FREELANCER BAŞARILI - BAŞARISIZ TAHMİNİ

Şevval CANDAN

Bilgisayar Mühendisliği
Ankara Üniversitesi
Ankara, Türkiye
1sevval.candan@gmail.com

***Özet* — Freelancer çalışan kişilere ait toplam kazandıkları tutar, yaşadıkları ülke, saat başı çalışma ücreti, işlerini zamanında yapıp yapmadıkları gibi çeşitli verilerin bir işe girerken kabul alma olasılığı üzerindeki etkisi incelenmiştir. Bu veriler freelancer.com web sitesinin API tokenı ve web scraping yöntemi kullanılarak elde edilmiştir. Elde edilen veriler üzerinde birçok sınıflandırma yöntemi uygulanmıştır. Daha sonrasında ise sonuçlar görselleştirilmiştir.**

Anahtar Kelimeler — freelancer, freelancer çalışma, kabul oranı, makine öğrenmesi, modelleme, analiz

I. BAŞLANGIÇ

Son yıllarda birçok insan evden çalışabilecekleri işlerde çalışmayı tercih etmektedir. Teknolojinin yaygınlaşması, internete her yerden ulaşılabilmesi gibi sebeplerle artık insanlar herhangi bir şirkete bağlı kalmadan online işlere başvuru yapıp istedikleri yerden çalışabilmektedirler. Modern dünyada freelancer işlerde çalışıp para kazanan insan sayısı gün geçtikçe artmaktadır. Internette freelancer işlerin çalışanlarla buluştuğu birçok web sitesi bulunmaktadır. Örneğin; https://www.upwork.com, https://www.freelancer.com, https://www.fiverr.com ... Freelancer işlere artan talep bu alandaki rekabeti artırmaktadır. Bu projede freelancer.com web sitesinden alınan kullanıcı verilerinden faydalanılarak hangi özelliklere sahip kullanıcıların müşteriler tarafından daha çok tercih edileceği incelenmiştir. Birçok sınıflandırma modeli ile tahminler yapılmıştır. Daha doğru tahmin üreten model olan Random Forest modeli seçilmiştir. Elde edilen sonuçlar freelancer olarak çalışmak isteyen insanlara yardımcı olacak detaylar sunmaktadır.

II. YÖNTEM

A. Veri Seti

Bu projede veri seti freelancer.comdaki kullanıcıların verileri oluşturulmuştur. Veri setindeki değişkenler : Username, Profile URL, Amount Earned, Completed Jobs, USD/Hour, On Time, On Budget, Repeat Hire Rate, Accept Rate, Country, Overall Rating, Reviews Count, Education, Work Experience,Number of Certificates. Modelin daha iyi öğrenebilmesi için oluşturulmuş yeni değişkenler: Worked Hours(Amount Earned / (USD/Hour)), Rating Strength, Work_Experience_Weighted, Education_x_Certificates, Education_x_WorkExp, Continent. Rating Strength yorum sayısı fazla olanların ratingine daha fazla ağırlık verilmesini sağlamıştır. Work_Experience_Weighted ile kullanıcının iş deneyimi arttıkça ağırlık arttırılmaktadır. Country değerleri continent ile sınıflandırılmıştır. Veri setindeki kategorik değişkenler sayısal değişkenlere dönüştürülmüştür. Veri seti toplam 3021 adet değerden oluşmaktadır. Bu çalışmada kullanılan veri setinin büyüklüğü ve değişken çeşitliliği freelancer kullanıcılarının kabul ya da red alma sonucunu değerlendirebilmek için yeterli kapsama sahiptir.

B. Problem Tanımı

Bu çalışmanın amacı, modelin kullanıcıların verilerini analiz ederek accept oranını tahmin edilmesidir. Model doğrudan accept rate sayısal ifadesini tahmin etmek yerine bu oranı **ikili sınıflandırma problemi** olarak ele almaktadır. Eşik değer üzerindeki değerler başarılı (1) altındaki değerler başarısız (0) olarak sınıflanırıldı. Bu eşik değer veri setinde yer alan accept rate değerlerin ortalaması alınarak oluşturuldu. Ortalama değerin üstünde accept rate değerine sahip olanları modele başarılı, altında olanları ise başarısız olarak öğretilmiştir.

