

**EGE ÜNİVERSİTESİ**

**MÜHENDİSLİK FAKÜLTESİ**

**BİLGİSAYAR MÜHENDİSLİĞİ BÖLÜMÜ**

**YAPAY ZEKÂ YÖNTEMLERİ (3+0)**

**2021-2022 BAHAR YARIYILI**

**PROJE-2 DÖNEM RAPORU**

**TESLİM TARİHİ**

20/06/2022

**Konu:**

Futbol Ligleri Puan Tablosu Tahminleyen Yapay Zekâ Programı

**HAZIRLAYANLAR**

Çağatay Çetinkol, 05190000026

Anıl Muslu, 05190000045

İçindekiler

[Problemin Tanımı: 3](#_Toc106592806)

[Araştırma (Ön Çalışma): 3](#_Toc106592807)

[Kullanılan Ortam, Yöntem ve Kütüphaneler: 3](#_Toc106592808)

[Önerilen (Geliştirilen/Kullanılan) Yöntem: 3](#_Toc106592809)

[Deneysel Çalışmalar: 4](#_Toc106592810)

[Kaynak Kodlar: 10](#_Toc106592811)

[Sonuç: 11](#_Toc106592812)

[Ek1: 11](#_Toc106592813)

[Ek2: 11](#_Toc106592814)

[Ek3: 11](#_Toc106592815)

[Ek4: 12](#_Toc106592816)

[Kaynakça: 12](#_Toc106592817)

[Özdeğerlendirme Tablosu 13](#_Toc106592818)

[İş Bölümü ve Çalışma Süreleri Tablosu 14](#_Toc106592819)

# Problemin Tanımı:

Avrupa’nın 5 büyük futbol ligindeki takımların geçmiş sezonlardaki istatistiksel verilerine bakarak sonraki sezon performanslarını ve puan durumlarını tahmin eden makine öğrenmesi yazılımının yapılması.

# Araştırma (Ön Çalışma):

Öncelikle ilgi alanımız ile ilgili nasıl bir proje ve çözüm yapabileceğimiz üzerine düşündük.

Ders notlarımızdan ML ve DL konularını inceledik. İnternet üzerinden (Kaggle[[1]](https://www.kaggle.com/datasets), Google DatasetsTool[[2]](https://datasetsearch.research.google.com/), UCI Machine Learning Repository: Data Sets[[3]](https://archive.ics.uci.edu/ml/datasets.php) sitelerinden yararlanarak) konumuzla ilgili veri setlerini inceleyerek yararlanabileceğimiz veri setlerini ve verileri bulduk.

Konumuza ilişkin yazıları ve kodları araştırdık ve ilgili çalışmaları inceledik.

Fikrimizi netleştirerek problemimizi tam olarak (yukarıda tanımlandığı şekilde) tanımladık.

Veri setlerinden hangi verilerin bizim çözümümüzde işe yarar olduğunu ve hangilerinin gereksiz olduğunu belirledik. Hedef verimizi ve modelimizi hangi özniteliklere göre eğiteceğimizi kararlaştırdık.

# Kullanılan Ortam, Yöntem ve Kütüphaneler:

Python Anaconda dağıtım sistemi, Jupyter Notebook geliştirici arayüzü ortamlarını kullandık. Pandas, Numpy, Matplotlib ve scikit-learn kütüphanelerinden yararlandık. Random Forest, Yapay Sinir Ağları ve bu yöntemlerin ölçeklendirilmiş (Scaled) hallerini kullandık. Normalizasyon yöntemi olarak “Min Max Scaler” yöntemini denedik.

# Önerilen (Geliştirilen/Kullanılan) Yöntem:

Öncelikle Avrupa’nın 5 büyük ligini elde etmek ve iki farklı veri setinden yararlanacağımız için veri uyumsuzluğunu ortadan kaldırmak amacıyla veri setimizdeki Rusya Futbol Ligini çıkartıyoruz.

Modelimizin eğitimini aşırı şekilde etkileyecek olan veya etkilemeyecek olup fazlalığa sebep olacak olan sütunları verisetimizden çıkartıyoruz. Diğer veri setinden bulduğumuz daha yararlı sütunları (istatistikleri) veri setimize ekliyoruz.

