Şarkıların Müzikal Özelliklerine Göre Şarkıları Kümeleme

Hayrettin Kaan Özsoy
Bilgisayar Mühendisliği Bölümü
TOBB Ekonomi ve Teknoloji Üniversitesi
Ankara, Türkiye
hozsoy@etu.edu.tr

***Abstract* – Bu çalışma, müzik türlerinin belirgin olmadığı ancak şarkıların müzikal özelliklerini içeren büyük bir veri seti kullanarak benzer özelliklere sahip şarkıları gruplamayı hedeflemektedir. Bu gruplama işlemi, bir denetimsiz öğrenme yöntemi olan kümeleme (clustering) modellerini kullanarak yapılacaktır. Oluşturulan bu gruplar, şarkıların müzikal benzerliklerine dayalı olarak müzik akış servislerinde kullanılan çalma listelerinin oluşturulmasına olanak sağlayacaktır. Problemin çözümü için kullanılacak farklı kümeleme modellerinden elde edilecek sonuçlar incelenerek en uygun kümeleme modeli seçilecektir.**

Dizin terimleri – clustering, unsupervised learning, music genre

I. GİRİŞ

"Müzik türü", benzer özellikteki şarkıları kategorize etmek için kullanılan bir terimdir. Müzik türleri; şarkıların enstrümantasyon, tempo, söz içerme oranı, akustiklik seviyesi gibi çeşitli özelliklerine dayalı olarak tanımlanır.

Müzik türleri; şarkıları genel olarak gruplandırmak ve analiz etmek için yararlı bir ölçüttür. Fakat; dünya üzerinde binlerce farklı müzik türünün bulunması, büyük bir müzik arşivindeki şarkıların ayırt edilmesi sırasında birtakım zorluklar yaşanmasına sebep olacaktır.

Bu çalışmada; bir müzik akış servisinin her bir kullanıcıya özgü, her birinin farklı bir müzik türündeki şarkıları içerdiği müzik listeleri oluşturmak istediğinde yaşanan karmaşa göz önüne alınmıştır. Her etiketli türe ait farklı listeler oluşturmak yerine; farklı etiketli türlere sahip olsa da benzer özellikte olan şarkıları barındıran listelerin oluşturulması gerektiği gözlemlenmiştir. Bunun için, şarkıların denetimli şekilde yapılmış tür sınıflandırmaları yerine şarkıların birtakım müzikal özelliklerine dayalı olarak denetimsiz bir şekilde oluşturulmuş gruplar kullanılacaktır. Oluşturulan grup sayısı arttıkça, aynı grup içerisinde aynı etiketli türe sahip şarkıların bulunma olasılığı artacaktır. Ancak, grup sayısı gereğinden fazla arttığında çalışmamızın genel amacından uzaklaşılmış olacaktır. Bu nedenle çalışmamızda, grup sayısının en uygun şekilde belirlenmesi hedeflenmektedir. Ayrıca, her bir grupta yer alan şarkıların olabildiğince benzer olması gerekliliği de göz ardı edilmeyecektir.

Bu çalışmada, istenen grupları oluşturmak için denetimsiz (unsupervised) öğrenmenin önemli bir kısmını içeren kümeleme (clustering) modelleri uygulanacaktır. Birçok kümeleme yaklaşımı ve bu yaklaşımlara ait modeller bulunmaktadır. Her biri farklı yaklaşımlara ait olan dört farklı model, ilgili veriler kullanılarak denenecektir. En iyi performansı gösteren ve beklentilerimizi karşılayan model seçilecektir.

Bu raporun ikinci bölümünde, veri seti üzerinde gerçekleştirilen veri analizi aşamalarından bahsedilecektir. Üçüncü bölümde, çalışmamızda kullanılan dört farklı kümeleme modelinin detaylarından bahsedilecektir. Dördüncü kısımda ise, kümeleme modelleri için küme sayısının seçilmesi için gereken aşamalar ve kümeleme sonuçlarının değerlendirilmesi için kullanılan ölçüm metriklerinden bahsedilmektedir. Beşinci kısımda, her bir model üzerinde uygulanan eğitim aşamaları sonucunda elde edilen sonuçlardan bahsedilecektir. Altıncı bölümde ise; problemimiz için en uygun çözümü sağlayacak modelin tercih edilme nedenlerinden, diğer modellerin ise tercih edilmemesine sebep olan durumlardan bahsedilecektir.

