NBA Oyuncu Verilerinin Analizi

1st Mustafa Kemal Ekim
Computer Engineering Department
Yıldız Technical University
İstanbul, Turkey
kemal.ekim@std.yildiz.edu.tr

***Abstract*—Sporda veri madenciliği son yıllarda oldukça popüler hale gelmiştir. Veri madenciliği konuları, hesaplamalı ve istatistiksel olarak bu konuları derinlemesine araştırmamıza olanak sağlamaktadır. Çeşitli yöntemler kullanılarak takımların, oyuncuların hatta personel ekiplerinin veri analizi yapılarak son derece yararlı ve kullanışlı analizler yapılabilmektedir. Bu makale, NBA oyuncularının 2021-2022 sezonundaki istatistiklerini kullanarak çıkarımlar ve belirli analizler yapılan projenin detaylarını sunmaktadır. Örneğin NBA liginde sezon sonunda en değerli oyuncu, en iyi savunma oyuncusu, en çok gelişme gösteren oyuncu gibi ödüller verilmektedir. Bu ödüller, jüriler tarafından belirlenen kriterler göz önüne alınarak puanlanmaktadır. Projemizde ise veri setindeki değerler kullanılarak verinin işlenmesi sonucu ödülü gerçekten kimin hak ettiği bulunacaktır.**

***Index Terms*—basketbol, spor, oyuncu, python**

I. Giriş

Spor analizi ve tahmini, dünyanın her yerinde popülerlik kazandı. Özellikle son yıllarda basketbol gibi daha iyi bilinen sporlar için ön plana çıktığını görüyoruz. Ulusal Basketbol Birliği veya NBA, Kuzey Amerika'nın en önde gelen erkek profesyonel basketbol ligidir. NBA'i oluşturan takımlar ve oyuncular için sayısız istatistik bulunmaktador. Verilerin işlenmesi insanlar tarafından sağlanırsa yapılan analizlerin çoğu duygulardan etkilenebilir. Toplanan NBA verileri üzerinde objektif olarak istatistiksel analizler yaparak, bu duygusal etkilerden bağımsız tahminler yapabilmek için veri madenciliğine başvururuz.
Özellikle duygudan bağımsız analiz ve karar verilmesi gereken durumlarda veri işleme konusu ön plana çıkmakta. Bu makale, NBA oyuncularının istatistiksel verileri üzerinde gerçekleştirilen çeşitli analizleri sunmakta ve bulgularını özetlemektedir. Belirli bir maçın sonucunu tahmin etmek veya en iyi savunmacıyı seçmek sınıflandırma problemleri arasına girer, oysa oynanan dakikalardan oyun başına skor gibi konuların tahmini bir lineer regresyon kullanılarak yanıtlanır.

II. İlgili İşler

- Her spor dalı için takım oluştururken sporcu seçimi önemli bir iştir . Bu seçimde sporcuların daha önceki karşılaşmalarda gösterdikleri performansın bilinmesi büyük önem taşımaktadır. Son yıllarda özellikle yurt dışında birçok spor ekibi yetenekli yeni oyuncuları keşfetmek, mevcut oyuncuların eksikliklerini ortaya çıkarmak ve rakip durumunu analiz ettirmek amacıyla istatistikçi ve analist çalıştırmaktadır.
- Atletizm kategorilerindeki sporcuların fiziksel durumlarına ve performanslarına göre antrenman sezonunun etkinliğini hesaplamak için periyodik testler yapılır. Antrenörler, bilinen atletik verileri kullanarak sezon öncesi antrenmanlarını daha verimli hale getirebilirler veya test edilen aktivitelerin sporcular üzerindeki etkisini azaltılabilir. Ayrıca, belirli organizasyonlarda oyuncularının belirli koşullar altında nasıl performans göstereceklerini de tahmin edebilirler.

