T.C.
SAKARYA ÜNİVERSİTESİ
BİLGİSAYAR VE BİLİŞİM BİLİMLERİ FAKÜLTESİ

ISE 401 BİLİŞİM SİSTEMLERİ MÜHENDİSLİĞİ TASARIMI

PYTHON İLE SPOTIFY VERİLERİ ANALİZ ÇALIŞMASI

B161200031 – Zeynep Nida YALIN

Bölüm	**:**	**BİLİŞİM SİSTEMLERİ MÜHENDİSLİĞİ**
Danışman	**:**	**Prof. Dr. İsmail Hakkı CEDİMOĞLU**

2020-2021 Güz Dönemi

İÇİNDEKİLER

ÖZET

Anahtar Kelimeler : Veri, Analiz, Müzik

Veri kümelerinin giderek artması ile bu verilerin etkin bir şekilde kullanılması üzerine çalışmalar yapılması ve verilerin yönetilmesi bir gereklilik haline gelmiştir. Verilerin anlamlı ve yararlı bir hale getirilebilmesi için bu verilerin işlenmesi ve bilgi haline dönüştürülmesi gerekmektedir. Günümüzde veri bilimi daha çok üretim, ekonomi, bilişim, eğitim, sağlık vs. gibi alanlarda kullanılsa da bu bilimin verinin olduğu her yerde kullanılabilmesi mümkündür.

Bu çalışma kapsamında Spotify platformundan elde edilen veri seti üzerinde veri analizi yapılmıştır. Yapılan bu çalışmada ses özelliklerinin şarkılara, müzik türlerine göre değişimi incelenmiştir.

BÖLÜM 1. GİRİŞ

1.1. Veri Analizi Nedir?

Veri analizi, ham bilgiler toplanarak, inceleme ve temizleme sonucunda asıl yararlı bilgilere ulaşma metodu olarak geçer. Gerekli verilerin ve bilgilerin toplandığı, aynı zamanda bir elemeden geçirilerek yararlı olmayan bilgi ve verilerin çıkarıldığı modelleme işlemine denir. Toplanan bu bilgiler aşamasında asıl olan sonuca ulaşmaktır. Sonuca gidecek yolda verilerin çıkarılması oldukça önemlidir. Çıkarılan verilerle birlikte nasıl yol izleneceği ve neler yapılacağı netleşir. Veri analiz sistemi dönüşüm süreci olarak da kabul edilebilir. (1)

1.1.2. Veri Analiz Süreci Nasıl İşler?

Veri analizi yaparken belli aşamalara ihtiyaç duyulur. Analiz işlemleri öncesinde bir veri akışı sağlamak gerekir. Yani konuyla alakalı veriler entegre edilmelidir. Verilerin entegre edilmesi daha düzenli bir çalışma sağlar. Bu sayede analiz yaparken hatasız ilerleme kaydedilir. Entegre süresi ne kadar iyi olursa çalışmanın vereceği sonuçta o kadar iyi olur. Entegre süresi boyunca ayıklanmamış veriler, uzman ya da uzmanlar tarafından temizlenir. Saf bir hale getirilir. Gereksiz bilgi ve veriler çıkarılır. Buradaki süreç bittikten sonra asıl analiz işlemine geçilir.

Veri ve bilgi kısmı analizin en önemli noktalarından biridir. Doğru kaynağa ulaşabilmek için bilgiler eksiksiz şekilde verilmelidir. Sonuçlarda hatayla karşılaşmamak için gerekli çalışmalar titizlikle yürütülmelidir. Ancak yine de bir sorun ortaya çıkarsa en kısa sürede çözmek gerekir. Analizler, gerekli program yardımıyla yapılsa da insan eli değdiği için ufak tefek sorunlar olabilir. Burada yapılacak yöntem ise, sorunun ana kaynağını bulmak olacaktır. Sorunu bulduktan sonra, analiz yöntemiyle çözüme kavuşturmak mümkündür. (1)

1.2. Veri Ön İşleme

Veri hazırlamak ya da veri ön analizleri, ham halde ve dağınık şekilde bulunan verileri analize hazır hale getirmek için yürütülen çalışmalardır. (2) Bu işlemler şunlardır:

1.2.1. Veri temizleme

- Eksik verilerin doldurulması, gürültülü verilerin düzeltilmesi, aykırı verilerin (outlier) temizlenmesi, uyuşmazlıkların (inconsistencies) çözümlenmesi

1.2.2. Veri entegrasyonu

- Farklı veri kaynaklarının, Veri Küplerinin veya Dosyaların entegre olması

1.2.3. Verinin küçültülmesi

- Boyut Küçültme (Dimensionality reduction)
- Sayısal Küçültme (Numerosity reduction)
- Verinin Sıkıştırılması (Data compression)

1.2.4. Verinin dönüştürülmesi ve ayrıklaştırılması

- Normalleştirme (Normalization)
- Kavram Hiyerarşisi (Concept hierarchy generation) (3)

BÖLÜM 2. KULLANILAN TEKNOLOJİLER

2.1. Anaconda Navigator

Anaconda Navigator, uygulamalarınızı başlatmamıza ve komut satırı komutlarını kullanmanıza gerek kalmadan Anaconda paketlerini, ortamları ve kanalları kolayca yönetmenizi sağlayan Anaconda'da bulunan masaüstü grafik kullanıcı arabirimidir. (4)

2.2. Jupyter Notebook

Jupyter Notebook, bir web tarayıcısı üzerinden notebook belgesi formatındaki kodları düzenlemeyi ve çalıştırmayı sağlayan bir sunucu-istemci uygulamasıdır. İlk çıktığında isim olarak IPython Notebook diye biliniyordu. Başlangıçta sadece Python'ı desteklese de zamanla gelişerek Julia, Octave, R, Haskel, Ruby gibi dilleri de desteklemeye başladı. (4)