II. VERİ ANALİZİ

Çalışmada kullanılan veri yaklaşık otuz bin satırlık bir veri setinden oluşuyor. Her satırdaki değerler, bir şarkının özelliklerini içeriyor. Veri setinde, şarkıların özellikleri dışında bilgiler bulunan, model eğitimimizi olumsuz etkileyecek öznitelikler çıkartılır. “Acousticness, danceability, duration, energy, instrumentalness, liveness, loudness, speechiness, tempo, valence, popularity” olmak üzere on bir farklı öznitelik, şarkıların benzerliklerini ölçmede kullanılacaktır.

Veri setimizin boş (null) değerler içeren satırları, kullanılacak veriler arasından çıkarıldı. Null değerler, modelin performansının düşmesine yol açabilir. Birden fazla kez bulunan veriler de overfitting sorunlarına sebep olacağı için yalnızca bir kez bulunacak şekilde seçildi.

Öznitelikler arasındaki ilişkiyi gözlemlemek için Korelasyon matrisi kullanılacaktır. İki öznitelik arasındaki korelasyon değeri -1 ile 1 arasında değişir; güçlü pozitif bir ilişki varsa 1, tam tersi durumda ise -1’dir. Burada, “energy” ve “loudness” arasında özniteliğin yaklaşık 0.85 olması nedeniyle bir tanesi (energy) öznitelik seçimi yapılarak veri setinden çıkarıldı. Bu

durum, model eğitiminin karmaşıklığının ve hesaplama maliyetinin bir miktar azalmasına yardımcı oldu.

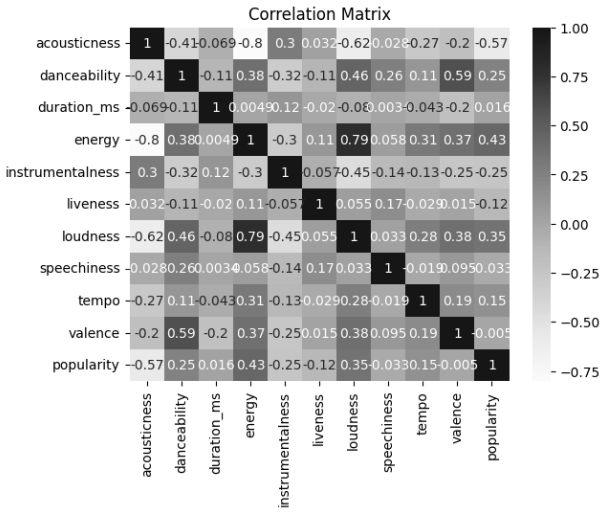

Fig. 1. Correlation Matrix

PCA (Principal Component Analysis), makine öğrenmesi ve veri analizinde çok kullanılan bir boyut azaltma tekniğidir. Fakat, bu işlem sonrasında kullanılan modellerde yanlış sınıflandırma oranının arttığı gözlemlendi. (Silhouette Score'a göre) Bu nedenle, korelasyon matrisinden yola çıkarak öznitelik seçimi yöntemi daha uygundur.

Veri normalizasyonu, farklı ölçeklerde veya aralıklarda veriler barındıran özniteliklerin olması durumunda bu verileri aynı ölçekte veya aralıkta ifade etmek için yapılan bir işlemdir. Örneğin, speechiness ve valence özniteliklerinin standard sapma ve ortalama değerleri sıfıra yakınken tempo ve popularity özniteliklerinin sapma değerleri 24, ortalama değerleri ise 100 civarındadır. Bu ölçek farklılıklarının algoritma performansını olumsuz etkilememesi için Z-Score Normalizasyonu uygulandı. Bu normalizasyon türü, veri değerlerini ortalama sıfır ve standart sapma birim olacak şekilde dönüştürmek için kullanılır. Kullanılan veri seti, etiketsiz verilerden oluştuğu için verinin dengeli olup olmaması konusunda yorum yapamayız.