III. Verilerin Alınması

Bu proje için kullanılan veriler Basketball-Reference.com ve NBA.com siterinden sağlanmıştır. Belirtilen ilk site, her oyuncunun ve takımın tüm temel istatistiklerini içerir. Bu bilgiler oyun istatistiklerinden oyuncu maaşlarına kadar uzanır. [1] Verilere .CSV formatında kolayca erişebildik. Resmi olan NBA.com web sitesi ise bize her oyuncu için topa sahip olma süresi, şut mesafesi ve defans mesafesi gibi bir dizi daha gelişmiş istatistik sağladı. [2]
Verileri aldığımız site basketbol oynamayan insanlara da hitap ettiği için istatistik sütunlarındaki yazılar açıklayıcı ve uzun bilgiler ile dolu idi.

```
1 data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 716 entries, 0 to 715
Data columns (total 28 columns):
 #   Column
---  ------
 0   FULL NAME
 1   TEAM
 2   POS
 3   AGE
 4   GP
 5   MPG
 6   MIN%Minutes PercentagePercentage of team minutes used by a player while he was on the floor
 7   USG%Usage RateUsage rate, a.k.a., usage percentage is an estimate of the percentage of team p
 8   TO%Turnover RateA metric that estimates the number of turnovers a player commits per 100 poss
 9   FTA
 10  FT%
 11  2PA
 12  2P%
 13  3PA
 14  3P%
 15  eFG%Effective Shooting PercentageWith eFG%, three-point shots made are worth 50% more than tw
 16  TS%True Shooting PercentageTrue shooting percentage is a measure of shooting efficiency that
 17  PPGPointsPoints per game.
 18  RPGReboundsRebounds per game.
 19  TRB%Total Rebound PercentageTotal rebound percentage is estimated percentage of available reb
 20  APGAssistsAssists per game.
 21  AST%Assist PercentageAssist percentage is an estimated percentage of teammate field goals a p
 22  SPGStealsSteals per game.
 23  BPGBlocksBlocks per game.
 24  TOPGTurnoversTurnovers per game.
 25  VIVersatility IndexVersatility index is a metric that measures a player's ability to produce
 26  ORTGOffensive RatingIndividual offensive rating is the number of points produced by a player
 27  DRTGDefensive RatingIndividual defensive rating estimates how many points the player allowed
dtypes: float64(21), int64(4), object(3)
```

Pandas kütüphanesinin rename fonksiyonu sayesinde bu isimleri daha anlaşılır ve görüntü kirliliği yaratmayacak şekilde düzenledik.

```
data.rename(columns= {'eFG%Effective Shooting PercentageWith eFG%
                      'TS%True Shooting PercentageTrue shooting p
                      'TRB%Total Rebound PercentageTotal rebound
                      'MIN%Minutes PercentagePercentage of team m
                      'USG%Usage RateUsage rate, a.k.a., usage pe
                      'TO%Turnover RateA metric that estimates th
                      'APGAssistsAssists per game.': 'APG',
                      'SPGStealsSteals per game.' : 'SPG',
                      'BPGBlocksBlocks per game.' : 'BPG',
                      'PPGPointsPoints per game.' : 'PPG',
                      'RPGReboundsRebounds per game.' : 'RPG',
                      'TOPGTurnoversTurnovers per game.' :'TOPG',
                      'VIVersatility IndexVersatility index is a
                      'AST%Assist PercentageAssist percentage is
                      'ORTGOffensive RatingIndividual offensive r
                      'DRTGDefensive RatingIndividual defensive r
```

```
data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 716 entries, 0 to 715
Data columns (total 28 columns):
 #   Column      Non-Null Count  Dtype
---  ------      --------------  -----
 0   FULL NAME   716 non-null    object
 1   TEAM        716 non-null    object
 2   POS         716 non-null    object
 3   AGE         716 non-null    float64
 4   GP          716 non-null    int64
 5   MPG         716 non-null    float64
 6   MIN%        716 non-null    float64
 7   USG%        716 non-null    float64
 8   TO%         703 non-null    float64
 9   FTA         716 non-null    int64
 10  FT%         716 non-null    float64
```

IV. Python

Python, genellikle karmaşık veri kümelerini düzene sokmak ve işlemek için kullanılan çeşitli avantajlara sahip, çok işlevli bir programlama dilidir. Python'un veri analizi için en iyi seçenek haline getiren bir dizi ayırt edici özelliği bulunmaktadır.