2.3. Python Veri Bilimi Kütüphaneleri

2.3.1. NumPy

NumPy, genel amaçlı bir dizi işleme paketidir. Yüksek performanslı çok boyutlu bir dizi nesnesi ve bu dizilerle çalışmak için araçlar sağlar. Aşağıdakiler dahil olmak üzere çeşitli özellikler içerir:

- Güçlü bir N boyutlu dizi nesnesi
- Gelişmiş (yayın yapan) işlevler
- C / C ++ ve Fortran kodunu entegre etmek için araçlar
- Kullanışlı doğrusal cebir, Fourier dönüşümü ve rastgele sayı yetenekleri
- Açık bilimsel kullanımlarının yanı sıra NumPy, verimli ve çok boyutlu bir jenerik veri kabı olarak da kullanılabilir.

- NumPy'nin çok çeşitli veritabanları ile sorunsuz ve hızlı bir şekilde entegre olmasını sağlayan Numpy kullanılarak rastgele veri türleri tanımlanabilir. (5)

2.3.2. Pandas

Pandas, veri analizi için çok sayıda araç sağlayan açık kaynaklı bir Python paketidir. Paket, birçok farklı veri işleme görevi için kullanılabilen çeşitli veri yapılarıyla birlikte gelir. Ayrıca Python'da veri bilimi ve makine öğrenimi problemleri üzerinde çalışırken kullanışlı olan, veri analizi için çağrılabilecek çeşitli yöntemlere sahiptir. (6)

2.3.3. Pandas profiling

Pandas profil oluşturma, sadece birkaç satır kodla hızlı bir şekilde keşifsel veri analizi yapabileceğimiz açık kaynaklı bir Python modülüdür. Her sütun için, aşağıdaki istatistikler - sütun türü ile ilgiliyse - etkileşimli bir HTML raporunda sunulur:

- Temeller: tür, benzersiz değerler, eksik değerler
- Minimum değer, Q1, medyan, Q3, maksimum değer, aralık, çeyrekler arası aralık gibi nicelik istatistikleri
- Ortalama, mod, standart sapma, toplam, medyan mutlak sapma gibi tanımlayıcı istatistikler
- Most frequent values
- Histogram
- Yüksek korelasyonlu değişkenleri vurgulayan korelasyonlar
- Spearman, Pearson ve Kendall matrisleri (7)

2.3.4. Matplotlib

Matplotlib, çeşitli basılı kopya formatlarında ve platformlar arasında etkileşimli ortamlarda yayın kalitesinde rakamlar üreten bir Python 2D çizim kütüphanesidir.

Matplotlib, Python scripts'de, Python ve IPython shells'de, Jupyter Notebook'da, web uygulama sunucularında ve dört grafik kullanıcı arayüzü araç setinde kullanılabilir.

Matplotlib, işleri kolaylaştırmaya ve zor şeyleri mümkün kılmaya çalışır. Sadece birkaç satır kodla grafikler, histogramlar, güç spektrumları, çubuk grafikler, hata grafikleri, dağılım grafikleri vb. oluşturabilirsiniz. Basit çizim için pyplot modülü, özellikle IPython ile birleştirildiğinde MATLAB benzeri bir arayüz sağlar. (8)

2.3.5. Seaborn

Seaborn, Matplotlib tabanlı bir Python veri görselleştirme kitaplığıdır. Verilerinizi keşfetmenize ve anlamanıza yardımcı olur. Çizim fonksiyonları, tüm veri kümelerini içeren veri çerçeveleri ve diziler üzerinde çalışır. Bilgilendirici grafikler üretmek için ise gerekli anlamsal eşleştirme ve istatistiksel toplama işlemini dahili olarak gerçekleştirir. (9)

2.3.6. Dataprep

DataPrep, tek bir kütüphane içeren birkaç satır kod kullanarak verilerinizi hazırlamanıza olanak tanır. DataPrep şu amaçlarla kullanılabilir:

- Ortak veri kaynaklarından veri toplayın (dataprep.connector aracılığıyla)
- Keşif veri analizinizi yapın (dataprep.eda aracılığıyla)
- Verileri temizleyin ve standartlaştırın (dataprep.clean aracılığıyla) (10)

2.3.6.1. Dataprep.eda

Dataprep.eda paketi, kullanıcının basit API'lerle önemli özellikleri keşfetmesine izin vererek bu süreci basitleştirir. Sütun dağılımları analizi, korelasyon analizi ve eksik değerler analizi işlevlerini sağlar. (11)

BÖLÜM 3. ÇALIŞMA HAKKINDA

3.1. Projenin Amacı

Projenin amacı ilk olarak gerçek veri seti ile analiz çalışması yapmaktır. Bunun yanında hayatımızda büyük bir yeri olan müziğin değişimini ve gelişimini incelemektir.

3.2. Veri Toplama

Kaggle makine öğrenmesi, veri analizi, yapay zeka, istatistik, veri görselleştirme, matematik gibi bilimleri bir arada toplayan bir oluşumdur. İlk başta makine öğrenmesi yarışmaları sunan bir platform olarak kurulsa da son yıllarda veri setleri ve beklenen çıktısı belirlenmiş problemler kullanıcılara sunuluyor. Veri setine ve değerlendirme metriklerine erişimi olan kullanıcılar, kendi analiz ve modelleri için kullanabiliyorlar.

Bu çalışmada da veri setine kaggle aracılığı ile ulaşılmıştır.