III. KÜMELEME MODELLERİ

Kümeleme (Clustering), denetimsiz öğrenme (unsupervised learning) alanında kullanılan bir veri madenciliği ve veri analizi yöntemidir. Temel amacı, veri seti içerisindeki noktaları belirli ölçütlere göre gruplara (kümelere) ayırmaktır. Kümeleme işlemi, sınıflandırma işlemine (classification) benzer olmakla birlikte sınıflandırma işleminde etiketler önceden bilinirken kümeleme işleminde etiketsiz verilerle işlem yapılır.

Kümeleme algoritmalarında veri noktaları arasındaki benzerliği ölçmek için uzaklık hesaplaması yapılır. Bu hesaplamalarda Euclidean mesafesi, Manhattan mesafesi, kosinüs benzerliği gibi özel metrikler/ölçütler kullanılır. Kullandığı algoritmaya, yaklaşıma göre birçok kümleme algoritması bulunmaktadır. Bu çalışmada, dört farklı yaklaşıma ait toplam dört farklı model üzerinde çalışıldı.

K-Means Clustering:

Merkez tabanlı kümeleme (Centroid-based clustering) yaklaşımını kullanan bu model, veri noktalarını kümelemek için küme merkezlerini (centroids) kullanır, her bir veri noktasını merkezlere olan benzerliğine göre kümelere atar. Her veri noktasını kümelere atamak için uygulanan yinelemeli (iterative) bir süreçtir. Her adımda, yavaş yavaş veri noktaları benzer özelliklere göre kümelenir. Bu modelde amaç, her veri noktasının ait olması gereken doğru kümeyi belirlemek için veri noktaları ile ilgili küme merkezi arasındaki mesafelerin toplamını en aza indirmektir.

Bu modelde, veri uzayının K adet kümeye ayrılması sağlanır. Sonrasında, her bir kümenin başlangıç merkezleri (centroids) atanır. Başlangıç aşamasında merkezler, veri setinden K farklı rastgele değer seçilerek atanır. Her bir noktanın kümelere atanması, en yakın merkeze olan uzaklığına bağlı olarak gerçekleştirilir. Merkez ile nokta arasındaki uzaklığı ölçmek için "Öklid mesafesi (Euclidean distance)" metriği kullanılır.

Öklid mesafesi: İki nokta arasındaki doğrusal uzaklığı ölçerek noktalar arasındaki benzerlik hakkında bilgi sağlar. Her bir veri noktası, minimum Öklid mesafesine sahip olduğu merkezin kümesine atanır:

$$S_i^{(t)} = \left\{ x_p : \left\| x_p - m_i^{(t)} \right\|^2 \leq \left\| x_p - m_j^{(t)} \right\|^2 \ \forall j, 1 \leq j \leq k \right\}$$

Bu hesaplamalar sonucunda oluşturulan her bir kümedeki veri noktalarının ortalamaları hesaplanarak küme merkezleri yeniden belirlenir:

$$m_i^{(t+1)} = \frac{1}{\left| S_i^{(t)} \right|} \sum_{x_j \in S_i^{(t)}} x_j$$

Noktalara atanan kümeler değişmeyene kadar ve optimal merkezler elde edilene kadar yukarıdaki iki hesaplama iteratif olarak yapılır. Bu model basit ve hızlı olmasına karşın başlangıç merkezlerinin rastgele seçilmesi nedeniyle sonuçları etkileyebilir. Bu nedenle, modelin farklı başlangıç noktaları ile eğitilmesini sağlayarak algoritma performansını arttırmak mümkündür.

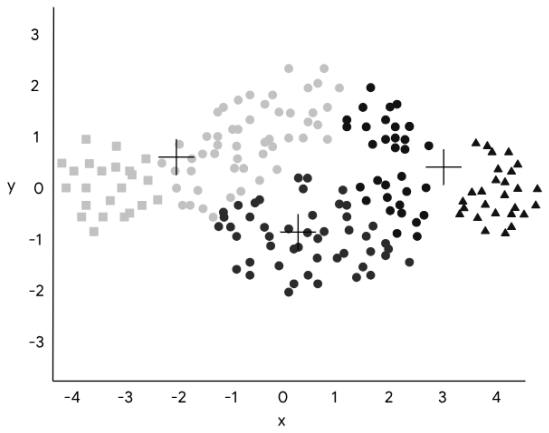

Fig. 2. K-Means Clustering sonucunda oluşmuş kümeleme modeli

K-Means Clustering modeli, hızlı ve ölçeklenebilir bir model olması sebebiyle büyük veri setleri ile çalışmaya uygundur. Basit ve anlaşılır olması, küme sayısının probleme uygun olarak belirlenebilmesi bu modelin avantajlarındandır.

