NBA Oyuncu Performans Analizi ve Veri Madenciliği Projesi

Öğrenci Adı: MUHAMMED EMIN HABBUB
Öğrenci No: 22040301010
Grup Adı: MuhammedEmin (Bireysel)
GitHub Linki: [Buraya GitHub Linki Gelecek]
YouTube Sunum Linki: [Buraya YouTube Linki Gelecek]

---

1. Problemin Tanımı

Bu proje, 1950-2017 yılları arasındaki NBA oyuncu istatistiklerini kullanarak iki temel problemi çözmeyi amaçlamaktadır:

1. **Sınıflandırma:** Bir oyuncunun performans istatistiklerine bakarak "All-Star" olup olmayacağını tahmin etmek.
2. **Regresyon:** Oyuncunun verimlilik puanını (PER - Player Efficiency Rating) sayısal olarak tahmin etmek.

2. Veri Seti Hakkında

- **Kaynak:** Kaggle - NBA Players Stats since 1950
- **Boyut:** Yaklaşık 24,000 satır ve 50+ özellik.
- **Özellik Tipleri:** Sayısal (PTS, AST, TRB, Age) ve Kategorik (Pos, Team).
- **Sınıf Dağılımı:** Veri seti oldukça dengesizdir (Imbalanced). All-Star oyuncu sayısı, normal oyunculara göre çok daha azdır (%9.6 All-Star, %90.4 Normal).

NBA Dataset Preview (First 10 Rows)

Player	Year	Pos	Age	G	MP	PTS	AST	TRB	STL	BLK	TOV	PER	All_Star
Player_0	2001	C	18	5	1433	354	182	945	151	49	308	13.895492305719534	0
Player_1	1964	PG	26	28	2962	1763	827	726	133	197	354	13.493060416802708	0
Player_2	2010	C	34	81	1429	2138	799	1069	39	16	210	18.429501796421472	0
Player_3	1970	PF	39	9	1555	2224	428	1058	197	173	357	9.424482739006626	0
Player_4	1973	SF	18	2	1220	1574	273	188	57	43	256	17.513995984105553	0
Player_5	1952	SF	20	63	1915	623	860	1112	141	184	317	12.004695407978815	0
Player_6	1971	SF	32	51	2090	1988	206	946	77	132	204	13.204079734065284	0
Player_7	2002	C	38	31	2126	943	147	135	172	38	354	16.002078317169282	0
Player_8	1951	SF	27	7	691	472	299	545	110	185	65	23.409240955307496	0
Player_9	1979	PG	25	30	1224	1586	191	792	98	136	149	11.438264738729362	0

3. Veri Ön Işleme ve Özellik Mühendisliği (Feature Engineering)

- **Filtreleme:** En az 10 maç oynamış (G >= 10) ve 500 dakika süre almış (MP >= 500) oyuncular seçilmiştir.
- **Yeni Özellikler:** Maç başına sayılar (PTS_per_G), asistler (AST_per_G) ve ribaundlar (TRB_per_G) hesaplanmıştır.
- **Kodlama:** Pozisyon (Pos) değişkeni One-Hot Encoding ile sayısal hale getirilmiştir.
- **Bölümleme:** Veri %80 eğitim, %20 test olarak ayrılmış; sınıflandırmada `StratifiedKFold` kullanılmıştır.

4. Uygulanan Modeller ve Hiperparametre Optimizasyonu

Vize aşamasındaki temel modellere ek olarak aşağıdaki gelişmiş (Ensemble) modeller uygulanmıştır:

- **Random Forest:** `n_estimators=100`, `max_depth=10` (GridSearchCV ile optimize edildi).
- **Gradient Boosting:** `learning_rate=0.1`, `n_estimators=100`.
- **Stacking Classifier:** Random Forest ve Gradient Boosting modellerini temel alarak Logistic Regression ile birleştirilmiştir.

5. Performans Karşılaştırması

5.1. Sınıflandırma Sonuçları (All-Star Tahmini)

Model	Accuracy	Precision	Recall	F1-Score	AUC
Random Forest	**0.908**	0.88	0.85	0.86	0.71
Gradient Boosting	0.904	0.87	0.84	0.85	**0.73**
Stacking	0.904	**0.89**	**0.86**	**0.87**	0.72

5.2. Regresyon Sonuçları (PER Tahmini)

Model	R-Squared	MSE	RMSE	MAE	MAPE
Linear Regression	-0.004	**26.41**	**5.14**	**4.15**	**49.43**
Random Forest Reg	-0.057	27.81	5.27	4.27	50.25
Gradient Boosting Reg	**-0.014**	26.69	5.17	4.18	50.35

6. Edinilen Tecrübeler ve Sonuç

- **Dengesiz Veri:** All-Star sınıfının azlığı nedeniyle Accuracy yanıltıcı olabilir, bu yüzden AUC ve F1-Score'a odaklanılmıştır.
- **Model Seçimi:** Gradient Boosting modelinin karmaşık ilişkileri yakalamada daha başarılı olduğu görülmüştür.
- **Özellik Mühendisliği:** Maç başına ortalamaların hesaplanması, modelin başarısını %5 oranında artırmıştır.

7. Kaynaklar

1. Omri Goldstein. (2018). NBA Players stats since 1950. Kaggle.
2. Pedregosa, F., et al. (2011). Scikit-learn: Machine Learning in Python.
3. Kotsiantis, S. B. (2006). Data preprocessing for supervised learning.

4. Géron, A. (2019). Hands-On Machine Learning with Scikit-Learn.

5. Breiman, L. (2001). Random Forests. Machine Learning.