3.3. Veri Setinin Detayları

- 27000+ sanatçı
- 160.000+ şarkı (1921-2020 yıl aralığı)
- 2600+ müzik türü
- 11 ses özelliği
- Kategorik özellikler
    - key (oktav üzerindeki tüm tuşlar, 0 ile 11 arasında değişen değerler olarak kodlanır, C'den 0 olarak başlar, C# = 1 olarak devam eder.)
    - artists (Sanatçıların listesi)
    - year (Şarkıların yayınlandığı yıl)
    - name (Şarkının ismi)

BÖLÜM 4. ANALİZ

Bu kısımda python ile seçilen veri seti üzerinde analiz ve hesaplamalar yapılacak ve grafikler oluşturulacaktır.

4.1. Kütüphanelerin import edilmesi

```
import numpy as np

import pandas as pd

import matplotlib.pyplot as plt

import seaborn as sbn

from pandas_profiling import ProfileReport

from dataprep.eda import plot, plot_correlation, plot_missing
```

Numpy ve pandas ile sayısal analizler, matplotlib ve seaborn ile görselleştirme yapılacaktır.

Pandas profiling ve dataprep ile veri setinin genel bir analizine ulaşılacaktır. Bu iki kütüphane az kod ile çok işlem yapmayı sağlıyor. Yapılan bu işlemler diğer kütüphaneler ile de yapılabilirdi fakat bu şekilde zamandan ve yerden tasarruf edilmiş oldu.

4.2. Dataframelerin oluşturulması

```
dframe = pd.read_excel("data.xlsx")


dframegenre = pd.read_excel("databygenres.xlsx")
```

Veri seti içerisinde iki adet excel tablosu bulunuyor. Bu tabloların her biri ile dataframe oluşturuldu ve bunların üzerinde işlemler yapıldı.

4.2.1. Şarkı tablosu – dframe

Çalışmadaki en kapsamlı tablodur. 1921-2020 yılları arasında piyasaya sürülmüş 160 binden fazla şarkının sanatçı bilgisi, yayınlandığı yıl bilgisi, bunun yanında ses özellikleri ve popülarite değerini içermektedir. Örnek görüntü olması adına dframe.head() ile dataframe'deki ilk 5 satır görüntülendi.

```
dframe.head()
```

	artists	name	acousticness	danceability
0	['Carl Woitschach']	Singende Bataillone 1. Teil	0.995	0.708
1	['Robert Schumann', 'Vladimir Horowitz']	Fantasiestücke, Op. 111: Più tosto lento	0.994	0.379
2	['Seweryn Goszczyński']	Chapter 1.18 - Zamek kaniowski	0.604	0.749
3	['Francisco Canaro']	Bebamos Juntos - Instrumental (Remasterizado)	0.995	0.781
4	['Frédéric Chopin', 'Vladimir Horowitz']	Polonaise-Fantaisie in A-Flat Major, Op. 61	0.990	0.210

duration_ms	energy	instrumentalness	liveness	loudness	popularity	speechiness	tempo	valence	key	year
158648	0.1950	0.563	0.1510	-12.428	0	0.0506	118.469	0.7790	10	1928
282133	0.0135	0.901	0.0763	-28.454	0	0.0462	83.972	0.0767	8	1928
104300	0.2200	0.000	0.1190	-19.924	0	0.9290	107.177	0.8800	5	1928
180760	0.1300	0.887	0.1110	-14.734	0	0.0926	108.003	0.7200	1	1928
687733	0.2040	0.908	0.0980	-16.829	1	0.0424	62.149	0.0693	11	1928

Pandas profiling raporu

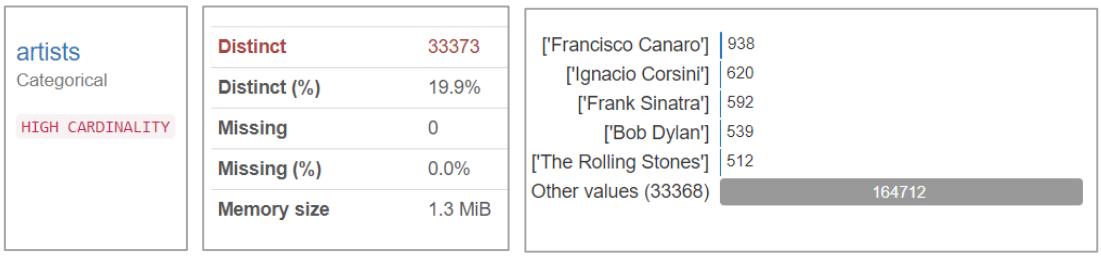
```
profile=ProfileReport(dframe,title="Data Pandas-Profiling Report")
profile.to_file("reportdata.html")
```

Pandas profiling ile rapor oluşturuldu ve html sayfası olarak kaydedildi.

Dataset statistics	
Number of variables	16
Number of observations	167913
Missing cells	0
Missing cells (%)	0.0%
Duplicate rows	0
Duplicate rows (%)	0.0%
Total size in memory	20.5 MiB
Average record size in memory	128.0 B

Oluşturulan raporda veri seti istatistiklerinin yazdığı tabloya bakılırsa analiz için kullanılan bu dataframe içerisinde 167913 satır, yani şarkı sayısı, olduğu görülmektedir. Tablo içerisinde eksik bilgi ve aynı bilgileri içeren satırların olmadığı belirtilmiştir.

Rapor içerisinde tablonun her sütunu için ayrı bir analiz oluşturulur. Bunlardan bazıları incelendi.

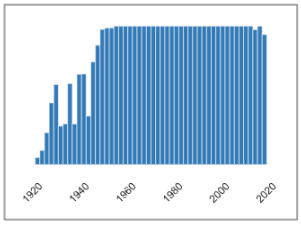


Artists sütunu için hazırlanan istatistiksel tablo incelendiğinde tablonun 33373 adet farklı sanatçıdan oluştuğu ve hiç boş değer bulunmadığı görülmektedir.