Hierarchical Clustering:

Bağlantı tabanlı kümeleme (Connectivity-based clustering) yaklaşımını kullanan bu model, hiyerarşik ağaç yapısı kullanarak kümeler oluşturulmasını sağlar. "Agglomerative" ve "Divisive" olmak üzere iki farklı kümeleme stratejisi bulunur:

Agglomerative clustering, başlangıçta her bir veri noktasını tek bir küme olarak ele alır, "bottom-up" yaklaşımını kullanır. Her iterasyonda kümelerin birleştirilmesi için kullanılan üç farklı uzaklık ölçüm metriği bulunur:

Complete-linkage clustering (Maksimum ikili mesafe):

$$\max\{\, d(x,y) : x \in \mathcal{A},\, y \in \mathcal{B} \,\}$$

Single-linkage clustering (Minimum ikili mesafe):

$$\min\{\, d(x,y) : x \in \mathcal{A},\, y \in \mathcal{B} \,\}$$

Average-linkage clustering (Ortalama ikili mesafe):

$$\frac{1}{|\mathcal{A}| \cdot |\mathcal{B}|} \sum_{x \in \mathcal{A}} \sum_{y \in \mathcal{B}} d(x,y)$$

Divisive clustering, başlangıçta tüm veri noktalarını tek bir küme olarak ele alır, "top-down" yaklaşımı kullanılır. Her iterasyondaki küme bölünmesi işleminde, küme içi varyansın en aza indirilmesi ve kümeler arası varyansın en üst düzeye çıkarılması hedeflenir.

$$\frac{1}{|A \cup B|} \sum_{x \in A \cup B} \|x - \mu_{A \cup B}\|^2 = \mathrm{Var}(A \cup B)$$

Minimum varyans formülü

Bu yapının küme birleştirme ve bölme adımlarının görselleştirilmesini sağlayan "dendrogram" görselleştirmesi bulunur. Dikey eksende yer alan değerler, kümeler arasındaki mesafeyi temsil eder.

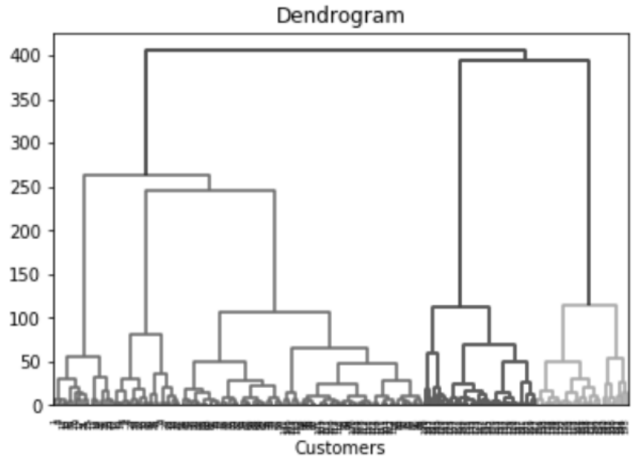

Fig. 3. Hiyerarşik kümeleme aşamalarını görselleştiren dendrogram

Hiyerarşik kümeleme modelinde, küme sayısını önceden belirtmeye gerek yoktur. Kullanılan veri hakkında yeterli bilgiye sahip olunmadığı, problemin çözümü için gereken ideal küme sayısı belirsiz olduğu durumlar için faydalıdır. Dendrogram yapısı sayesinde; kümeleme aşamalarının detaylı olarak incelenmesi, sonuçların daha iyi anlaşılması sağlanır. Fakat, büyük veri setleri için hesaplamalar maliyetli ve zaman alıcı olabilir.