- Kolay Öğrenim: Python, basitlik ve okunabilirliğe odaklanan bir dil olmasına ek aynı zamanda veri analistleri için çok sayıda yararlı seçenek sunar. Python öğrenmeye yeni başlamış bir insan bile birkaç satır kod ile etkili çözümlere anında ulaşabilir.
- Esneklik: Python'un çok yönlülüğü, veri bilimcilerin onu popüler kılmasında etkili olmuştur. Veri modelleri oluşturma, veri setlerini işleme, makine öğrenimi algoritmaları geliştirme gibi görevleri kısa sürede tamamlamak için python kullanımı uygun olacaktır.
- Kütüphaneler: Python birçok kütüphaneye ücretsiz erişim sağlar. Bu kütüphanelerden bazıları Pandas, SciPy, scikit-learn, seaborn gibi kütüphanelerdir. Ayrıca bu kütüphanelerin sürekli geliştiğini ve ilgilendiğimiz konulara sağlam çözümler sunduğunu unutmayalım.
- Görselleştirme: İşlediğimiz verilerin rahat anlaşılabilmesi için görselleştirme çok önemlidir. Python, kullanıcılara basit kütüphaneler ile çok sayıda farklı görselleştirme seçeneği sunar. Bar ve dairesel grafikler, histogramlar, ısı haritaları gibi görselleştirme teknikleri veri analistleri tarafından sıkça kullanılan tekniklerdir.

Özetleyecek olursak, python yukarıda bahsedilen nedenlerle sektörde önemli bir yere sahiptir. Sürekli gelişmekte olması da uzun bir süre boyunca veri konularında adından söz ettireceğini bizlere göstermektedir.

V. Tahmin

Projede bir model oluşturarak veriyi 80 train ve 20 test olacak şekilde ikiye böldük ve train kısmını eğitmeye başladık. Bu eğitimi 'LinearRegression' kütüphanesi ile gerçekleştirdik. Veri setinin eğitim ve test olarak bölünmesi önemli bir adımdır çünkü doğru seçilmediği taktirde az öğrenme ya da çok öğrenme (ezberleme) gibi durumlar ortaya çıkabilir.

```
# %80 train, %20 test
from sklearn.model_selection import train_test_split
x_train,x_test,y_train,y_test=train_test_split(data[['PPG']],data[['APG']],test_size=0.2,random_state=2)
```

```
# Predicting APG from PPG
from sklearn.linear_model import LinearRegression
linear = LinearRegression()
linear.fit(x_train, y_train)
predicts = linear.predict(x_test)
```

Eğitim sonucunda iki istatistiğin birbirlerini tahmin etmesini 20'lik kısım üzerinden karşılaştırarak modelin doğruluk oranını gördük. İlişkisi yüksek olan sütunlar, örneğin oynanan dakika ve atılan sayı, daha yüksek bir doğruluk oranına sahipken korelasyonu düşük olan sütunlar, örneğin oyuncunun yaptığı sayı ve yaptığı asist, nispeten daha az bir doğruluk oranına sahip oldu. Aşağıdaki görsellerden bunu inceleyebilirsiniz.

```
# success rate
linear_score = linear.score(x_test, y_test)
print("Score: ", linear_score)

# Since points and assists are different concepts, the accuracy rate is low.
```

```
Score:  0.4947593819968187
```

```
x_train2, x_test2, y_train2, y_test2 = train_test_split(data[['MPG']], data[['PPG']]
linear2 = LinearRegression()
linear2.fit(x_train2, y_train2)
predicts = linear2.predict(x_test2)
print("Score: ", linear2.score(x_test2, y_test2))

# The more time a player has, the more chances to score.
```

```
Score:  0.738672950115288
```