2 farklı verisetini birbirine başarılı şekilde birleştirebilmek için aynı değere karşılık gelen fakat isimlendirme vs. gibi durumlarda farklılıkları bulunan değerleri belirleyip düzenleme işlemlerini yapıyoruz.

Verisetlerimizi birbiri ile birleştirerek tek bir bütün haline getiriyoruz.

Her lig için 2015-2019 yıllarına ait verileri eğitim verisi olarak kullanıyoruz. 2020 yılına ait verilerimizi test verisi olarak kullanıyoruz.

Eğittiğimiz modelimizin hesapladığı puana göre o ligdeki tüm takımları sıralıyoruz. (Modelimizi eğitirken “Supervised Learning” yöntemlerinden Yapay Sinir Ağları ve Random Forest yöntemlerinden yararlanıyoruz.)

2020 yılına ait gerçek puanlarla modelimizin hesapladığı puanları karşılaştırarak doğruluk oranlarına bakıyoruz.

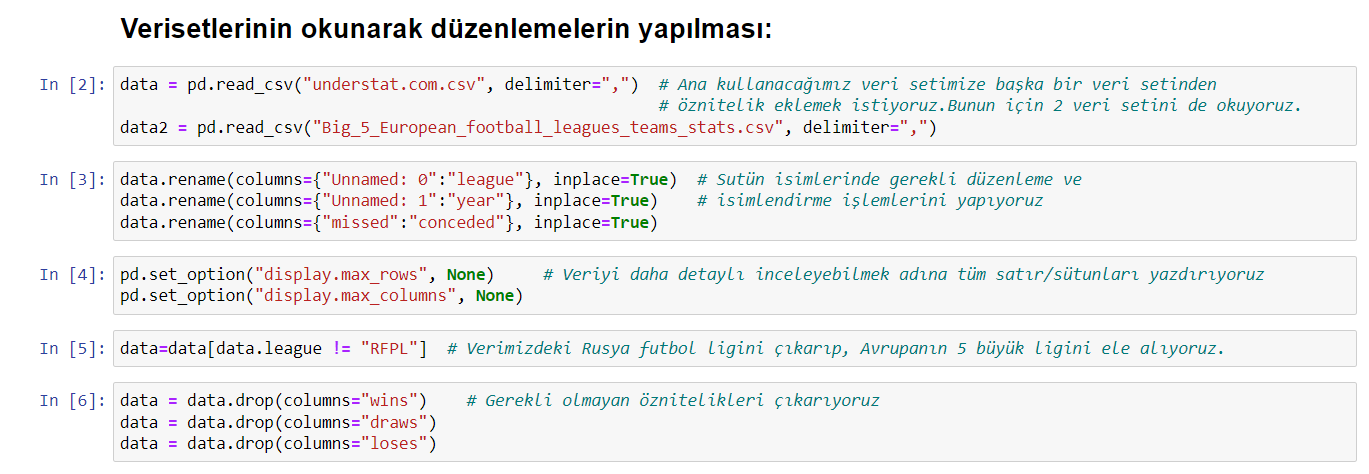
Puan skalası çok geniş olduğu için doğruluk oranları puana göre bakılırsa düşük çıkmaktadır ancak hesapladığımız puana göre takımları sıralayıp ligi bitirme pozisyonlarına göre değerlendirdiğimiz zaman daha sağlıklı sonuçlar elde etmekteyiz.

Örneğin gerçeğine çok yakın değerde puan tahmini yapmasına rağmen tam olarak puanı tahmin edemediği zaman doğruluk değerimiz olumsuz etkilenmektedir ancak aslında projemizin amacı gereği puana göre sıralama yaptığımızda belirli takımların puan tablosundaki yerlerinin aslında çok yakın olduğunu yani gerçeğe yakın bir tahminleme yapıldığını görmekteyiz.

Modelimizin başarısına arttırmaya yönelik normalizasyon yöntemlerinden “Min Max Scaler” yöntemini de uygulamaktayız.

# Deneysel Çalışmalar:

Veri setimizi elde ederken Kaggle sitesi üzerinden “Football Data: Expected Goals and Other Metrics“[[1]](https://www.kaggle.com/datasets/slehkyi/extended-football-stats-for-european-leagues-xg?select=understat.com.csv) veri setini temel aldık. Bu veri setinden çözümümüz için gerekli olan ve olmayan verileri belirledik. Daha sonra gereksiz sütunları kaldırarak “Big 5 European football leagues teams stats”[[2]](https://www.kaggle.com/datasets/jehanbhathena/big-5-european-football-leagues-stats) veri setinden bulduğumuz ve gerekli gördüğümüz başka verileri ekliyoruz (Görsel-1).