Gaussian Mixture Model Clustering:

Distribution-based clustering yaklaşımını kullanan bu model, Gauss dağılımını baz alan bir denetimsiz kümeleme modelidir. Her bir kümenin bir veya daha fazla Gauss (normal) dağılım ile temsil edildiği, olasılık temelli bir modeldir. Veri noktalarının, karmaşık olasılık dağılımlarını kullanarak kümelere ayrılmasına olanak tanınır. Bu modeldeki her bir küme, bir Gauss dağılımını temsil eden bir merkeze sahiptir. Her bir veri noktası, tüm Gauss dağılımlarının veri noktasına ne kadar uygun olduğunu hesaplayan olasılık yoğunluk fonksiyonları kullanılarak değerlendirilir:

$$p(X|\,\theta) = {}_{i=i}^{n}\Sigma w_i N(X|\mu_i, \sigma_i)$$

Bu fonksiyon, her bir veri noktasının her bir kümeye ait olma olasılığının hesaplanmasını sağlar. Bu modelde, her kümeye ait ortalama ve varyans değerlerinin bulunması gerekir. Bu parametrelerin bulunması için "Beklenti Maksimizasyon (EM) algoritması" kullanılır.

Algoritmanın beklenti (expectation) aşamasında, rastgele seçilen ortalama ve varyans değerleri kullanılarak veri noktalarının her bir kümeye ait olma olasılıkları hesaplanır.

$$Q(\theta|\theta m\text{-}1)=Ep\theta m\text{-}1(X1|X2)\{\log p\theta(X1,X2)\}$$

Maksimizasyon aşamasında; maksimum olabilirlik hesaplanarak parametrelerin yeni değerleri elde edilir.

$$\theta m= \text{argmax}\theta Q(\theta|\theta m\text{-}1)$$

Bu model ile yapılan eğitim sonucunda, her bir veri noktası en yüksek olasılığa sahip olduğu kümeye atanır.

Gaussian Mixture Model Clustering, veri noktalarının karmaşık olasılık dağılımlarını kullanarak daha esnek ve genel bir kümeleme yapabilme yeteneğine sahiptir. Bu nedenle, verilerin çeşitli yapılarını ve karmaşıklıklarını ele alabilir.

DBSCAN:

"Density-Based Spatial Clustering of Applications with Noise", yoğunluk tabanlı bir kümeleme algoritmasıdır, özellikle aykırı veri tespiti ile birlikte kullanılabilir. DBSCAN, veri noktalarını yoğun bölgelerdeki gruplara (kümeler) ayırırken, düşük yoğunluktaki bölgelerdeki veri noktalarını aykırı veri olarak tanımlar. DBSCAN algoritmasında "eps" ve "minPts" olarak iki farklı parametreye ihtiyaç vardır.

eps: Bu değer, bir veri noktasının komşuluk ilişkilerini tanımlar. Yani, iki nokta arasındaki mesafe bu değerden küçük eşitse, bu noktalar komşu kabul edilir. Eps değerinin çok küçük bir değer olarak seçilmesi durumunda veri noktalarının büyük bir kısmı aykırı değer olarak kabul edilecektir. Değerin çok büyük seçilmesi durumunda ise veri noktalarının çoğu aynı kümelerde yer alacaktır.

minPts: Bu değer, eps yarıçapı içerisinde bulunan minimum komşu veri noktası sayısını belirtir. Veri kümesi ne kadar büyük olursa, MinPts değerinin de o kadar büyük seçilmesi gerekir.

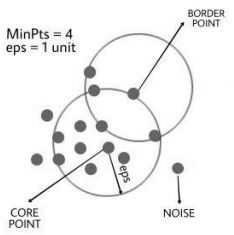

Fig. 4. DBSCAN kümeleme modeli

DBSCAN kümeleme modelinde; öncelikle rastgele veri noktaları seçilir, bu noktalar etrafında eps yarıçapına bağlı olarak bir yoğunluk bölgesi oluşturulur. Eğer seçilen noktaların eps yoğunluk bölgesinde minimum sayıda veri noktası (minPts) bulunuyorsa seçmiş olduğumuz nokta "çekirdek nokta" olarak işaretlenir. Çekirdek noktanın eps yarıçapı içerisinde kalan veri noktaları bu kümeye dahil edilir. Küme oluşturma işlemi, tüm veri noktaları için tekrarlanır. Bu nedenle, her çekirdek nokta, bir kümenin parçasıdır. Kümeler belirlendikten sonra, herhangi bir küme içerisinde yer almayan ve yeterli sayıda komşu noktaya sahip olmayan noktalar "aykırı veri" olarak kabul edilirler.