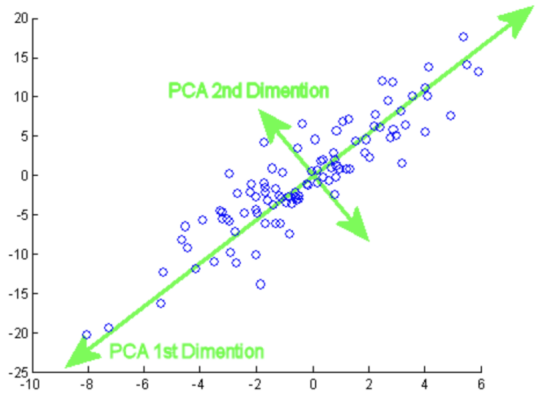
VI. Kümeleme

Kümeleme, birbirine benzeyen nesnelerin aynı kümede yer almasını sağlayarak veri setini istenen sayıda kümeye ayırma problemidir. Bu projede 'K-means clustering' adı verilen gözetimsiz öğrenme algoritmasını kullandık. Küme sayısını belirlemek algoritma açısından oldukça önemlidir. Eğer k sayısı az sayıda ise birbirine benzemeyen nesneler aynı kümeye düşecektir, eğer k sayısı çok fazla ise kümeleme işlemi anlamsız olacaktır çünkü neredeyse her nesnenin kendine ait kümesi olacaktır, bu istenmeyen bir durumdur. K sayısını seçmek için 'The Elbow Method' ve 'The Silhouette Method' gibi yöntemler kullanılabilir, bu makalede detayları işlenmeyecektir. Projede sklearn.cluster kütüphanesinin KMeans fonksiyonu kullanılarak oyuncular 5 farklı kümeye dağıtılmıştır. Sonrasında PCA (Principal Component Analysis) kullanılarak her oyuncu bir kümeye atanmıştır.

```python
#Cluster of players (k-means)
from sklearn.cluster import KMeans
model = KMeans(n_clusters=5, random_state=1)  # KMeans model (5 clusters)
efficient_cols = data._get_numeric_data().dropna(axis=1)  # get pure data
model.fit(efficient_cols) # training
labels = model.labels_
labels
```

```
array([2, 2, 1, 4, 2, 3, 4, 3, 2, 0, 4, 4, 2, 1, 4, 3, 3, 3, 4, 4, 4, 4,
       2, 4, 2, 4, 0, 0, 1, 0, 3, 3, 0, 4, 3, 1, 1, 3, 4, 4, 0, 0, 4, 2,
       2, 3, 4, 0, 4, 4, 4, 0, 3, 0, 0, 0, 0, 2, 4, 3, 3, 4, 4, 4, 1, 0,
       2, 4, 0, 4, 0, 1, 2, 2, 2, 0, 4, 2, 4, 4, 4, 4, 0, 0, 2, 1, 4, 4,
       4, 1, 4, 4, 3, 0, 3, 4, 1, 4, 3, 0, 4, 4, 2, 4, 4, 2, 4, 4, 0, 4,
       4, 4, 4, 4, 4, 4, 2, 2, 4, 3, 0, 0, 2, 0, 4, 3, 3, 4, 4, 4, 4, 0,
       0, 4, 0, 4, 3, 4, 4, 1, 0, 0, 3, 2, 4, 0, 4, 0, 4, 4, 1, 4, 4, 2,
       4, 2, 0, 4, 0, 1, 3, 2, 4, 4, 4, 4, 4, 4, 4, 4, 4, 0, 0, 2, 4, 4,
