**Veri Analizi ve Makine Öğrenmesi**

**Giriş**

Bu çalışmada, penguen türlerini tahmin etmek amacıyla makine öğrenmesi yöntemleri adım adım uygulanmıştır. Kaggel’ dan indirilen veri seti, Palmer Penguins Dataset olup üç farklı türü (Adelie, Gentoo ve Chinstrap) içermektedir. Projenin temel amacı, makine öğrenmesi sürecinin uçtan uca deneyimlemesi ve farklı ön işleme teknikleri ile model performanslarının karşılaştırılmasını sağlamaktır.

Çalışma kapsamında öncelikle veri keşfi (EDA) gerçekleştirilmiş, eksik ve aykırı değerler incelenmiş, ardından ön işleme adımları (encoding, scaling) uygulanmıştır. Daha sonra farklı modeller (Logistic Regression, XGBoost) denenmiş, elde edilen sonuçlar karşılaştırılmış ve en iyi performansı veren model belirlenmiştir. Nihai olarak, kullanıcıdan alınan girdi değerleri üzerinden canlı tahminler yapabilen bir sistem tasarlanmış ve terminal üzerinde test edilmiştir.

**1.Veri Keşfi (Exploratory Data Analysis - EDA)**

Veri keşfi aşamasında, öncelikle veri setinin genel yapısı incelenmiştir. Veri setinde toplamda üç farklı penguen türü (**Adelie, Gentoo, Chinstrap**) bulunmaktadır. Her birey için **island,** **bill\_length\_mm**, **bill\_depth\_mm**, **flipper\_length\_mm**, **body\_mass\_g**, **sex, health\_metrics, life\_stage** ve **year** gibi özellikler kayıt altına alınmıştır.Türlerin dengeli bir şekilde dağılıp dağılmadığı incelenmiş, grafiklerle görselleştirilmiştir.

metin, ekran görüntüsü, yazı tipi, siyah içeren bir resim

Yapay zeka tarafından oluşturulmuş içerik yanlış olabilir.Veri seti, penguenlerin biyolojik ve çevresel özelliklerini içermektedir. Aşağıda veri setinden örnek satırlar verilmiş, veriyle ilgili ilk izlenim oluşmuştur.

Tabloda görüldüğü üzere her satır bir penguene ait özellikleri ifade etmektedir. **Tür (species)** sütunu hedef değişken olup, diğer sütunlar bağımsız değişkenler olarak kullanılmaktadır.

Sayısal Değişkenlerin Tanımlayıcı İstatistikler .

metin, ekran görüntüsü, yazı tipi, sayı, numara içeren bir resim

Yapay zeka tarafından oluşturulmuş içerik yanlış olabilir.Bu tablo, veri setindeki sayısal değişkenlerin dağılımını genel hatlarıyla özetlemektedir. Ortalama, minimum, maksimum ve çeyreklik değerler sayesinde verilerin merkezi eğilimi ve yayılımı hakkında bilgi edinilmiştir. Ayrıca standart sapma değerleri, değişkenlerin ne kadar farklılık gösterdiğini ortaya koymaktadır.

**Aykırı Değer Analizi**

metin, ekran görüntüsü, diyagram, çizgi içeren bir resim

Yapay zeka tarafından oluşturulmuş içerik yanlış olabilir.

Görseldeki kutu grafiği (boxplot), **vücut ağırlığı (body\_mass\_g)** değişkenindeki dağılımı ve aykırı değerleri göstermektedir.

Aykırı değerlerin bulunması, modelleme aşamasında dikkat edilmesi gereken bir durumdur. Bu tür değerler modelin öğrenme sürecini olumsuz etkileyebileceği için ilerleyen adımlarda ya temizlene ya da dönüştürme işlemi yapılabilir.

Aykırı değerlerin tespiti IQR algoritmasıyla yapılmıştır

**2.Ön İşleme (Preprocessing)**

Veri keşfi (EDA) aşamasında, verilerin genel dağılımı ve özellikleri incelenmiş, olası eksiklikler ve aykırı değerler gözlemlenmiştir. Ancak EDA yalnızca veriyi **anlama ve yorumlama** aşamasıdır.