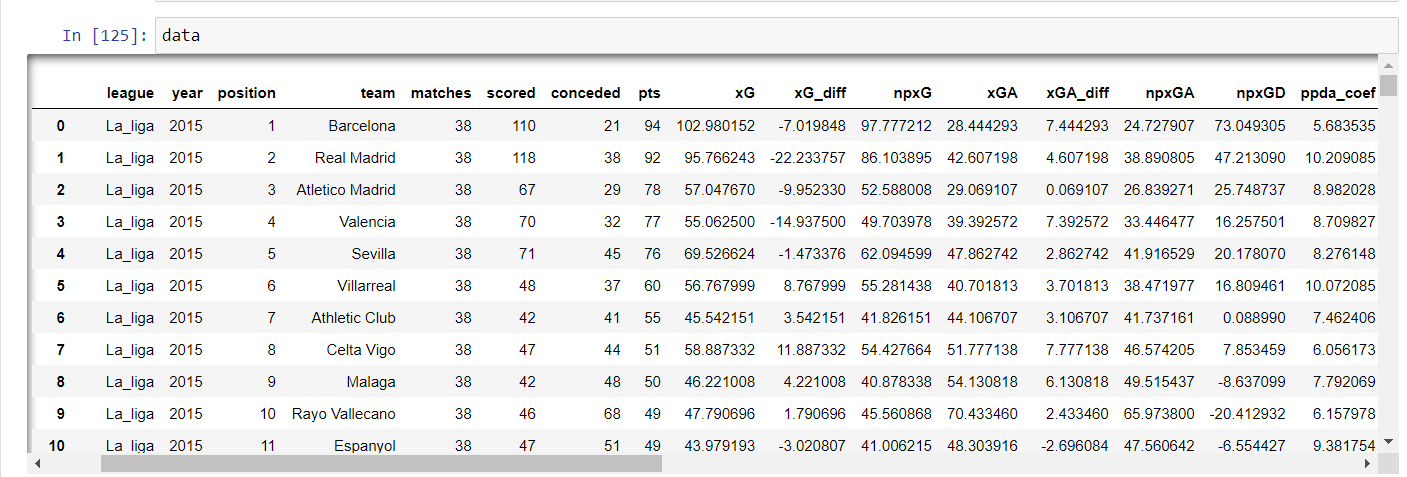
Görsel-1

metin içeren bir resim

Açıklama otomatik olarak oluşturuldu

Görsel-2

Veri setimiz çalışmamızın konusuna uygun olarak belli futbol kulüplerinin belirli yılardaki bazı istatistiksel verilerini (attığı gol, yediği gol, gol beklentisi, asist beklentisi vs.) içermektedir. 33 adet öznitelik verimiz, 1 adet hedef çıktımız, toplamda ise 582 satır örnek verimiz bulunmaktadır (Görsel-3).



Görsel-3

Öncelikle Yapay Sinir Ağı yöntemini (MLP Classifier) kullanarak algoritmamızı işletiyoruz. Bunun için gerekli olan tanımlamaları yapıyoruz. Modelimizin tahminlediği tüm liglere ait olan puanları gözlemliyoruz (Görsel-4).