III. MODELLEME SÜRECI KARŞILAŞILAN ZORLUKLAR

Bu çalışmada accept rate oranını tahmin edilebilmesi için birden fazla model ile çalışılmıştır. En iyi sonucu veren Random Forest modeli hiperparametre optimizasyonu yapılmıştır. Çok sayıda farklı parametre değerleri Randomized Search CV ile rastgele denenmiştir. En iyi performansı gösteren model seçilmiştir.

A. Veri Setinin Training ve Test Olarak Ayrılması

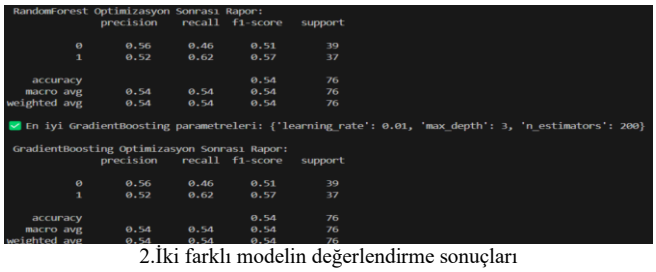
Feature ve target değişkenleri belirlendikten sonra. Veri seti %80 eğitim %20 test verisi olmak üzere rastgele ikiye ayrılmıştır.

```
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)
```

1. Eğitim verisi ve test verisinin oluşturulması kodu

B. Yaşanan Başarısızlıklar

Oluşturulan veri seti ile birden fazla sınıflandırma modeli kullanılmıştır. İlk çalışılmaya başlandığında 200 gibi kısıtlı bir veri seti kullanılmıştı. Feature engineering gibi yöntemler denenilmemişti. Ayrıca başarılı - başarısız sınıfları arasında dengesizlik tespit edilmişti. Bu dengesizliğin sebebi freelancer sitesinden çekilen profillerin çoğunluğunun yüksek başarı oranına sahip olmasıdır. Bu durum verilerin rastgele çekilmesi sonucunda oluşmuştur. Bunlardan kaynaklı olarak logistic regression, random forest classifier, gradient boosting classifier gibi sınıflandırma modelleri doğru sonucu tahmin etme konusunda başarız sonuç ürettiği gözlemlenmiştir. Aşağıdaki sonuçlardan anlaşılacağı üzere model yaklaşık %50 oranında doğru sonuç üretirken geri kalan için yanlış sonuçlar üretmektedir. Model tesadüfi tahminler yaptığı sonuçlardan anlaşılmıştır.



2.İki farklı modelin değerlendirme sonuçları

XXX-X-XXXX-XXXX-X/XX/$XX.00 ©20XX IEEE

C. Model İyileştirme Süreci

Bu problem için veri setinin genişletilmesi, değişkenlerin feature engineering ile arttırılması ya da ağırlıklandırılması ve SMOTE gibi yöntemlerle azınlık sınıfı için mevcut azınlık sınıfı örneklerine benzer ama yapay veriler üretilmesi aşamaları uygulanmıştır. Bu aşamalardan sonra modellerin başarı oranı yükselmiştir.

```
continent_map = {
    "Europe": [
        "Serbia", "Turkey", "United Kingdom", "Germany", "Sweden", "Romania", "Greece", "Ukraine",
        "Croatia", "Ireland", "Georgia", "Bosnia And Herzegovina", "Andorra", "Netherlands", "Bulgaria",
        "Moldova, Republic Of", "Finland", "Belarus", "Poland", "Czech Republic", "Norway", "Italy",
        "Latvia", "Armenia", "Portugal", "France", "Austria", "Estonia", "Macedonia", "Kosovo",
        "Kazakhstan", "Montenegro", "Albania", "Slovenia", "Cyprus"
    ],
    "Asia": [
        "India", "Pakistan", "Bangladesh", "Malaysia", "Sri Lanka", "Georgia", "Iraq", "Palestinian Territory",
        "Indonesia", "United Arab Emirates", "Jordan", "Lebanon", "Vietnam", "Armenia", "Qatar", "Uzbekistan",
        "Saudi Arabia", "Kazakhstan", "Japan", "Nepal", "Tajikistan", "Taiwan", "Singapore", "Cambodia",
        "Thailand", "Kyrgyzstan"
    ],
    "Africa": [
        "Nigeria", "Egypt", "Tunisia", "Morocco", "Malawi", "Cameroon", "Ethiopia", "Cote D'Ivoire",
        "Madagascar", "Ghana", "Libya", "Zimbabwe", "Algeria", "Mauritius", "Liberia",
        "Tanzania, United Republic Of", "South Africa"
    ],
    "North America": [
        "United States", "Canada", "Mexico", "Jamaica", "Panama", "El Salvador", "Costa Rica", "Dominican Republic",
        "Trinidad And Tobago", "Nicaragua"
    ],
    "South America": [
        "Venezuela", "Colombia", "Argentina", "Peru", "Uruguay", "Brazil", "Chile", "Paraguay", "Ecuador"
    ],
    "Oceania": [
        "Australia", "New Zealand"
    ],
    "Other": []
}
```