       1, 4, 1, 4, 0, 4, 0, 0, 1, 4, 4, 4, 0, 0, 4, 0, 4, 4, 3, 4, 4, 0,
       0, 0, 4, 0, 3, 1, 4, 4, 4, 4, 0, 4, 4, 4, 2, 3, 4, 4, 1, 4, 4, 4,
       0, 2, 0, 2, 1, 4, 4, 2, 4, 4, 2, 3, 3, 2, 4, 0, 0, 0, 3, 0, 2, 0,
       0, 0, 4, 0, 2, 0, 4, 2, 3, 3, 0, 0, 4, 2, 0, 3, 4, 4, 1, 2, 0, 0,
       4, 4, 2, 2, 2, 4, 4, 4, 4, 2, 1, 3, 0, 4, 0, 4, 4, 4, 4, 4, 0, 4,
       1, 0, 0, 0, 4, 4, 4, 3, 2, 4, 4, 4, 0, 4, 3, 4, 4, 4, 2, 4, 4, 3,
       0, 4, 4, 0, 1, 2, 4, 1, 0, 0, 0, 4, 4, 4, 1, 1, 4, 4, 0, 4, 4, 4,
       4, 4, 3, 4, 0, 4, 4, 1, 4, 0, 0, 4, 4, 1, 0, 4, 4, 0, 2, 4, 4, 4,
       0, 2, 4, 4, 0, 4, 4, 4, 3, 4, 4, 4, 4, 3, 0, 4, 4, 4, 4, 0, 0, 4,
       4, 4, 4, 4, 2, 3, 4, 0, 4, 0, 0, 4, 1, 4, 0, 0, 0, 2, 0, 4, 4, 3,
       0, 4, 2, 4, 0, 4, 4, 3, 3, 4, 0, 0, 0, 0, 4, 2, 3, 4, 3, 0, 2, 0,
       0, 4, 4, 3, 4, 4, 0, 1, 4, 4, 4, 4, 3, 2, 0, 0, 2, 0, 2, 4, 0, 4,
       4, 0, 3, 3, 4, 0, 1, 4, 4, 4, 3, 0, 3, 1, 1, 3, 4, 4, 4, 4, 0, 4,
       4, 1, 4, 4, 2, 4, 4, 3, 4, 4, 4, 4, 0, 1, 0, 0, 4, 0, 4, 4, 0, 2,
       4, 3, 4, 0, 4, 0, 0, 2, 0, 0, 4, 4, 3, 4, 0, 2, 4, 4, 4, 0, 4, 0,
       4, 3, 4, 3, 4, 4, 4, 4, 0, 2, 2, 0, 0, 4, 4, 4, 4, 4, 2, 2, 0, 4,
       3, 3, 4, 0, 3, 2, 0, 4, 2, 4, 2, 4, 0, 0, 0, 4, 4, 3, 4, 1, 0, 0,
       4, 4, 4, 2, 4, 0, 0, 0, 3, 0, 4, 4, 4, 2, 0, 4, 4, 0, 3, 3, 0, 3,
       4, 2, 0, 4, 4, 4, 4, 4, 4, 4, 2, 4, 4, 4, 4, 2, 4, 3, 0, 1, 4, 4, 4,
       3, 4, 4, 4, 4, 4, 3, 0, 0, 0, 4, 0, 4, 4, 4, 4, 4, 4, 4, 4, 4, 0,
       0, 2, 3, 2, 4, 0, 2, 1, 0, 0, 4, 4, 4, 0, 4, 4, 2, 4, 4, 4, 4, 3,
       0, 4, 4, 4, 0, 4, 0, 4, 2, 0, 1, 3, 0, 4, 4, 0, 1, 4, 4, 2, 3, 3,
       3, 1, 0, 1, 0, 4, 3, 0, 4, 4, 4, 4, 4, 0, 3, 4, 4, 4, 4, 0, 4, 4,
       1, 3, 2, 0, 2, 4, 0, 1, 4, 2, 0, 0, 0, 0, 4, 0, 4, 4, 4, 4, 4, 4,
       1, 4, 0, 4, 4, 4, 4, 4, 1, 0, 4, 2], dtype=int32)
```