Bunun bir sonraki adımı olan **ön işleme (preprocessing)** ise, veriyi **makine öğrenmesi modellerinin anlayabileceği forma dönüştürme** sürecidir. Bu aşamada:

Eksik değerler temizlenir veya doldurulur.

Kategorik veriler sayısal formata çevrilir (**encoding**).

Aykırı değerler (outlier) kontrol edilip gerekirse işlenir.

Sayısal veriler aynı ölçeğe getirilir (**scaling**).

EDA bize veriyi tanıtırken Preprocessing ise veriyi model için hazırlar.

**One-Hot Encoding**

Veri setindeki kategorik değişkenler (**island, sex, diet, life\_stage, health\_metrics**) doğrudan sayısal değerlerle ifade edilemez. Bu nedenle **One-Hot Encoding** yöntemi uygulanmıştır. Bu yöntem, her kategori için yeni bir sütun oluşturur ve gözlemin o kategoriye ait olup olmadığını **True/False (1/0)** ile belirtir.

metin, ekran görüntüsü, yazı tipi, yazılım içeren bir resim

Yapay zeka tarafından oluşturulmuş içerik yanlış olabilir.Bu dönüşüm sayesinde, model her adayı ayrı bir özellik olarak değerlendirebilmektedir. Böylece kategorik değişkenler, modelin öğrenme sürecine katkı sağlayacak şekilde sayısallaştırılmıştır (True-False).

**Label Encoding**

Label Encoding yönteminde, kategorik değişkenler her kategoriye **sayısal bir etiket (integer)** verilerek dönüştürülür. Örneğin:

island değişkeni için:

Biscoe → 0 Dream → 1 Torgersen → 2

Bu yöntem, özellikle **sıralı (ordinal)** kategorik değişkenlerde oldukça faydalıdır çünkü doğal bir “küçükten büyüğe” ilişkisi vardır. Ancak **sırasız (nominal)** değişkenlerde model, sayılar arasında anlamlı bir büyüklük ilişkisi olduğunu varsayabilir ve bu durum hatalara yol açabilir.

Biz modellerdeki sonucu gözlemlemek amacıyla verilerimizi hem label ile hem de one -hot ile ayrı ayrı encode ediyoruz.

**Ölçeklendirme (Scaling)**

Veri setinde yer alan sayısal değişkenlerin ( **bill\_length\_mm, bill\_depth\_mm, flipper\_length\_mm, body\_mass\_g**) ölçüm aralıkları birbirinden farklıdır. Örneğin:

Gaga uzunluğu değerleri **20–60 mm** arasında,

Vücut ağırlığı değerleri ise **2500–10000 gr** arasında değişmektedir.

Bu tür farklı ölçekler, özellikle **mesafe tabanlı algoritmalar** veya **gradient tabanlı yöntemler (Logistic Regression, XGBoost)** için modelin öğrenmesini zorlaştırabilir. Çünkü büyük sayılara sahip değişkenler, küçük ölçekli değişkenlerin önemini gölgeleyebilir.

Bu nedenle, sayısal sütunlar **StandardScaler** yöntemi ile dönüştürülmüştür.

Bu yöntem sayesinde tüm sayısal sütunlar aynı seviyeye çekilmiş ve modelin öğrenme süreci dengelenmiştir.

**3. Modelleme**

**3.1 Logistic Regression**

Veri ön işleme adımlarından sonra ilk olarak **Logistic Regression** modeli tercih edilmiştir. Bunun sebebi:

Basit ve hızlı uygulanabilir olması,

Çıktılarının kolay yorumlanabilmesi,

Başlangıç için **baseline accuracy** (referans doğruluk) sağlamasıdır.

**Veri Ayrımı (Train-Test Split)**

Veri seti, modelin performansını doğru şekilde ölçebilmek için eğitim ve test olmak üzere **%80 eğitim** (train) **%20 test** ikiye ayrılmıştır:

Bu sayede model, verinin bir kısmıyla eğitilmiş, geri kalanıyla test edilerek gerçekçi performans ölçümü yapılmıştır.