Aynı şekilde diğer tabloya bakıldığında ise en çok şarkısı bulunan 5 sanatçı listelenmiştir. Buna göre tabloda en çok şarkısı bulunan sanatçı 938 şarkı ile Francisco Canaro'dur.

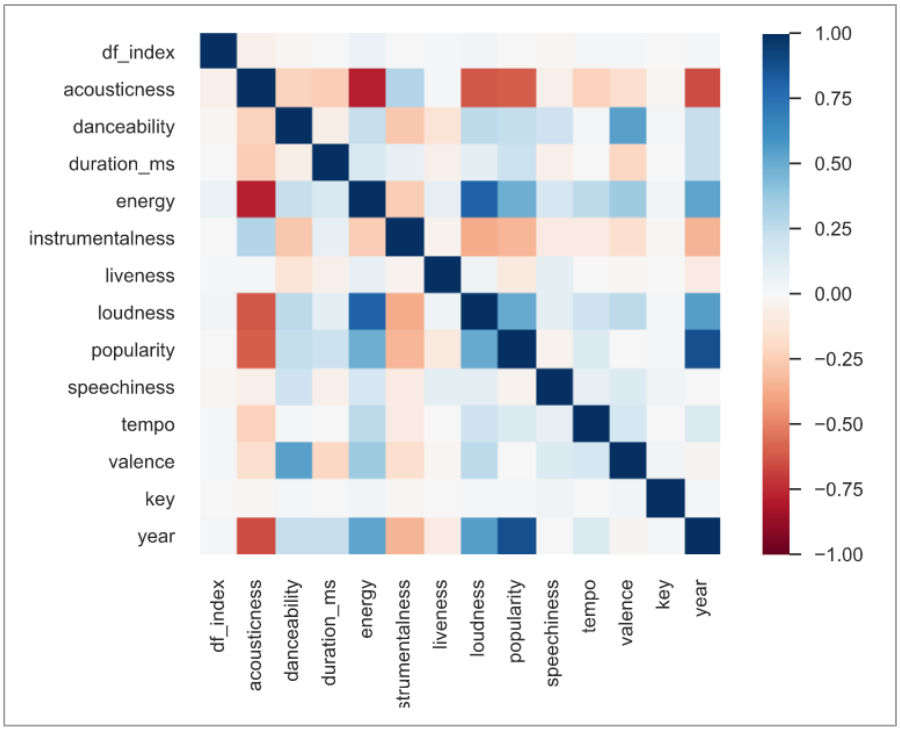
year
Real number (ℝ≥0)

Distinct	100	Mean	1977.694639
Distinct (%)	0.1%	Minimum	1921
Missing	0	Maximum	2020
Missing (%)	0.0%	Zeros	0
Infinite	0	Zeros (%)	0.0%
Infinite (%)	0.0%	Memory size	1.3 MiB



Yıl sütunu için oluşturulan istatistiksel tabloya bakıldığında sütunun numaralardan oluştuğu, 100 farklı yılın bulunduğu, 1921-2020 yılları arasında değerlerin yer aldığı, hiç boş değerin ve sıfır rakamının bulunmadığı gözlemlenmektedir.

Grafik ise bu yılların her birinde bulunan şarkı sayıları ile oluşturulmuştur. İlk yıllar için şarkı sayısı oldukça az olmakla beraber 1950 sonrasında sayının giderek arttığı görülmektedir.



Pandas profiling ile oluşturulan bu rapor korelasyon analizini de içermektedir. Değer 1'e yaklaştıkça, yani renk mavileştikçe, iki değer arasındaki ilişkinin güçlü ve doğrusal olduğu, -1'e yaklaştıkça, yani renk kırmızılaştıkça, ilişkinin negatif yönlü olduğu, 0 noktasında ise aralarında bir ilişki olmadığı bilinmektedir.

Buna göre year ile popularity değerleri arasında ve loudness ile energy değerleri arasında da doğrusal ve diğer değerlere göre daha güçlü bir ilişkinin bulunduğu gözlemlenmektedir.

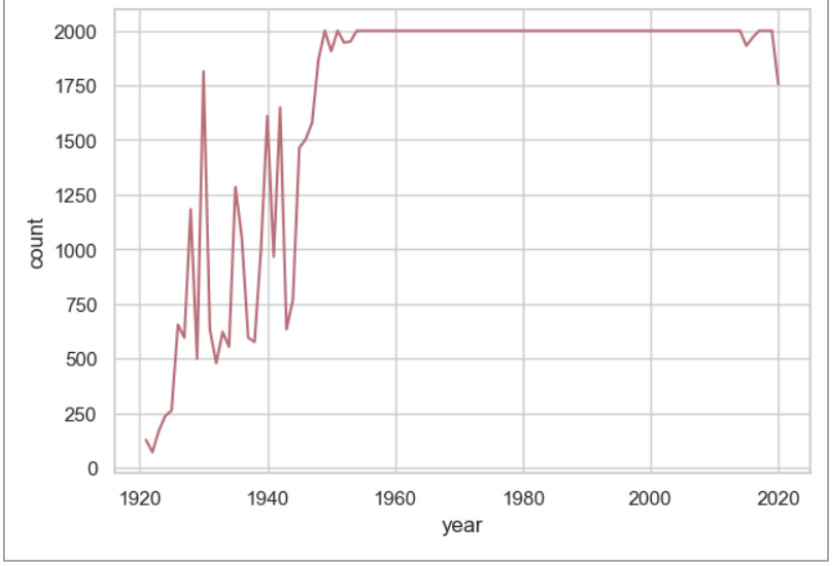
Yıllara göre şarkı sayıları

```
figure1 = plt.figure(dpi=100)

eksen1 = figure1.add_axes([0.1,0.1,0.9,0.9])

eksen1.plot(dframe[["name"] + ["year"]].groupby("year").count(),color= "#B76E79")
eksen1.set_xlabel("year")
eksen1.set_ylabel("count")
```

```
Text(0, 0.5, 'count')
```