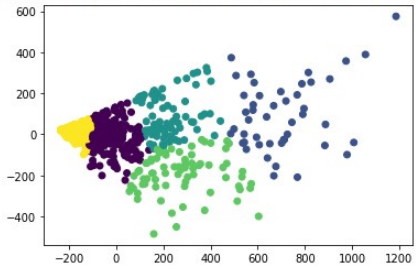
PCA'dan kısaca bahsedecek olursak, principal component analysis yani temel bileşen analizi, çok boyutlu bir uzaydaki verilerin, varyansı en üst düzeye çıkaracak şekilde daha düşük boyutlu bir alana izdüşümü bulma yöntemidir. Uzaydaki bir dizi nokta için, tüm noktaların en düşük ortalama mesafesine sahip "en uygun çizgi" seçilir. [3]



[4] Bu işlemin görselleştirilmesi ve bazı örnekleri aşağıdaki fotoğraflardan inceleyebilirsiniz.

```python
# plot clusters
from sklearn.decomposition import PCA
pca_2 = PCA(2)
plot_cols = pca_2.fit_transform(efficient_cols)
plt.scatter(x=plot_cols[:,0], y=plot_cols[:,1], c=labels)
```

```
<matplotlib.collections.PathCollection at 0x7f3f32d527d0>
```



```python
LeBron = efficient_cols.loc[ data['FULL NAME'] == 'LeBron James', :]

Jokic = efficient_cols.loc[ data['FULL NAME'] == 'Nikola Jokic', :]

Mikal = efficient_cols.loc[ data['FULL NAME'] == 'Mikal Bridges', :]
```

```python
LeBron_list = LeBron.values.tolist()
Jokic_list = Jokic.values.tolist()
Mikal_list = Mikal.values.tolist()

print("LeBron's Cluster: ", model.predict(LeBron_list))
print("Jokic's Cluster: ", model.predict(Jokic_list))
print("Mikal's Cluster: ", model.predict(Mikal_list))
```

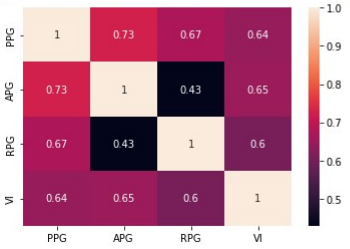
```
LeBron's Cluster:  [1]
Jokic's Cluster:  [1]
Mikal's Cluster:  [2]
```

VII. Korelasyon

Korelasyon analizi, iki değişken arasındaki ilişkiyi hesaplamak için kullanılan istatistiksel bir yöntemdir. Bir sütundaki verilerin, diğer sütundaki verilerin değişimiyle ne kadar ilişkide olduğunu korelasyon katsayısı ile inceleriz. [5] Aşağıdaki örnekte sayı, asist, ribaund ve versatility index'in korelasyonunu ve ısı haritasını görebilirsiniz.

```python
import seaborn as sns
#Heat map
# VI = measures a player's ability to produce in points, assists, and rebounds
correlation = data[['PPG', 'APG', 'RPG', 'VI']].corr()
sns.heatmap(correlation, annot=True)
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7f3f39bfd990>
```



VIII. MVP Ödülü

Sezonun en değerli oyuncusu ödülü jürilerin verdiği oylar sonucunda belirlendiği için duygu durumlarından etkilenmesi oldukça muhtemeldir. Veri madenciliği sayesinde önemli kriterler işlenerek objektif bir biçimde sezonun en değerli oyuncusu belirlenebilir. Projede kriterleri kendim seçtim ve oranladım. Bu kriterler oynanan maç +66 olmalı, en az 25 sayı

ortalaması, en az 4 asist ortalaması, en az 4 ribaund ortalaması ve şut yüzdesinin 40'tan fazla olması oldu. Bunun sonucunda 716 kişiden sadece 7 kişi MVP yarışına kalabildi.

```
mvp_candidates[['FULL NAME', 'GP', 'PPG', 'APG', 'RPG', '2P%']]
```

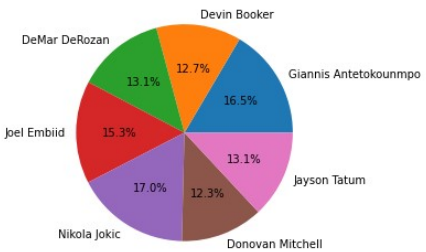
	FULL NAME	GP	PPG	APG	RPG	2P%
13	Giannis Antetokounmpo	67	29.9	5.8	11.6	0.616
64	Devin Booker	68	26.8	4.8	5.0	0.508
150	DeMar DeRozan	76	27.9	4.9	5.2	0.520
184	Joel Embiid	68	30.6	4.2	11.7	0.529
343	Nikola Jokic	74	27.1	7.9	13.8	0.652
453	Donovan Mitchell	67	25.9	5.3	4.2	0.533
623	Jayson Tatum	76	26.9	4.4	8.0	0.524

Kalan 7 adayın kriterlerdeki ortalaması alındı ve belirlenen yüzdeler ile her adayın bir puanı ortaya çıktı.