DBSCAN algoritmasında, başlangıçta küme sayısını belirtmeye gerek yoktur. Ayrıca, aykırı değerlere sahip veri setleriyle çalışmaya uygundur.

IV. KÜME SAYISINI BELİRLEME VE ÖLÇÜM METRİKLERİ

Bazı kümeleme modellerini kullanmadan önce oluşacak kümelerin sayısının belirtilmesi gerekir. Küme sayısını düşük bir değer olarak belirlemek, yetersiz öğrenmeye (underfitting) neden olurken yüksek bir değer seçimi ise aşırı öğrenmeye (overfitting) neden olacaktır. En optimal küme değerini tespit edebilecek herhangi bir yöntem bulunmasa da en iyi sonucun elde edilmesini sağlayabilecek bazı teknikler vardır: Cross-validation, Elbow method, Information Criteria, the Silhouette method, and the G-means algorithm

Elbow method, uygulanabilirliği ve anlaşılırlık açısından en yaygın kullanılan tekniklerden birisidir. Denenecek K değerleri belirlendikten sonra her K değeri için, tüm veri noktalarından merkeze olan ortalama mesafeler hesaplanır. Dikey eksende mesafe değerleri, yatay eksende ise seçilen k değerleri olacak şekilde Elbow grafiği çizilir. Ortalama mesafenin ani bir düşüş gösterdiği kısımdaki noktanın k değeri, optimal değer olarak seçilir.

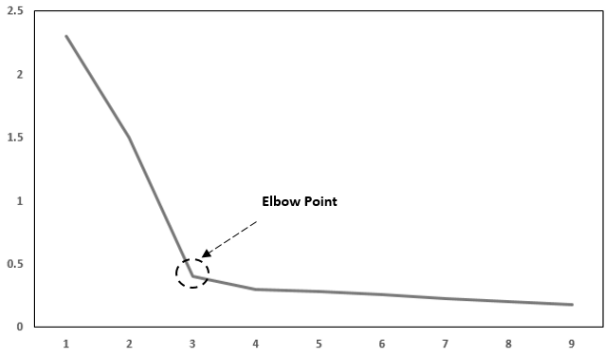

Fig. 5. Elbow method

Silhouette method, silhouette katsayısını kullanır. Bu katsayı, bir veri noktasının bir küme içinde diğer kümelere kıyasla ne kadar benzer olduğunu kontrol ederek kümelerin kalitesini ölçmeyi sağlar. Bu katsayının değeri -1 ile +1 sınırlarında yer alır. +1, veri noktasının komşu kümeden uzakta olduğunu, yani en iyi şekilde konumlandırıldığını gösterir. 0, iki komşu küme arasındaki karar sınırının açık veya çok yakın olduğunu gösterir. -1, veri noktasının yanlış kümeye atandığını gösterir.

Her bir k değeri için silhouette hesabı yapıldıktan sonra grafik çizimi yapılır, en büyük silhouette score değerine sahip küme sayısı değeri seçilir.

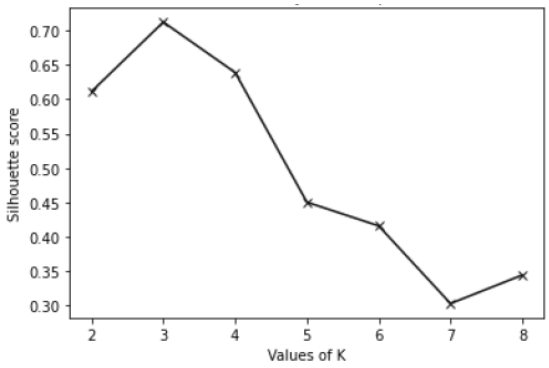

Fig. 6. Silhouette Score Analizi Grafiği

Kümeleme algoritmalarının başarısını değerlendirmek ve karşılaştırmak için kullanılan çeşitli ölçüm metrikleri vardır. Bu metrikler, bir kümeleme sonucunun ne kadar iyi olduğunu nicel olarak değerlendirmenize yardımcı olabilir:

Silhouette Score, Her veri noktasının kendi kümesi içindeki benzerliğini (homojenlik) ve en yakın diğer küme ile olan benzerliğini (ayrıklık) ölçer. Silhouette skoru, -1 ile 1 arasında bir değere sahip olur ve daha yüksek değerler daha iyi bir kümeleme sonucunu gösterir.