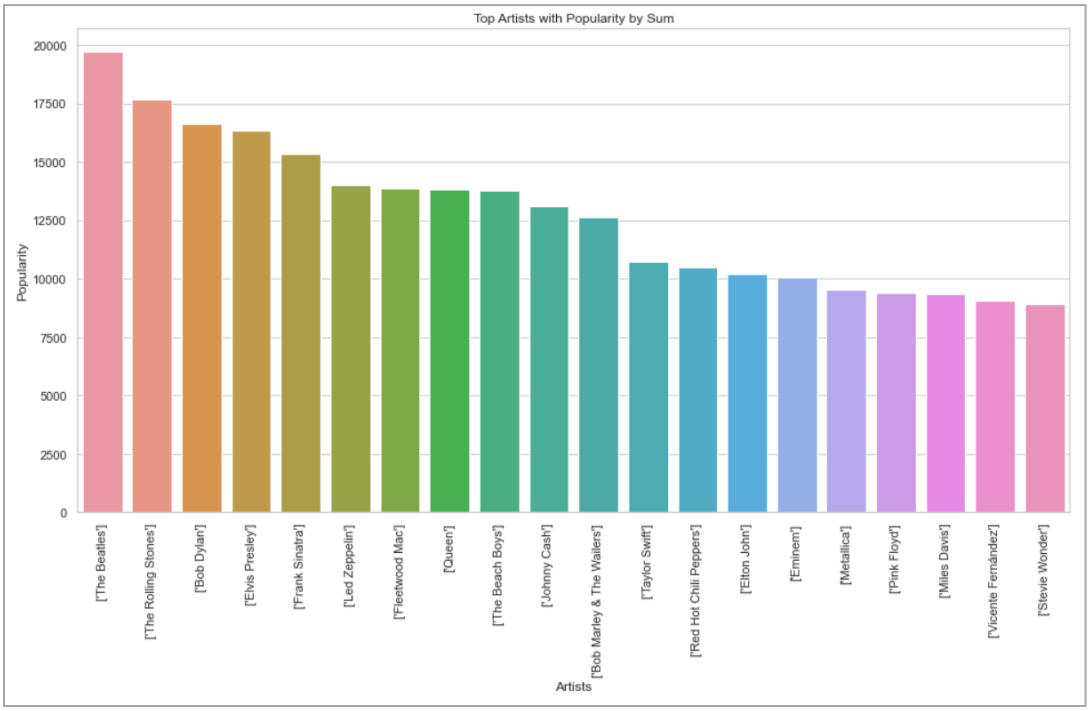


Pandas profiling raporunda da bulunan bu grafik, daha geniş ve detaylı bir şekilde incelendi. Öncelikle grafik çizimi için boyut belirlendi. Tablodaki name ve year sütunları kullanılarak year değerleri üzerinde gruplandırma yapılarak name değerleri sayıldı ve matplotlib kütüphanesinin çizdirme fonksiyonu kullanılarak bu grafik oluşturuldu.

Grafiğe bakıldığında ilk yıllarda şarkı sayılarının 250'nin altında olduğu, 1950'li yıllardan sonra giderek arttığı ve uzun bir süre 2000'lerde devamlılık sağladığı gözlemlenmektedir.

En popüler sanatçılar

```
plt.figure(figsize=(16, 8))
sbn.set(style="whitegrid")
x = dframe.groupby("artists")["popularity"].sum().sort_values(ascending=False).head(20)
ax = sbn.barplot(x.index, x)
ax.set_title('Top Artists with Popularity by Sum')
ax.set_ylabel('Popularity')
ax.set_xlabel('Artists')
plt.xticks(rotation = 90)
```