3. Country değerlerinin sınıflandırılmasında kullanılan continent_map

```
df['Worked Hours'] = (df['Amount Earned'] / df['USD/Hour']).replace([np.inf, -np.inf], np.nan).fillna(0)
df['Rating Strength'] = (df['Overall Rating'] * np.log10(1 + df['Reviews Count'])).fillna(0)
weight_map = {0: 0.5, 1: 1, 2: 2, 3: 4}  # 3'e en yüksek ağırlık
df['Work_Experience_Weighted'] = df['Work Experience'].map(weight_map)

df['Education'] = df['Education'].astype(int)  # 0/1 olduğundan emin ol

# Eğitim varsa deneyimle çarp
df['Education_x_WorkExp'] = df['Education'] * df['Work Experience']

# Eğitim varsa sertifika sayısıyla çarp
df['Education_x_Certificates'] = df['Education'] * df['Number of Certificates']

bins = [-1, 0, 3, 7, 15]
labels = ['No Certificates', 'Few', 'Medium', 'Many']
df['Certificates_Binned'] = pd.cut(df['Number of Certificates'], bins=bins, labels=labels)
df = pd.get_dummies(df, columns=['Certificates_Binned'], drop_first=True)
cert_bins_cols = [col for col in df.columns if col.startswith('Certificates_Binned_')]
```

4. Yapılan feature engineering işlem kodları

D. Modelleme Süreci

Logistic Regression ve decision tree sınıflandırma modelleri bu problem üzerinde çalıştırıldı. Fakat değerlendirme metrik sonuçlarına göre random forest modelinden daha az başarılı oludukları için bu modeller kullanılmayacağına karar verildi.

Model Adı	Accuracy (%)	F1 Score (%)	(%) ROC AUC
Logistic Regression	65,00%	73%	0,68
Decision Tree	67,20%	75%	0,70
Random Forest	68,00%	75%	0,75
Random Forest(Randomized SearchCV)	69,80%	77%	0,74

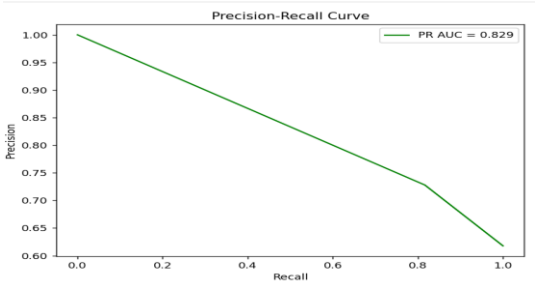
5. Modellere göre evaluation metrics sonuçları

E. Model İyileştirme Süreci

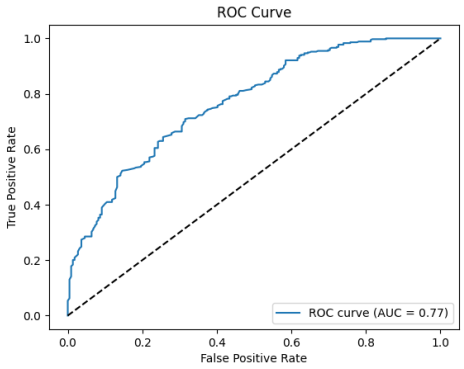
Random forest modeli çeşitli hiperparametre değerleri ayarlanılarak çalışan bir modeldir. Bu çalışmada random forest modelin başarı oranını arttırmak için hiperparametre ayarlanılmıştır. Bu süreçte modelin en iyi parametrelerini rastgele deneyip bulan RandomizedSearchCV hiperparametre optimizasyon yöntemi kullanılmıştır. Bu yöntemde ağaç derinlikleri None, 10, 20, 30; ağaç sayısı 100 ve 400 arasında; dallanma için gereken minimum örnek sayısı 0, 5, 10; yaprak düğümde minimum örnek sayısı 1, 2, 4 parametre optimizasyonunda kullanılan değerler olarak belirlenmiştir. Optimizasyon yapıldıktan sonra eğitilen modelde kullanılan en iyi parametrelerin max_depth için 30, max_features için log2, min_samples_leaf için 1, min_samples_split için 2, n_estimators için 413 olduğu gözlemlenmiştir. Bu da en başarılı olan modelin bu parametre ayarlarına sahipken gerçekleştiğini göstermektedir.