metin içeren bir resim

Açıklama otomatik olarak oluşturuldu

Görsel-4

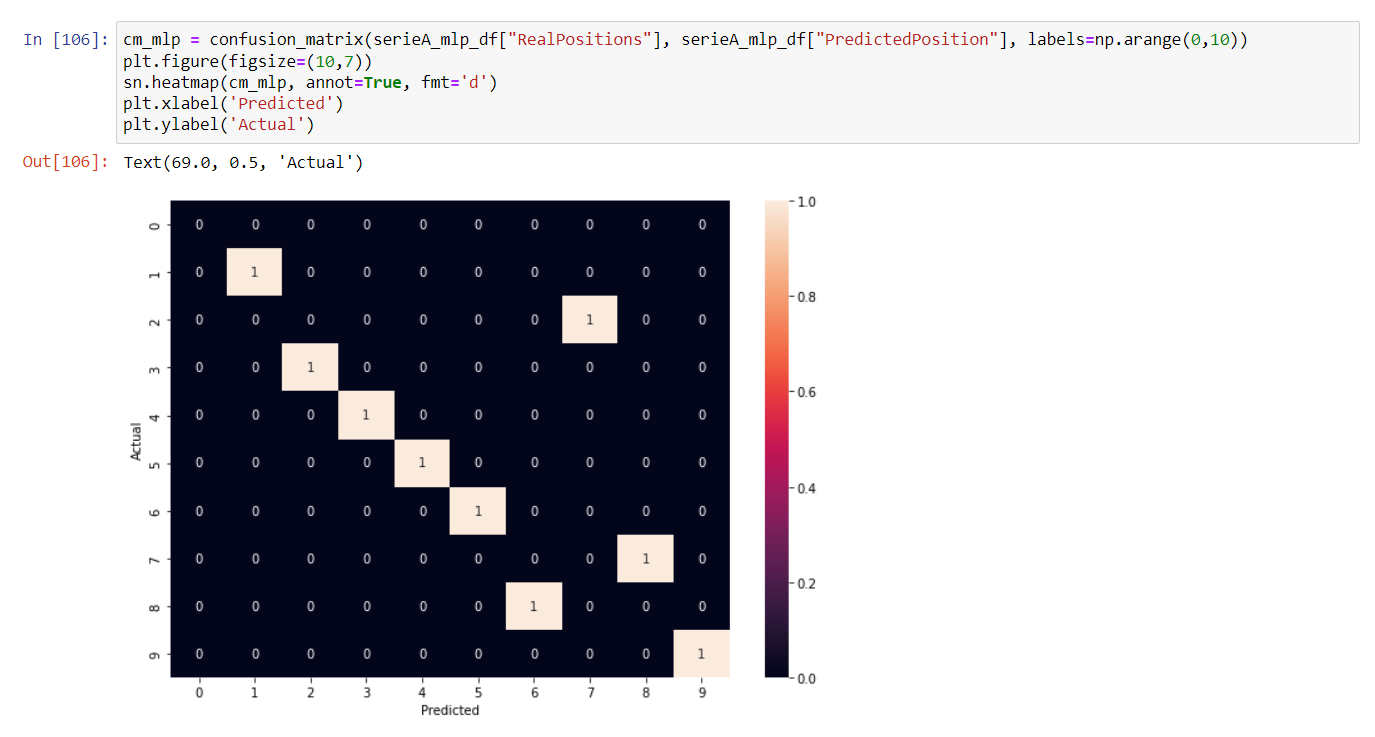
İlk denediğimiz olan MLP Classifier yönteminde Serie-A ligini ele alıyoruz. Buna göre gerekli tanımlamalar ve bölümlemeler yapıldıktan sonra sonucumuzun doğruluk değerlerine bakıyoruz (Görsel-5). Doğruluk değerimizin pozisyona göre incelendiği zaman daha yüksek olduğunu gözlemleyebilmekteyiz.

metin içeren bir resim

Açıklama otomatik olarak oluşturuldu

Görsel-5

Bu denememize ait hata matrisi oluşturmamızı sağlayan kod ve ekran görüntüsü şu şekildedir (Görsel-6).



Görsel-6

Hata matrisimize baktığımız zaman asıl amacımız olan pozisyon bilgisine göre çok da sağlıklı olmayan bir sonuç elde ediyoruz. Bunun için sonuçlarımızı daha doğru şekilde yorumlayabilmek adına yeni bir dataframe şeklinde sonuçlarımızı görselleştiriyoruz. Bu tablo sayesinde hem gerçekçi bir görünüm hem de daha sağlıklı bir değerlendirme elde edebiliyoruz (Görsel-7). (Bu sebepten dolayı daha sonraki denemelerde yalnızca tablo görünümüne yer verilmiştir. Hata matrisine istenildiği takdirde kod işletilerek ulaşılabilir)

tablo içeren bir resim

Açıklama otomatik olarak oluşturuldu

Görsel-7

Sonuçlarımızdaki doğruluk değerlerini geliştirebilmek adına veri ölçeklendirme yöntemini kullanarak MLP Classifier yöntemini bu şekilde tekrar deniyoruz.