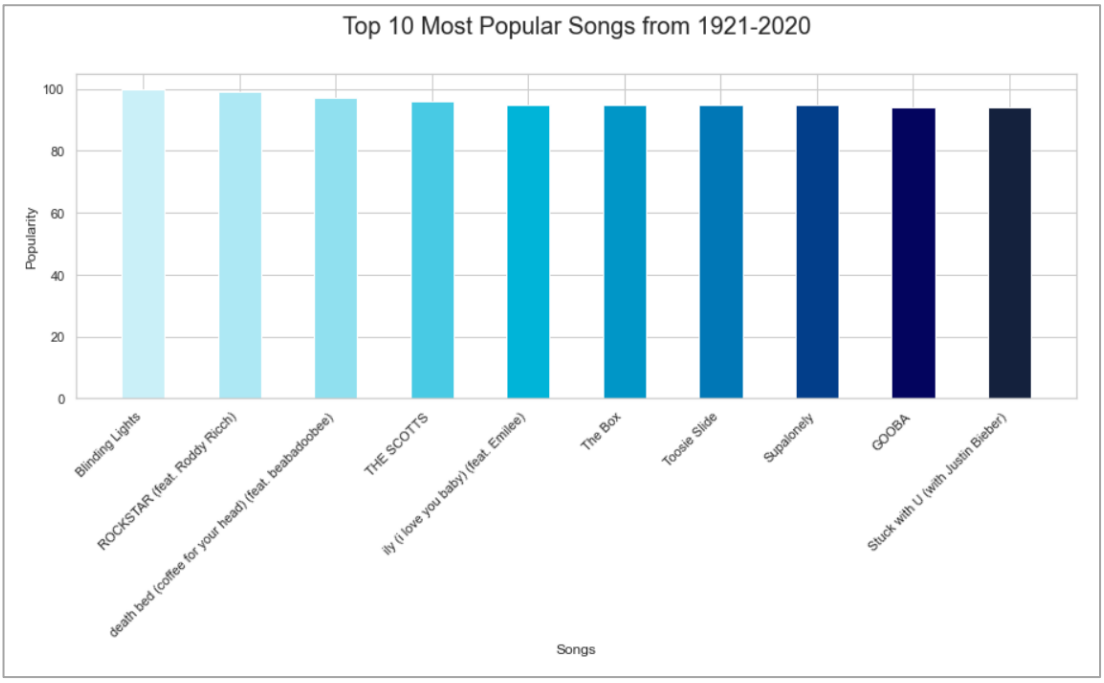


Dataframe artists sütunu ile gruplandırılarak her sanatçının tabloda bulunan şarkılarının popularity değerlerinin toplamı ile en popülerden en bilinmeyene sıralanmış ve bunların arasından ilk 20 sanatçı seçilerek ve seaborn kütüphanesinin bir fonksiyonu ile grafik oluşturulmuştur.

Grafik incelendiğinde ise tabloda bulunan 33373 farklı sanatçı arasından en popüler olanının The Beatles grubunun olduğu görülmektedir.

En popüler şarkılar

```
xyz = dframe[["artists"] + ["name"] + ["popularity"]].sort_values("popularity",ascending=False).head(10)
fig = plt.figure(figsize=(15,5))
plt.bar( xyz['name'],
         xyz['popularity'],
         width=0.45,
         color = ['#caf0f8','#ade8f4','#90e0ef','#48cae4','#00b4d8','#0096c7','#0077b6','#023e8a','#03045e','#14213d'])
plt.xticks(rotation=45,ha='right')
plt.title('Top 10 Most Popular Songs from 1921-2020',y=1.1,fontsize=20)
plt.xlabel('Songs')
plt.ylabel('Popularity')
ax.axes.get_xaxis().set_visible(True)
```

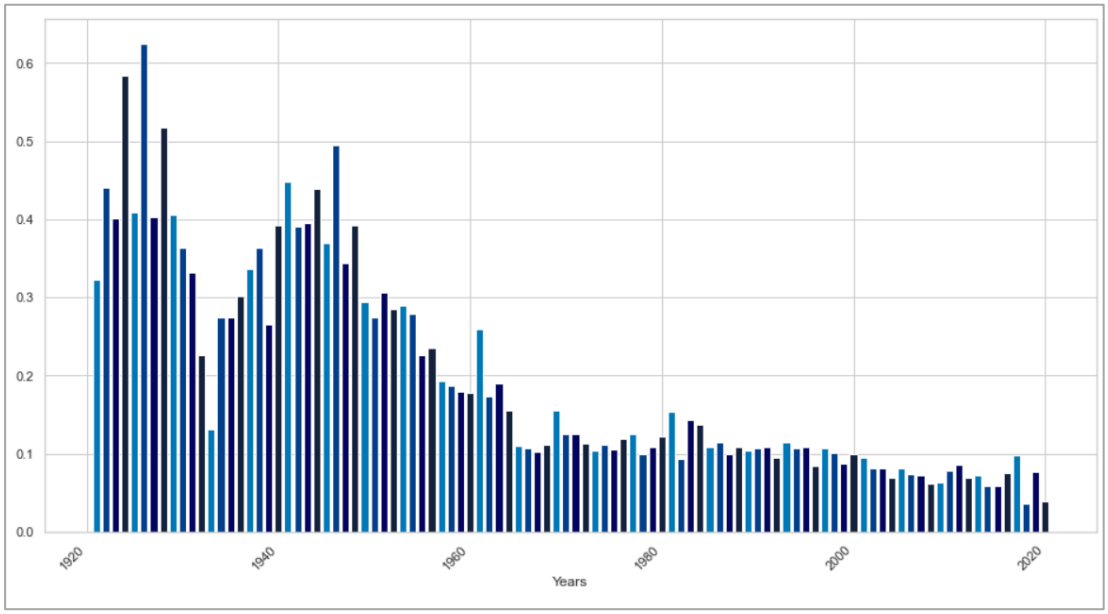


artists	name	popularity
['The Weeknd']	Blinding Lights	100
['DaBaby', 'Roddy Ricch']	ROCKSTAR (feat. Roddy Ricch)	99
['Powfu', 'beabadoobee']	death bed (coffee for your head) (feat. beabad...	97
['THE SCOTTS', 'Travis Scott', 'Kid Cudi']	THE SCOTTS	96
['Surf Mesa', 'Emilee']	ily (i love you baby) (feat. Emilee)	95
['Roddy Ricch']	The Box	95
['Drake']	Toosie Slide	95
['BENEE', 'Gus Dapperton']	Supalonely	95
['6ix9ine']	GOOBA	94
['Ariana Grande', 'Justin Bieber']	Stuck with U (with Justin Bieber)	94

Dataframe'in artists, name ve popularity sütunları ile, popularity değerlerine göre en yüksekten en alçağa kadar sıralanan ve ilk 10 değeri alan yeni bir dataframe oluşturuldu. Bu yeni dataframedeki name sütunu songs adı ile grafiğin x koordinatını oluştururken şarkıların popularity değerleri ise grafiğin y koordinatını oluşturmaktadır. Tabloda ise bu dataframe görüntülenmektedir. Oluşturulan bu tablo ve grafik incelendiğinde 167913 şarkı arasında en popüler şarkının The Weeknd – Blinding Lights olduğu görülmüştür.