	FULL NAME	GP	PPG	APG	RPG	eFG%	MVP_Points
343	Nikola Jokic	74	27.1	7.9	13.8	0.620	12.974156
13	Giannis Antetokounmpo	67	29.9	5.8	11.6	0.582	12.573790
184	Joel Embiid	68	30.6	4.2	11.7	0.534	11.640985
150	DeMar DeRozan	76	27.9	4.9	5.2	0.521	9.988546
623	Jayson Tatum	76	26.9	4.4	8.0	0.526	9.943546
64	Devin Booker	68	26.8	4.8	5.0	0.530	9.650985
453	Donovan Mitchell	67	25.9	5.3	4.2	0.533	9.378790

Hesaplanan puanlara göre Nikola Jokic MVP ödülünün sahibi oldu. Gerçekte de ödülü kazanan o olduğundan jürilerin duygudan bağımsız puanlar verdiğini söyleyebiliriz. Aşağıdaki dairesel grafikte hangi oyuncunun ödülü yüzde kaç hak ettiğini görebilirsiniz.

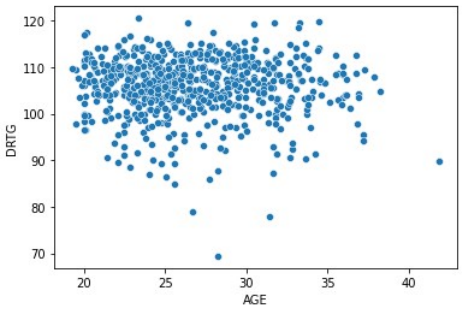
```
# Ağırlıklı kriterlere göre mvp tahminleri ve pie chart
mvp_points


fig1, ax1 = plt.subplots()
ax1.pie(mvp_candidates['MVP_Points'], labels=mvp_candidates['FULL NAME'],
ax1.axis('equal')
plt.show()
```



IX. Aykırı Verilerin Ayrıştırılması

Özetle, aykırı veri, veri kümesindeki diğer gözlemlerden önemli ölçüde farklı olan herhangi bir veri alanıdır. Aykırı verilreiçinde bulunduğu kümedeki veya toplama verilerindeki diğer verilerden farklı davranır ve hatalara neden olur. Aykırı verileri temizlerken; Kutu grafiği kullanma, 5 sayı özeti ve standart sapma ile uç değerleri belirleme gibi yöntemler kullanılır. Aykırı değerlerin çıkarılması, ortalama ile doldurulması veya silme işlemi ile biter. [6]

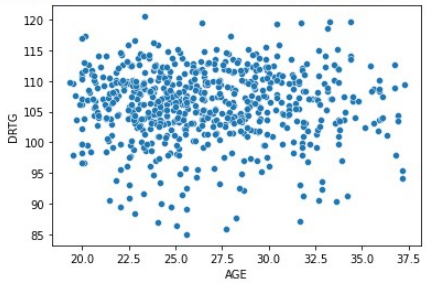
```
# MPG and PPG correlation (line graph)
ax = sns.scatterplot(x="AGE", y="DRTG", data=data)
ax.set_title("Age and Defence Correlation")
```



Yukarıdaki fotoğrafta yaş ve defansif güç arasındaki bağlantıyı noktalar şeklinde görülüyor. Aşağıdaki işlemde uç noktaları veri setimizden çıkarıp işlemlere devam ediyoruz.

```
wo_outliers = data[(data['AGE'] > 37.5) | (data['DRTG'] < 85)]
wo_outliers = pd.concat([data, wo_outliers]).drop_duplicates(keep = False)

sns.scatterplot(x="AGE", y="DRTG", data=wo_outliers)
```

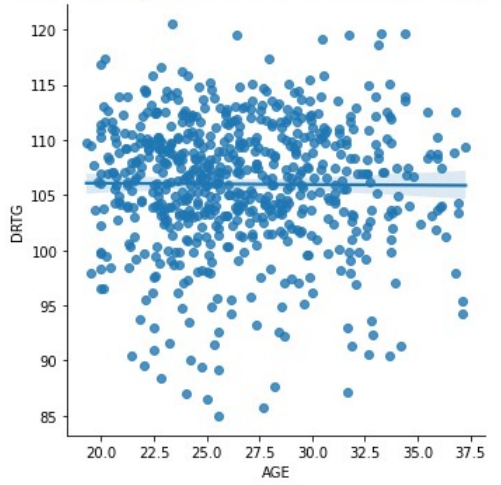
```
<matplotlib.axes._subplots.AxesSubplot at 0x7f3f2feb5650>
```