Scaled MLP Classifier yönteminde farklı bir ligi gözlemlemek adına Bundesliga ligini ele alıyoruz. Buna göre gerekli tanımlamalar ve bölümlemeler yapıldıktan sonra sonucumuzun doğruluk değerlerine ve puan tablomuza bakıyoruz (Görsel-8,9).

tablo içeren bir resim

Açıklama otomatik olarak oluşturuldu

Görsel-8

metin içeren bir resim

Açıklama otomatik olarak oluşturuldu

Görsel-9

Doğruluk değerlerinde istediğimiz gelişmeyi göremeyince Random Forest yöntemini kullanarak geliştirmeyi deniyoruz. Random Forest yöntemi için gerekli tanımlamalar ve işlemler yapıldıktan sonra doğruluk değerlerini gözlemliyoruz (Görsel-10,11).

tablo içeren bir resim

Açıklama otomatik olarak oluşturuldu

Görsel-10

metin içeren bir resim

Açıklama otomatik olarak oluşturuldu

Görsel-11

Doğruluk değerlerinde puana göre yüzde 40, pozisyona göre bakıldığında ise yüzde 65’lik bir oranı elde ediyoruz. Bu yönteme ait puan tablosunu da gözlemleyebilmekteyiz (Görsel-12).

tablo içeren bir resim

Açıklama otomatik olarak oluşturuldu

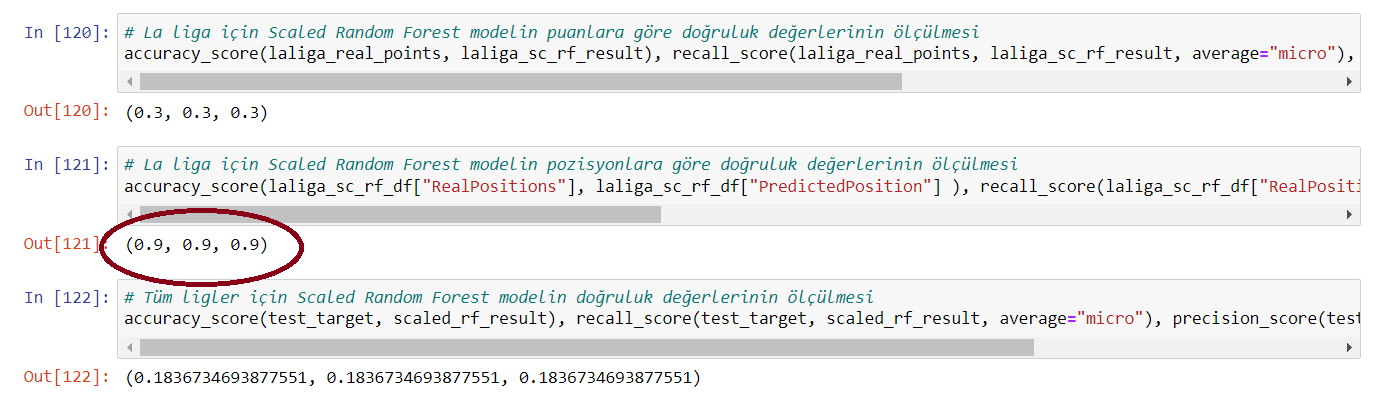
Görsel-12

Son olarak yöntemimizi biraz daha geliştirebilmek için Random Forest yöntemini veri ölçeklendirme yöntemiyle uyguluyoruz ve puan tablomuz ile doğruluk oranlarımızı gözlemliyoruz (Görsel-13,14).

tablo içeren bir resim

Açıklama otomatik olarak oluşturuldu

Görsel-13



Görsel-14

La liga ligini ele aldığımız bu son denememizde yüzde 90’a ulaşan bir doğruluk değeri yakalıyoruz ve yapay zekâ programımızın gerçeğe çok yakın değerler tahminleyebildiğini gözlemliyoruz.

# Kaynak Kodlar:

Kaynak kodumuz çok uzun olup raporda çok fazla sayfa kapladığı için rapor formatını bozmaması adına proje klasöründe “Kaynak Kod” isimli dosyaya ayrıca eklenmiştir.