Yıllara göre instrumentalness değerinin değişimi

```
instrumentalbyyear = dframe[["year","instrumentalness"]].groupby(["year"],as_index = False).mean().sort_values(by="year",ascending = True)
fig = plt.figure(figsize=(16,8))
plt.bar( instrumentalbyyear["year"],
         instrumentalbyyear["instrumentalness"],
         width=0.75,
         color = ['#0077b6','#023e8a','#03045e','#14213d'])
plt.xticks(rotation=45,ha='right')
plt.xlabel('Years')
ax.axes.get_xaxis().set_visible(True)
```



Dataframe'de bulunan year ve instrumentalness değerleri kullanılarak, bu year değerlerine göre, her yıl için tabloda bulunan şarkıların instrumentalness değerlerinin ortalaması alınıp, gruplandırılarak yeni bir dataframe oluşturulmuştur. Sonrasında ise bu dataframe kullanılarak, year değerleri x koordinatında Years ismiyle ve instrumentalness değerleri y koordinatında bulunacak şekilde matplotlib kütüphanesinin çizim fonksiyonları kullanılarak bir grafik çizilmiştir.

Grafiğe bakıldığı zaman son 100 yıl içerisinde şarkıların instrumentalness değerlerinde ciddi bir azalış gözlemlenmiştir.

4.2.2. Müzik türleri tablosu – dframegenre

Binlerce müzik türünün ve bu türlere ait ses özelliklerinin bulunduğu tablodur. Örnek görüntü olması adına dframe.head() ile dataframe'deki ilk 5 satır görüntülendi.

```
dframegenre.head()
```

	genres	acousticness	danceability	duration_ms	energy	instrumentalness	liveness	loudness	speechiness	tempo	valence	popularity	key
0	432hz	0.494780	0.299333	1.048887e+06	0.450678	0.477762	0.131000	-16.854000	0.076817	120.285667	0.221750	52.166667	5
1	a cappella	0.621532	0.577017	1.936522e+05	0.345694	0.003799	0.127087	-12.770211	0.095324	111.813230	0.453186	43.351819	11
2	abstract	0.359395	0.459500	3.430185e+05	0.487000	0.791400	0.119480	-14.092000	0.043420	124.743200	0.304990	41.500000	1
3	abstract beats	0.353347	0.694400	2.338244e+05	0.613400	0.349403	0.102453	-6.699800	0.143453	119.398400	0.634187	58.600000	10
4	abstract hip hop	0.205872	0.723132	2.490951e+05	0.645461	0.002853	0.168032	-7.216007	0.250104	112.160287	0.584392	43.804971	11

Bu tablo için dataprep kütüphanesi kullanılarak analiz yapılmış ve sonuçları değerlendirilmiştir.

Dataprep kütüphanesi raporu

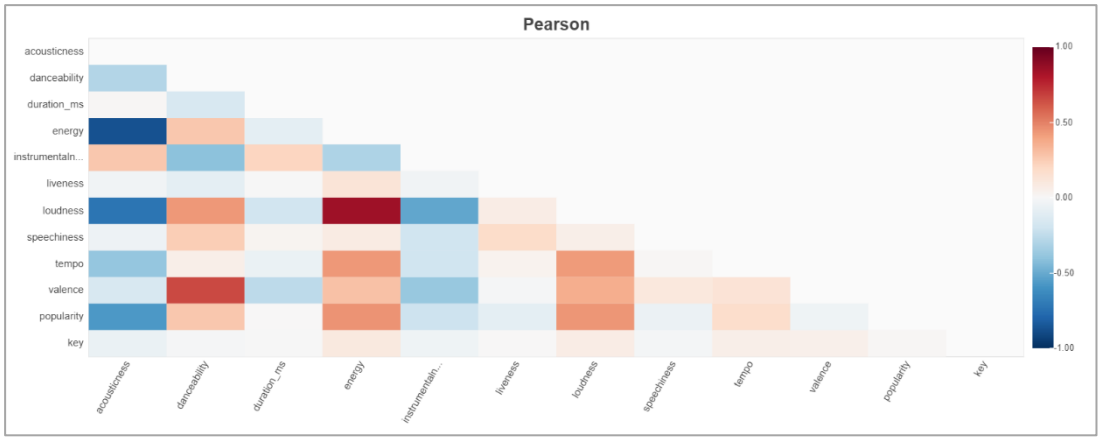
Dataprep kütüphanesi kapsamında plot fonksiyonu ile veri setinin genel özelliklerinin analizi, plot_correlation fonksiyonu ile korelasyon analizi ve plot_missing fonksiyonu ile de eksik değerlerin analizi yapılabilmektedir. Fakat bunun yerine dataprep.eda kullanılarak tüm fonksiyonların kullandığı bir rapor oluşturulması mümkündür ve kolaylık sağlamaktadır.

```
from dataprep.eda import create_report
create_report(dframegenre)
```

Dataset Statistics

Number of Variables	13
Number of Rows	2663
Missing Cells	0
Missing Cells (%)	0.0%
Duplicate Rows	0
Duplicate Rows (%)	0.0%
Total Size in Memory	432.5 KB
Average Row Size in Memory	166.3 B
Variable Types	Categorical: 1 Numerical: 12

İlk olarak raporun overview kısmında bulunan veri seti istatistikleri incelendi. Bu tabloya bakıldığında oluşturulan dataframe içerisinde 2663 adet satır, yani müzik türü olduğu görülmektedir. Aynı şekilde 13 adet sütun bulunmaktadır. Bunlardan 12 tanesi sayısal veriler içerirken 1 tanesi sayısal değer içermemektedir, bu da tablomuzun asıl konusu olan müzik türlerinin bulunduğu genres sütunudur.