Adjusted Rand İndeksi (ARI), kümeleme sonuçlarının veri setindeki gerçek etiketlere (gerçek sınıflar veya kategoriler) ne kadar yakın olduğunu ölçer. Değerler -1 ile 1 arasında olabilir ve 1, mükemmel eşleşmeyi gösterir.

Davies-Bouldin İndeksi, her kümenin diğer kümelerle karşılaştırıldığında ne kadar farklı olduğunu ölçer. Düşük değerler daha iyi bir kümeleme sonucunu gösterir.

Calinski-Harabasz İndeksi, küme içi varyansın küme dışı varyansa oranını kullanarak bir kümeleme sonucunun kalitesini değerlendirir. Yüksek değerler daha iyi bir sonuçu işaret eder.

V. DENEYSEL SONUÇLAR

Bu çalışmada, K-Means Clustering, Hiearchical Clustering, Guassian Mixture Model Clustering ve DBSCAN Clustering olmak üzere dört farklı kümeleme modeli kullanıldı. Her bir modelde normalizasyon işlemi uygulanmış ve uygulanmamış veriler kullanıldı. Veri seti üç farklı şekilde ayrıldı: Yaklaşık 236 veri içeren az miktarda veri, yaklaşık 13 bin veri içeren ortalama miktarda veri ve 23 bin veri içeren çok miktarda veri. Her model için bu üç farklı veri seti kullanılarak modellerin veri miktarına bağlı olarak performans değişimi gözlendi.

K-Means clustering modelinde yapılan eğitim işlemleri sonucunda, normalize edilmiş verilerin kullanıldığı modellerin düşük performans gösterdiği ortaya çıktı. Normalize edilmiş olsun ya da olmasın, veri sayısının model performansı üzerinde büyük bir fark yaratmadığı gözlendi.

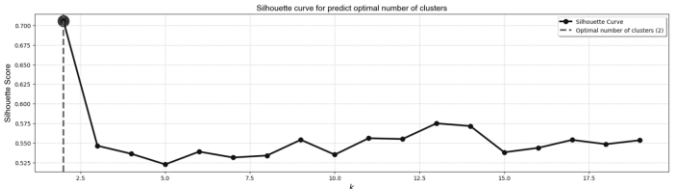

Fig. 7. Normalizasyon işlemi uygulanmamış veriler kullanılarak eğitilen K-Means clustering modeli

Oluşan kümeleri görselleştirilmesi sonucunda da, normalize edilmemiş verileri kullanarak eğitilen modelin daha başarılı olduğu görülmektedir.

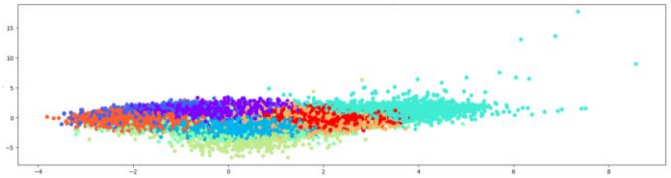

Fig. 8. Normalizasyon işlemi uygulanmış veriler kullanılarak eğitilen K-Means clustering modeli

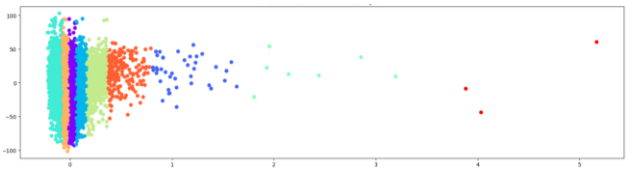

Fig. 8. Normalizasyon işlemi uygulanmamış veriler kullanılarak eğitilen K-Means clustering modeli

Genellikle, normalizasyon işlemi sonucunda elde edilen verilerin ilgili modellerde daha iyi performans göstermesi beklenir. Bu projede, kullandığımız veriler nedeniyle bu durumun tam tersi olduğu gözlenmiştir. Veri setindeki farklı özelliklerin ölçeklerinin aynı seviyeye getirilmiş olması bazı özelliklerin etkisini azaltarak modelin yanlış öğrenmesine sebep olmuş olabilir. Aynı şekilde, normalizasyon sonucunda aykırı değerler daha belirgin hale gelip modelin performansını etkiliyor olabilir. Farklı normalizasyon teknikleri denenmesine ragmen aynı durumun gözlenmesi, bu problemin veri setinden kaynaklandığını göstermektedir.