# Sonuç:

Yapmış olduğumuz proje sonucunda geliştirilmeye açık ancak kullanılabilirliği şimdiden yüksek olan bir program elde ettik. Dünyadaki spor dallarından en geniş kitleye sahip olan futbol üzerine yaptığımız bu çalışma gerekli ve yeterli veri bulunması halinde her futbol ligi için uygulanabilir durumdadır. Futbol sektöründe verimi ve başarıyı arttırmaya yönelik çalışmalarda kullanıma uygundur.

Bizler bu projeyi geliştirirken hem kullandığımız ortam, yöntem ve kütüphane gibi teknik araçları öğrenirken hem de bir projenin başarısının hangi parametrelere göre artıp hangi durumlarda azaldığını gözlemleme şansı yakaladık. Kaggle sitesi ve Jupyter ortamı gibi veri alanında önemli ve genel geçer araçların detaylarını öğrendik.

# Ek1:

Başarımı artırmak için verisetindeki özniteliklerimizi scale ettik, %5-10 arası bir başarı artırımı elde ettik.

2. olarak puan tahmini etme değil, tahmin ettiğimiz puanlamaya göre oluşturduğumuz sıralamadan lig sıralaması tahmin etme ve gerçeği ile karşılaştırma yoluna gittik. Böylece başarı oranımızı 2'ye kadar katlamayı başardık.

3. olarak farklı makine öğrenmesi algoritmaları denedik ve Random Forest'ta daha yüksek başarılar elde ettik.

# Ek2:

Literatüre katkımız (çalışmamızın farklılığı): Konumuzla alakalı internette 2 veri seti vardı. Biri daha geleneksel istatistikler içeriyorken, biri gelişmiş istatistikler kullanıyordu. Biz ikisinden de faydalanabileceğimizi düşündük. Bu ikisinin kombinasyonunu elde etmek için gerekli düzenlemeleri yapıp veri setlerimizi birleştirdik.

# Ek3:

İnternetten bu konuda yapılmış bazı çalışmalara dair yazılar okuduk. Kaggledan bulup faydalandığımız verisetleri hakkında yapılmış bazı jupyter notebook çalışmalarını inceledik. Alta bunların linkini bırakmakla beraber, faydalandığımız tüm kaynakları kaynakça kısmında bulabilirsiniz. Önceki yapay zekâ ve yöntemleri projemiz ve makine öğrenmesi konularımızdan yola çıkarak bu projeyi yaparken gideceğimiz yolu belirledik.

<https://www.kaggle.com/code/yassershrief/how-to-be-a-champion-in-european-leagues-football>

<https://www.kaggle.com/code/aknkaradeniz/football-ab-test>

# Ek4:

Kümeleme algoritmaları için kullanılan performans ölçütleri:

1)Silüet Katsaysı s=(b-a) /max(a,b)

Her veri için hesaplanır. a: bir veri ile aynı kümedeki diğer tüm noktalar arasındaki ortalama mesafe.

b: Bir örnek ile bir sonraki en yakın kümedeki diğer tüm noktalar arasındaki ortalama mesafe.

Bir veri setinin silüet katsayısı, her verinin silüet katsayısının ortalamasıdır. Puan yanlış kümeleme için -1, çok yoğun kümeleme için +1’le sınırlandırılmıştır.

Kümeler yoğun ve iyi ayrılmış olduğunda (better separation) puan daha yüksektir.

2) Dunn’s Index (DI): DI = minimum kümeler arası mesafe/maksimum küme boyutu

Kümeler arası büyük mesafeler (better separation) ve daha küçük küme boyutları (more compact clusters) ile büyük DI değerleri elde edilir. Daha yüksek bir DI değeri daha iyi bir kümeleme anlamına gelir.

Transformer modeller: giriş verilerinin her bir bölümünün önemini farklı şekilde ağırlıklandıran, kendi kendine dikkat mekanizmasını (self attention mechanism) benimseyen bir derin öğrenme modelidir. Öncelikle doğal dil işleme (NLP) ve bilgisayarlı görme (CV) alanlarında kullanılır.