F. Model Değerlendirilmesi

- **Accuracy (Doğruluk): 0.698:** Modelin test verilerinin yaklaşık %70 oranında doğru tahmin ettiğini göstermektedir.
- **Precision (Kesinlik): 0.728:** Modelin başarılı sınıfında tahmin ettiği verilerin yaklaşık %73 oranında doğru tespit ettiğini göstermektedir.
- **Recall (Duyarlılık): 0.816 :** Model gerçekte başarılı olan değerlerin %81'ini yakaladığını göstermektedir.
- **F1 Score: 0.770 :** Precision ve recall değerlerinin harmonik ortalaması F1 skor değerini vermektedir.



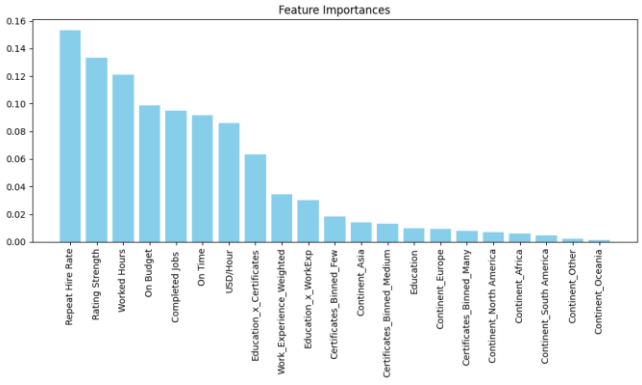
6. Modelin farklı eşik değerlerindeki precision ve recall dengesi



7.ROC -AUC grafiği

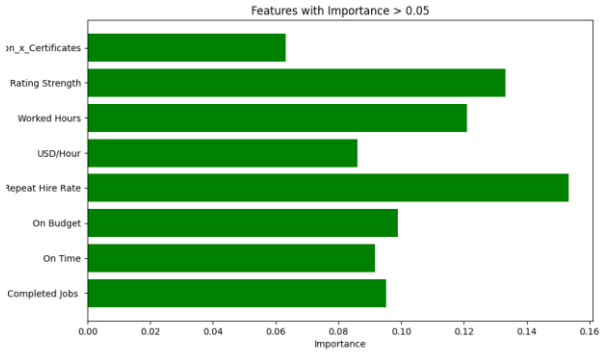
- **AUC : 0.77 :** Bire yakın bir değer olduğu gözlemlenmiştir. Bu da modelin pozitif sınıfı negatif sınıftan ayırma yeteneğinin yüksek olduğunu göstermektedir.

IV. Sonuçlar ve Grafik Yorumu

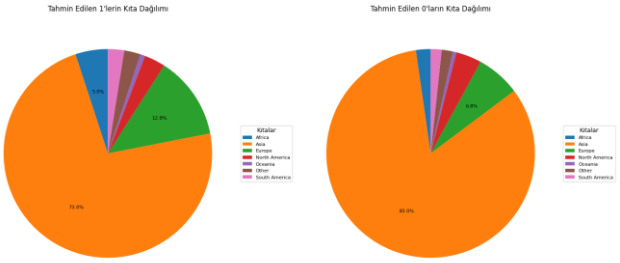


8. Feature Importance Grafiği

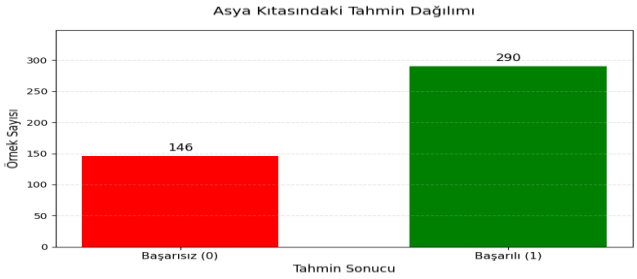
Random Forest algoritmasını öğrenirken hangi değerlerin daha çok modelin öğrenmesinde işe yaradığını gösteren grafik yukarıda verilmiştir. Model için en önemli özellik %16 değeri ile tekrar işe alım oranı olmuştur. Aşağıdaki grafikte ise en önemli etkisi olan 8 özellik gösterilmiştir.