Hierarchical clustering modelinde yapılan eğitim işlemleri sonucunda elde edilen değerler de yaklaşık olarak K-Means clustering uygulanan modellerdekilerle aynı çıkmıştır. Buna ek olarak, fazla veriyle eğitim yapmanın maliyetli olduğu görülmüştür.

Gaussian Mixture Model Clustering modeliyle eğitim sonucunda sıfıra yakın Silhouette score değerleri elde edildi. Büyük veri setleri için fazla maliyetli olduğu gözlemlendi.

DBSCAN modeli ile eğitim gerçekleştirildiğinde ise, silhouette score değerinin yüksek olduğu durumlarda küme sayısının bir veya iki olduğu görüldü.

VI. SONUÇ

Bu çalışmadaki problemimiz için kullanılacak modelin küme sayısının beş ile on beş arasında bir değer alması uygundur. Onlarca müzik türüne ait birçok şarkıyı bu sayı aralığındaki bir sayıda gruba ayırarak çalma listesi oluşturursak her listede birbirine benzer müzikal özellikteki şarkılar yer alır. Bununla birlikte; belirli bir sayıda müzik listesi bulunması, müzik dinleyicilerinin benzer tarzda şarkılar içeren listeler bulmak istediğinde karmaşa yaşamasını engeller.

Belirtilmiş olan gereksinimleri en iyi karşılayan model K-Means Clustering modelidir. Belirtilen aralıklarda seçilen bir K değeri için Silhouette Score değeri 0.57-0.60 aralığında olmaktadır. Bu değer aralığı, etiketsiz verilerin kullanıldığı bir model için geçerli bir aralıktır. Raporun "Deneysel Sonuçlar" bölümünde de görüldüğü üzere (Fig. 8.), K-Means Clustering modeli sonucunda düzgün kümelenmiş bir model ortaya çıkmıştır. Ayrıca, K-Means Clustering modellerinin büyük veri setlerinden performans açısından olumsuz etkilenmemesi de bu modeli tercih etmemizdeki en büyük etkenlerden birisidir.

Hierarchical Clustering modeli de K-Means Clustering modeli ile benzer Silhouette score değerlerine sahip olsa da, özellikle büyük veri setlerinde fazla maliyete neden olduğu için tercih edilmemiştir. Gaussian Mixture Modelinde, hem Silhouette score değerimiz düşük olması hem de, özellikle büyük veri setlerinde, eğitim aşamasının fazla maliyetli olması tercih etmememize neden olmuştur. DBSCAN modelinde ise, Silhouette score değerinin yüksek olduğu modellerde istenen küme sayısı elde edilememiştir, istenen küme sayısı içeren durumlarda ise Silhouette score değerimiz negatif değerlerde çıkmıştır.

REFERENCES

[1] neptune.ai, "K-Means Clustering Explained", https://neptune.ai/blog/k-means-clustering
[2] Wikipedia, "Clustering Analysis", https://en.wikipedia.org/wiki/Cluster_analysis
[3] scikit-learn, "Clustering", https://scikit-learn.org/stable/modules/classes.html#module-sklearn.cluster
[4] GeeksforGeeks, "Gaussian Mixture Model", https://www.geeksforgeeks.org/gaussian-mixture-model/
[5] GeeksforGeeks, "DBSCAN Clustering", https://www.geeksforgeeks.org/dbscan-clustering-in-ml-density-based-clustering/
[6] GeeksforGeeks, "Elbow Method for optimal value of k in KMeans", https://www.geeksforgeeks.org/elbow-method-for-optimal-value-of-k-in-kmeans/
[7] Kaggle, "Music Recommendation System using Spotify Dataset", https://www.kaggle.com/code/vatsalmavani/music-recommendation-system-using-spotify-dataset/notebook#Clustering-Genres-with-K-Means
[8] GeeksforGeeks, "Different Types of Clustering Algorithm", https://www.geeksforgeeks.org/different-types-clustering-algorithm/
[9] Wikipedia, "Hierarchical Clustering", https://en.wikipedia.org/wiki/Hierarchical_clustering

Proje sunum linki: https://youtu.be/4E_Mx5rzH84