Defansif güç özelliği anormal derecede düşük olan oyuncular ve yaşı çok büyük oyuncular veri setimizden çıkarıldığında ortada daha tutarlı bir veri kaldığını görüyoruz.

```
1 sns.lmplot(x="AGE", y="DRTG", data=wo_outliers)
```

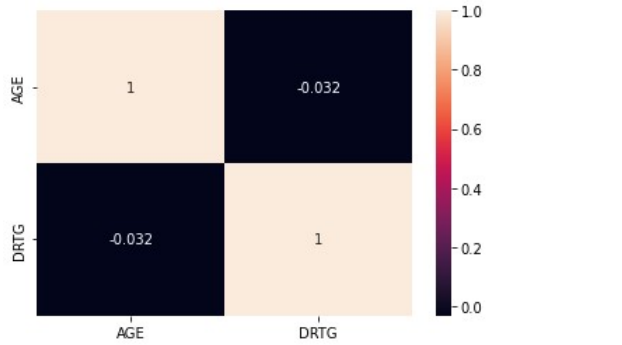
```
<seaborn.axisgrid.FacetGrid at 0x7f3f2ff68dd0>
```



Bir dağılım grafiğine en uygun çizgiyi eklemek bazı durumlarda yararlı olabilir ancak elimizde çok fazla sayıda oyuncu verisi olduğu için bu örnekte efektif olduğunu söyleyemeyiz.

```
1 correlation_age_drtg = data[['AGE', 'DRTG']].corr()
2 sns.heatmap(correlation_age_drtg, annot=True)
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7f3f30ba5b90>
```



İncelediğimiz örnekte korelasyonun eksi 0.032 seviyesinde çıktığını gördük. Bu gayet anlaşılabilir bir durum çünkü bir oyuncunun yaşı ilerledikçe defans konusunda güçlük çekmesi normaldir çünkü genç oyunculara karşı yavaş kalabilir, daha çabuk yorulabilir. Eğer örneğimizde yaş - hücum ikilisinin ilişkisi inceleseydik muhtemelen artı seviyede bir korelasyon sayısı karşımıza çıkacaktı çünkü oyuncu ne kadar fazla süre basketbol oynar veya izlerse oyunu o kadar iyi tanır ve hücumda ne yapması gerektiğini daha iyi bilir.

X. Sonuç

Günümüz dünyasında her alanda bir çok veri üretilmektedir ve artık insanların bu verileri kağıt kalem ile analiz etmesi mümkün olmamaya başlamıştır. Eğitim, ticaret, bilim, finans, sağlık ve spor gibi bir çok alanda işe yarar verileri depolama ve işleme günümüzde revaçtadır ve uzun bir süre boyunca popüler kalmaya devam edecek gibi durmaktadır.
Hazırlanan proje ve bu yazıda, NBA istatistiksel verileri üzerinde gerçekleştirilen birkaç analiz sunduk ve sonuçları özetledik. Bu makale basketbol taraftarlarına bilgileri daha iyi tanıtacak, bu arada basketbolla ilgilenmeyen insanlar için basit bir tanıtım olmaktadır. Python dili sayesinde veri setimiz üzerinde tahmin ve doğruluk oranı, kümeleme, korelasyon, aykırı veriler gibi konular projede işlenmiştir.

References

[1] https://www.basketball-reference.com/
[2] https://www.nba.com/
[3] https://scikit-learn.org/stable/modules/generated/sklearn.decomposition.PCA.html
[4] https://programmathically.com/principal-components-analysis-explained-for-dummies/
[5] https://tr.wikipedia.org/wiki/Korelasyon
[6] https://machinelearningmastery.com/how-to-use-statistics-to-identify-outliers-in-data/