9. Etkisi 0.05'ten fazla olan özellikler grafiği

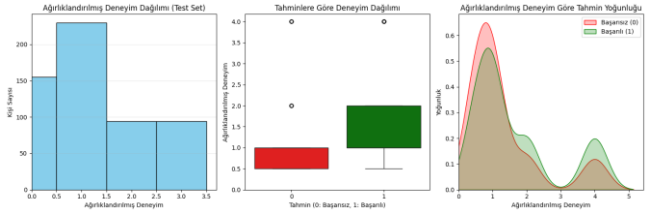


10. Başarılı ve başarısız seçilenlerin kıta dağılım daire grafiği



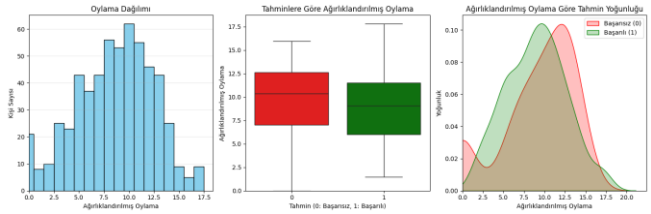
11. Asya kıtasındaki başarılı - başarısız profil sayıları grafiği

10 numaralı resme bakarak Asya kıtasındaki kullanıcıların diğer kıtalara göre daha fazla kabul aynı zamanda red oranı olduğu tespit edilmiştir. Bunun sebebi bu kıtada yaşayan daha çok kullanıcı olmasından kaynaklıdır. 11 numaralı görseldeki değerlere bakarak Asya kıtasındaki kullanıcıların başarılı olanlarının sayısının başarısız olanların sayısından çok daha fazla olduğu görülmüştür.



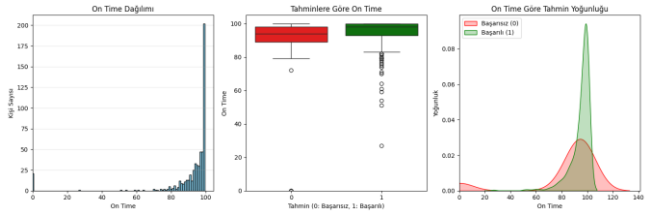
12. Deneyim Dağılımı, Tahminlere Göre Deneyim Dağılımı, Deneyim Ağırlığına Göre Tahmin Yoğunluğu Grafiği

12 numaralı görseldeki grafiklere bakarak başarısız olarak seçilen kişilerin ağırlıklandırılmış deneyiminin 1 ve daha düşük, 1'den fazla olan ağırlıklandırılmış deneyim değerinin başarılı olan kullanıcılarla eşleştirilmiştir.



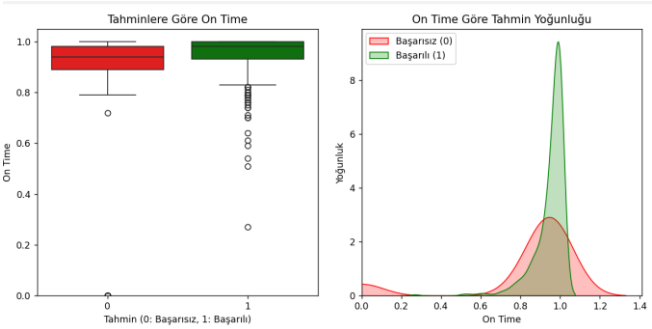
13. Ağırlıklandırılmış Oylama Dağılımı, Tahminlere Göre Ağırlıklandırılmış Oylama Dağılımı, Ağırlıklandırılmış Oylama Göre Tahmin Yoğunluğu Grafiği

13 numaralı görselde yer alan grafiklerden anlaşılacağı üzere modelin başarılı - başarısız tahminlerini bu değerlere bağlı kalmadan yaptığı gözlemlenmiştir.



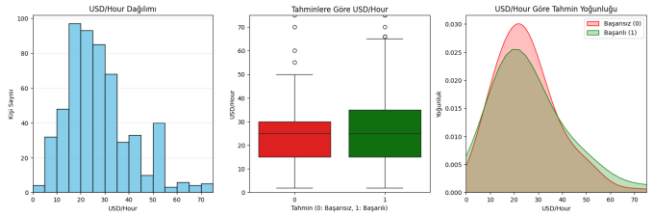
14. On Time Dağılımı, Tahminlere Göre On Time ve On Time Göre Tahmin Yoğunluğu

14 numaralı görseldeki tahmin yoğunluğu grafiğinden anlaşılacağı üzere 80-100 arasındaki değerlerde 1 tahmin etme olasılığının arttığı gözlemlenmiştir.



