büyük veri setlerinde yavaş çalışabilir.
    - Modelin başarısı, mesafe metriklerine ve komşu sayısına bağlıdır.
    - Genel doğruluk oranı Logistic Regression ve Decision Tree'ye göre düşüktür. Bu durum, sınıf 0 üzerinde nispeten daha az doğru tahmin yapmasıyla ilişkilidir.
    - Sınıf 1 için yüksek Recall sağlasa da düşük Precision, bu sınıfta birçok yanlış pozitif tahmine yol açmaktadır.

1. **Decision Tree Classifier:**

Eğitim sırasında "max\_depth" gibi hiperparametrelerin optimize edilmesi, modelin aşırı uyum yapmasını önlemiş ve doğruluk değerlerini artırmıştır. Ancak, daha fazla parametre optimizasyonu ile performans iyileştirilebilir.

**Avantajlar**

* + - Kolayca görselleştirilebilir ve yorumlanabilir. Özellikle kategorik verilerle çalışırken etkili olabilir.
    - Sınıf 0 üzerinde güçlü bir performans sergileyerek yüksek doğruluk ve F1-Score değerleri sunmuştur.
    - Özellikle düşük gelir grubunu belirleme konusunda etkili bir modeldir.

**Dezavantajlar**

* + - Aşırı uyum (overfitting) yapma eğiliminde olabilir, özellikle derin ağaçlarda. Bu, modelin genellenebilirliğini olumsuz etkileyebilir.
    - Sınıf 1 üzerinde sınırlı bir performans sergileyerek daha düşük Recall ve Precision değerleri sunmuştur. Bu durum, özellikle yüksek gelir grubunu doğru tahmin etmekte yetersizliğini ortaya koymaktadır.

1. **Random Forest Classifier:**

Yüksek doğruluk ve F1-Score, modelin dengeli bir genelleme yaptığını göstermektedir. Ancak, daha fazla hesaplama gücü gerektirir.

**Avantajlar**

* + - Karar ağaçlarının birleşiminden oluştuğu için daha güçlü ve genellenebilir bir modeldir.
    - Daha sağlam tahminler üretebilmektedir.
    - Aşırı uyum yapma olasılığı daha düşüktür.
    - Yüksek doğruluk oranı (%78.60) ve F1-Score (%61.66), modelin sınıflar arasında dengeli bir genelleme sunduğunu göstermektedir.
    - Sınıf 0 üzerinde çok güçlü sonuçlar üretmesi, düşük gelir grubunun doğru tahmin edilmesi için güvenilir bir model olduğunu göstermektedir.

**Dezavantajlar**

* + - Daha fazla hesaplama kaynağı gerektirir ve modelin yorumlanması karar ağaçlarına göre daha zordur. Hesaplama maliyeti özellikle büyük veri setleri için dikkate alınmalı.
    - Sınıf 1 için düşük Precision (%53.99), modelin pozitif sınıf üzerinde fazla yanlış pozitif tahmin yapmasına neden olmuştur. Bu durum, pozitif sınıfın daha önemli olduğu uygulamalarda kısıt oluşturabilir.

**7. Sonuç ve Öneriler**

**Genel Sonuçlar:**

* **Random Forest** modeli, genel olarak en yüksek doğruluk ve F1-Score değerlerine sahip oldu. Bu model, her bir karar ağacının birleşiminden oluşarak daha genellenebilir ve güçlü sonuçlar sundu.
* **K-Nearest Neighbors (KNN)**, özellikle Recall açısından iyi performans gösterse de test setindeki doğruluğu biraz daha düşük kaldı.
* **Logistic Regression** ve **Decision Tree** daha düşük doğruluklar gösterdi, ancak yorumlanabilirlik açısından avantajlıydılar. **Decision Tree**'nin aşırı uyum yapmasını engellemek için max\_depth parametresi sınırlanmış olsa da yine de genelleme konusunda bazı zorluklar yaşandı.

**Model Seçimi:**

* **Random Forest**, genel olarak her metriğe göre en iyi performansı gösterdiği için en uygun model olarak seçilebilir.
* **K-Nearest Neighbors (KNN)**, daha basit ve esnek bir model olup, düşük boyutlu veri setlerinde iyi çalışabilir, ancak büyük veri setlerinde performans sorunları yaşanabilir.
* Eğer modelin yorumlanabilirliği önemliyse, **Logistic Regression** ve **Decision Tree** iyi seçenekler olabilir. Ancak, **Decision Tree** modelinde aşırı uyum riskine karşı dikkatli olunmalıdır.

**Öneriler:**

* **Random Forest** kullanılarak modelin genellenebilirliği sağlanabilir, ancak büyük veri setlerinde hesaplama kaynakları göz önünde bulundurulmalıdır.
* Model seçiminde, veri setinin büyüklüğü, işlem süresi ve modelin yorumlanabilirliği göz önünde bulundurulmalıdır.

**8.Kaynakça**