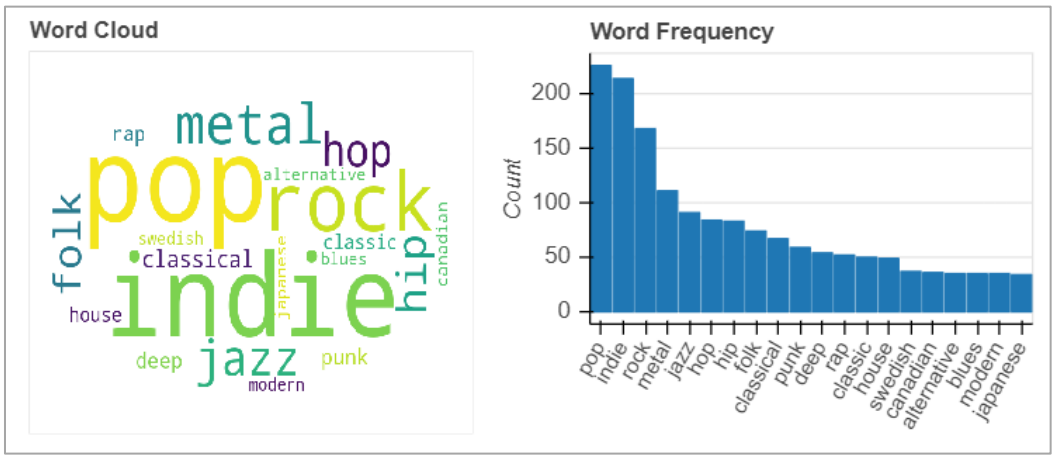


Dataprep kütüphanesi ile oluşturulan bu rapordaki korelasyon grafiğinin pandas profiling ile oluşturulan korelasyon grafiğinden farkı, aynı sütunlar için oluşturulan tam korelasyon değerlerini ve aynı değerler için hesaplanan korelasyon değerlerinin tekrarını göstermemesidir. Bu açıdan dataprep ile oluşturulan korelasyon grafiğini okuması pandas profiling ile oluşturulan grafiğe bakarak daha kolaydır.

Çalışmada kullanılan dataframe için yapılan bu korelasyon grafiğine bakıldığında energy ve loudness değerleri arasında güçlü ve pozitif doğrusal bir ilişki olduğu görülmektedir. Aynı şekilde danceability ve valence değerleri arasında da pozitif ve güçlü bir ilişki bulunmaktadır. Bunların yanında energy ve acousticness değerleri arasında negatif yönlü güçlü bir ilişki bulunmaktadır.

Rapordaki variables analizlerinin incelenmesi

Dataprep raporunda pandas profiling raporunda olduğu gibi sütunların her biri için ayrıca bir analiz kısmı bulunmaktadır. Genres sütunun analizinin olduğu kısım incelendiğinde, bu sütun sayısal veriler içermediği için plots sekmesinde kelime bulutu ve kelime sıklığı şeklinde iki adet grafik görülüyor.



Bu iki görüntü incelendiğinde mevcut müzik türleri içerisinde en çok kullanılan kelimelerin pop, indie, rock, metal şeklinde olduğu gözlemlenmiştir.

BÖLÜM 5. SONUÇ

Son yıllarda bilgi her alanda ve herkes için çok önemli bir kavram hâline gelmiştir. Ulaşılan bu veriler ancak anlaşılabilir bilgiye dönüştürülmesi ve iyi yönetilmesiyle fayda sağlanmaktadır. Bilgi hâline dönüştürülmeyen ham verinin herhangi bir faydası bulunmayacaktır.

Bunun yanında günümüzde müzik, hayatımızın bir parçası hatta günlük ihtiyaç olarak görülmektedir. Gün içerisinde dinlenilen şarkıların ne özelliklere sahip olduğunu bilmek ve bu konuda bilgi edinmek bu ihtiyacı daha anlamlı ve keyifli hale getirecektir.

Bu çalışmada verinin bilgiye dönüştürülmesi hedeflenmiştir. Bu sebeple spotify uygulamasının verileri ile oluşturulan, iki farklı tablo içeren veri seti ile veri analiz çalışması yapılmıştır. Kullanılan bu iki tablonun ilki uygulamada bulunan 160 binden fazla şarkıyı ve onların özellikleri içerirken diğeri kategorileştirilmiş müzik türleri ve bu türlerin özelliklerini içermektedir.

Yapılan analizlerin sonucunda en popüler şarkılar ve sanatçılardan müzik türlerinde en çok kullanılan kelimelere kadar oldukça çeşitli bilgiler elde edilmiştir.

KAYNAKÇA

(1). https://www.iienstitu.com/blog/veri-analizi-nedir-nasil-yapilir

(2). https://www.analytichouse.com/veri-on-isleme-nedir/

(3). https://sadievrenseker.com/wp/wp-content/uploads/2015/09/iticu_dm_3_youtube.ppt.pdf

(4). https://pythonprogramlama.com/anaconda-nedir

(5). https://www.geeksforgeeks.org/numpy-in-python-set-1-introduction/

(6). https://stackabuse.com/beginners-tutorial-on-the-pandas-python-library/

(7). https://pandas-profiling.github.io/pandas-profiling/docs/master/index.html

(8). https://matplotlib.org/3.1.1/

(9). https://seaborn.pydata.org/introduction.html

(10). https://pypi.org/project/dataprep/

(11). https://sfu-db.github.io/dataprep/user_guide/eda/introduction.html