**Veri Madenciliği Proje Ödevi**

Spotify verileri bağlamında baktığımızda, bu sınıflandırma sonuçlarının ne anlama geldiğini şöyle yorumlayabiliriz:

**Diyelim ki sınıflandırma amacın şöyle olsun:**

* **0:** Bir şarkının düşük popülerlikte olması (örneğin, az dinlenen, az beğenilen)
* **1:** Yüksek popülerlikte olması (çok dinlenen, beğenilen)

**Sonuçların Spotify özelinde yorumu:**

**Logistic Regression (LR)**

* **Doğruluk ~69%**, ama pozitif sınıf (popüler şarkılar) için recall düşük (0.28).
* Yani model popüler şarkıların sadece %28'ini doğru tahmin ediyor, bu çok düşük.
* Spotify için kritik olan popüler şarkıları doğru yakalamaksa (örneğin öneri sistemleri veya pazarlama için) LR pek işe yaramayabilir.

**K-Nearest Neighbors (KNN)**

* Doğruluk %71 civarında, pozitif sınıf için recall %52, precision %63.
* KNN, popüler şarkıların yaklaşık yarısını yakalıyor; bu daha dengeli bir sonuç.
* Spotify için kullanıldığında, popüler şarkıları belirlemede daha başarılı ve öneri sisteminde daha iyi olabilir.

**Decision Tree (DT)**

* Doğruluk %64, pozitif sınıf recall ve precision %50 civarında.
* Tek başına çok güçlü değil, ancak sonuçları yorumlamak ve önemli özellikleri anlamak için kullanılabilir.
* Spotify verisinde hangi özelliklerin popülerliği etkilediğini görmek istersen DT iyi bir başlangıç olabilir.

**Random Forest (RF)**

* En yüksek doğruluk (%75), pozitif sınıf için recall %50.
* Yani popüler şarkıların yarısını yakalayabiliyor.
* Bu model Spotify için en iyi dengeyi sağlıyor hem doğru genelleme yapıyor hem de popüler şarkıları anlamaya çalışıyor.
* Ancak, recall düşük kaldığı için çok sayıda popüler şarkıyı kaçırıyor.

**Spotify kullanımına göre çıkarımlar:**

* **Popüler şarkıları kaçırmak (düşük recall)**, öneri sistemlerinde kullanıcının sevdiği şarkıların önerilmemesi anlamına gelir. Bu kullanıcı deneyimini olumsuz etkiler.
* **Daha dengeli bir model**, yani hem doğruluğu hem de pozitif sınıf recall'u yüksek model, Spotify için daha faydalı olur.
* **Random Forest ve KNN** bu dengeyi sağlıyor; ama hala recall artırılabilir.
* Özellikle popülerlik tahmininde **sınıf dengesizliği** olabilir (az sayıda çok popüler şarkı vs çok sayıda az popüler şarkı). Modelde class\_weight parametresi veya oversampling teknikleri kullanılabilir.
* Spotify'da **özellik mühendisliği çok önemli**; ses özellikleri (tempo, dans edilebilirlik, enerji), kullanıcı davranışları (dinleme sayısı, tekrar oranı), sosyal etkiler gibi değişkenler modele eklenmeli.

**Özet:**

* Spotify verisinde popülerlik tahmininde **Random Forest ve KNN** daha başarılı, ancak pozitif sınıf olan popüler şarkıların tahmininde recall'u artırmak gerek. Bu, kullanıcıların sevdiği müziğin kaçırılmaması için kritik.

**Spotify Şarkı Sınıflandırması**

**1. Veri Setinin Kaynağı**

Bu proje için “Most Streamed Spotify Songs 2024” isimli veri seti kullanıldı.  
Veri seti, Spotify’daki en çok dinlenen şarkılar hakkında çeşitli istatistiksel bilgiler içeriyor. Özellikler arasında şarkının toplam dinlenme sayısı, farklı platformlardaki playlist sayısı, Shazam sorguları gibi veriler var.

**2. Kullanılan Değişkenler (Özellikler ve Hedef)**

| **Değişken Adı** | **Açıklama** |
| --- | --- |
| Track Score | Şarkının toplam puanı |
| Spotify Streams | Spotify üzerinden toplam dinlenme |
| Spotify Playlist Count | Spotify’daki playlist sayısı |
| Spotify Playlist Reach | Playlist erişimi |
| SiriusXM Spins | SiriusXM radyoda çalma sayısı |
| Deezer Playlist Count | Deezer playlist sayısı |
| Deezer Playlist Reach | Deezer playlist erişimi |
| Amazon Playlist Count | Amazon playlist sayısı |
| Pandora Streams | Pandora üzerinden dinlenme |
| Pandora Track Stations | Pandora çalma istasyonları |
| Soundcloud Streams | Soundcloud üzerinden dinlenme |
| Shazam Counts | Shazam sorgu sayısı |
| Explicit Track (Hedef) | Şarkının popüler olup olmadığı (0=Hayır, 1=Evet) |

**3. Uygulanan Algoritmalar ve Sonuçları**

| **Model** | **Doğruluk (%)** | **Yorumlar** |
| --- | --- | --- |
| **Logistic Regression** | **~69** | Basit ve hızlı, ancak explicit şarkıları yakalamada zorlandı. |
| **K-Nearest Neighbors** | **~70.6** | Komşu bazlı model, biraz daha dengeli sonuçlar verdi. |
| **Decision Tree** | **~64.2** | Tek ağaç kullanımı sınırlı kaldı, düşük doğruluk. |
| **Random Forest** | **~75** | Birden çok ağaçla daha sağlam sonuçlar, en iyi doğruluk oranı sağlandı. |

**4. Hangi Model Neden Daha İyi Performans Gösterdi?**

* **R**andom Forest, çok sayıda karar ağacını bir araya getirerek aşırı uyum riskini azalttı ve daha genel geçer sonuçlar üretti.
* Veri setindeki karmaşıklık ve dengesizliği en iyi bu model yönetti.
* K-NN ve Logistic Regression ise daha basit algoritmalar olup, özelliklerin ölçeklenmesi sayesinde makul başarı elde ettiler.
* Decision Tree tek başına yeterince güçlü olmayabilir; bu yüzden en düşük performansı gösterdi.

**5. Kısaca Özet**

* Bu proje Spotify’daki şarkıları explicit olup olmamalarına göre sınıflandırmaya çalıştı.  
  Farklı algoritmalar denendi, sonuçlar karşılaştırıldı ve en iyi sonuç Random Forest modeliyle elde edildi. Model performanslarını iyileştirmek için dengesiz sınıf problemlerine yönelik ek çalışmalar yapılabilir.