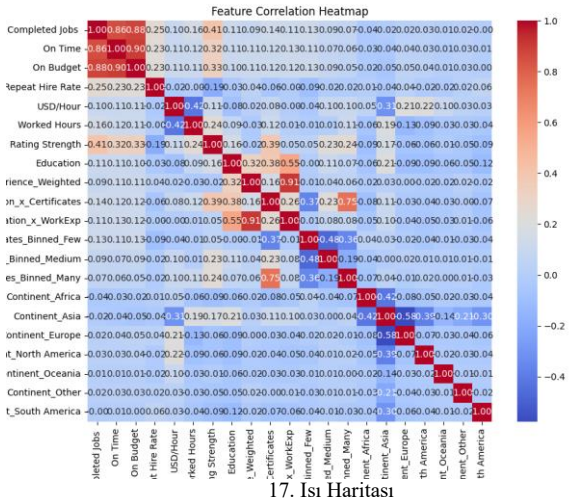
15. On time özelliği grafikleri

15 numaralı görselde verilerin dağılımı incelendiğinde modelin on time özelliğinin tek başına yeterli olmadığı gözlemlenmiştir.



16. USD/Hour özelliği ile ilgili grafikler

USD/Hour değeri düşük olduğunda başarılı olma olasılığının yüksek olduğu düşünülebilir. Ama sonuç bize bunun doğru olmadığını kullanıcının istediği tutarın miktarının fazla olması başarılı ya da başarısız olma durumunu o kadar çok etkilemediği anlaşılmıştır.



17. Isı Haritası

Bu ısı haritası her özelliğin birbirle olan etkileşimini göstermektedir. Haritada kutucuk kızıllaştıkça o iki özelliğin birbirleriyle arasındaki korelasyon katsayısı artmakta, yani aralarındaki ilişki güçlenmektedir.

V. SONUÇ OLARAK

Bu çalışmada freelancer kullanıcıların verileri incelenmiştir. Çeşitli özellikler kullanılarak modeller eğitilmiştir. En başarılı olan random forest modeli geliştirilmiş. Elde edilen sonuçlar recall, precision, accuarcy, f1 skor ve auc-roc metrikleri ile değerlendirilmesi yapılmıştır. Modelin ürettiği sonuçlar ve özellikler arasında ilişki grafikler ile görselleştirilmiş ve açıklanmıştır. Elde edilen bulgular, serbest çalışanların hangi profil özelliklerinin kabul edilme olasılığını artırabileceği konusunda yol gösterici bilgiler sunmaktadır.

ACKNOWLEDGMENT

Bu projede developerlar için API sağlayan freelancer.com'a teşekkür ederim.

REFERENCES

[1] *"Freelancer – Hire & Find Jobs," Freelancer.com. [Online]. Available: https://www.freelancer.com/. [Accessed: May 24, 2025].*

[2] *"Random Forest Classifier — scikit-learn 1.4.2 documentation," Scikit-learn.org. [Online]. Available: https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.RandomForestClassifier.html.*

[3] ["Gradient Boosting — scikit-learn 1.4.2 documentation," Scikit-learn.org. [Online]. Available: https://scikit-learn.org/stable/modules/ensemble.html#gradient-boosting.

[4] "Classification — scikit-learn 1.4.2 documentation," Scikit-learn.org. [Online]. Available: https://scikit-learn.org/stable/supervised_learning.html.

[5] G. van Rossum and F. L. Drake, *Python 3 Reference Manual*, Scotts Valley, CA: CreateSpace, 2009.

[6] F. Pedregosa *et al*., "Scikit-learn: Machine Learning in Python," *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.

[7] G. Lemaître, F. Nogueira, and C. K. Aridas, "Imbalanced-learn: A Python Toolbox to Tackle the Curse of Imbalanced Datasets in Machine Learning," *Journal of Machine Learning Research*, vol. 18, no. 17, pp. 1–5, 2017.

[8] W. McKinney, "Data Structures for Statistical Computing in Python," in *Proc. 9th Python in Science Conf.*, 2010, pp. 51–56.

[9] J. D. Hunter, "Matplotlib: A 2D Graphics Environment," *Computing in Science & Engineering*, vol. 9, no. 3, pp. 90–95, 2007.

[10] M. L. Waskom, "Seaborn: Statistical Data Visualization," *Journal of Open Source Software*, vol. 6, no. 60, p. 3021, 2021.