# Kaynakça:

<https://www.kaggle.com/datasets>

<https://datasetsearch.research.google.com/>

<https://archive.ics.uci.edu/ml/datasets.php>

<https://www.kaggle.com/code/aknkaradeniz/football-ab-test>

<https://www.kaggle.com/code/yassershrief/how-to-be-a-champion-in-european-leagues-football>

<https://ayyucekizrak.medium.com/yapay-zekaya-ba%C5%9Flama-rehberi-91e79d3de8e1>

<https://github.com/ayyucekizrak/Kapsamli_Derin_Ogrenme_Rehberi>

<https://www.kaggle.com/datasets/slehkyi/extended-football-stats-for-european-leagues-xg?select=understat.com.csv>

<https://www.kaggle.com/datasets/jehanbhathena/big-5-european-football-leagues-stats>

<https://www.kaggle.com/datasets/varpit94/football-teams-rankings-stats?select=Football+teams.csv>

<https://understat.com/>

<https://www.shanelynn.ie/pandas-drop-delete-dataframe-rows-columns/>

<https://stackoverflow.com/>

<https://www.geeksforgeeks.org/>

<https://www.kaggle.com/code/yassershrief/how-to-be-a-champion-in-european-leagues-football>

<https://www.kaggle.com/code/aknkaradeniz/football-ab-test>

<https://www.geeksforgeeks.org/different-ways-to-create-pandas-dataframe/>

<https://www.analyticsvidhya.com/blog/2020/10/quick-guide-to-evaluation-metrics-for-supervised-and-unsupervised-machine-learning/#:~:text=Clustering%20Performance%20Evaluation%20Metrics&text=Here%20clusters%20are%20evaluated%20based,then%20it%20has%20performed%20well>.

<https://www.programiz.com/python-programming/list#:~:text=In%20Python%2C%20a%20list%20is,brackets%20%5B%5D%20%2C%20separated%20by%20commas.&text=A%20list%20can%20have%20any,%2C%20string%2C%20etc.).&text=A%20list%20can%20also%20have,is%20called%20a%20nested%20list>.

# Özdeğerlendirme Tablosu

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | **İstenen Madde** | **Var** | **Açıklama** | **Tahmini Not** |
| **1** | **Kapak Sayfası, Problemin Tanımı, Kullanılan Ortam, Yöntem ve Kütüphaneler, Araştırma (10)** | x | Kapak sayfası yapıldı. Problem tanımlandı. Kullanılan ortam, yöntem ve kütüphaneler belirtildi. Yapılan araştırmalar yazıldı. | 10 |
| **2** | **Önerilen Yöntem (10)** | x | Çözüm için önerilen yöntem açıklandı. | 10 |
| **3** | **Deneysel Çalışmalar (10)** | x | Şu ana kadar yapılabilen deneysel çalışmalar belirtildi. | 10 |
| **4** | **Proje Rapor Biçimi, Organizasyonu, Boyutu, Kalitesi, Kaynakça ve atıflar (10)** |  | Projemizin ara raporu hazırlandı. Şu ana kadar yararlandığımız kaynaklar ve atıflar belirtildi. | 10 |
| **5** | **Sonuç (10)** |  | Sonuç ve kazanımlarımız istenilen şekilde belirtildi | 10 |
| **6** | **Ek 1: Başarım İyileştirme (10)** |  | Başarımı iyileştirmek adına hangi yolları izlediğimiz açıklandı | 10 |
| **7** | **Ek 2 (10)** |  | Çalışmamızın farklılığı belirtildi. | 10 |
| **8** | **Ek 3 (10)** |  | Faydalandığımız bağlantılar ve farklılıklarımız açıklandı. | 10 |
| **9** | **Ek 4 (10): Her madde 5’er puan.** |  | A ve B maddeleri öğrenilip kendi cümlelerimizle açıklandı. | 10 |
| **10** | **Özdeğerlendirme Tablosu (10)** |  | Yapıldı | 10 |
| **100 üzerinden Toplam Not:** | | | | 100 |

# İş Bölümü ve Çalışma Süreleri Tablosu

X değeri saat cinsinden:

|  |  |  |
| --- | --- | --- |
| İSTENEN MADDE | ÇAĞATAY | ANIL |
| 1. | X10 | X10 |
| 2. | X5 | X5 |
| 3. | X5 | X5 |
| 4. | X1 | X2.5 |
| 5. | X0.5 | X1. |
| 6. (Ek 1) | X0.5 | X0.25 |
| 7. (Ek 2) | X0.5 | X0.25 |
| 8. (Ek 3) | X0.5 | X0.25 |
| 9. (Ek 4) | X1 | X0.25 |