**Model Sonuçları**

**metin, ekran görüntüsü, yazı tipi içeren bir resim

Yapay zeka tarafından oluşturulmuş içerik yanlış olabilir.**Model, hem **Label Encoding** hem de **One-Hot Encoding** ile ayrı ayrı eğitilmiştir.

**Label Encoding :**

Doğruluk (Accuracy): **%71.6**

Özellikle “1” sınıfında (örneğin Chinstrap) precision ve recall değerleri oldukça düşük çıkmıştır. Bu da modelin bu sınıfı ayırt etmekte zorlandığını göstermektedir.

**One-Hot Encoding :**

Doğruluk (Accuracy): **%87.8**

Tüm sınıflarda precision, recall ve f1-score değerleri daha dengeli ve yüksek bulunmuştur.

**Değerlendirme**

Logistic Regression modeli, başlangıç için beklenen şekilde hızlı sonuç vermiştir. Label Encoding kullanıldığında performans düşük kalırken, One-Hot Encoding ile belirgin bir iyileşme elde edilmiştir. Değişkenlerde One-Hot Encoding yönteminin Logistic Regression için daha uygun olduğunu göstermektedir.

**Veri Dengesizliği ve Oversampling**

Veri setinde türler arasında dengesizlik bulunmaktadır. Örneğin:

Adelie (0): 1248 gözlem

Gentoo (2): 998 gözlem

Chinstrap (1): 498 gözlem

Bu durum, modelin az sayıda gözleme sahip sınıfları öğrenmesini zorlaştırabilir. Özellikle **Chinstrap türü** için modelin precision ve recall değerlerinin düşük çıkmasının olabilir.

**SMOTE (Synthetic Minority Oversampling Technique)**

Bu problemi azaltmak için **SMOTE** yöntemi uygulanmıştır. SMOTE, azınlık sınıflarına ait gözlemlerden yeni yapay örnekler (sentetik veriler) üreterek sınıf dağılımını dengeler. Böylece:

Modelin her sınıf için daha fazla örnek görmesi sağlanır,

Dengesizliğin neden olduğu yanlılık (bias) azaltılır.

Uygulama sonucunda, tüm sınıflar **1248 örnek** olacak şekilde dengelenmiştir.

metin, ekran görüntüsü, yazı tipi içeren bir resim

Yapay zeka tarafından oluşturulmuş içerik yanlış olabilir.Uygulama sonucunda, tüm sınıflar **1248 örnek** olacak şekilde dengelenmiştir.

**Sonuç**

SMOTE sonrasında model tekrar eğitilmiş, ancak sonuçlarda kayda değer bir iyileşme görülmemiştir.

**3.2 XGBoost**

Logistic Regression’dan sonra, daha güçlü ve karmaşık bir algoritma olan **XGBoost (Extreme Gradient Boosting)** uygulanmıştır. XGBoost, karar ağaçları üzerine kurulu, optimizasyon gücü yüksek ve sınıflandırma problemlerinde oldukça başarılı bir algoritmadır.

**Model Sonuçları**

Model hem Label Encoding hem de One-Hot Encoding ile çalıştırılmıştır:

**metin, ekran görüntüsü içeren bir resim

Yapay zeka tarafından oluşturulmuş içerik yanlış olabilir.Label Encoding**

Doğruluk (Accuracy): **%86.6**

Özellikle “2” sınıfında (örneğin Gentoo) **%97 recall** ile çok yüksek bir başarı elde edilmiştir.

“1” sınıfında recall (%75) görece daha düşük olsa da Logistic Regression’a kıyasla belirgin bir iyileşme sağlanmıştır.

**One-Hot Encoding**

Doğruluk (Accuracy): **%87.6**

Tüm sınıflarda precision, recall ve f1-score değerleri daha dengeli bir dağılım göstermiştir.

Özellikle “2” sınıfında **%97 recall** korunmuş, “1” sınıfında da Logistic Regression’a göre daha iyi sonuçlar elde edilmiştir.

**Değerlendirme**

XGBoost modeli, Logistic Regression’a kıyasla daha yüksek doğruluk ve iki farklı encode yönteminde de (label – onehot) daha dengeli sınıf performansı sağlamıştır. Özellikle One-Hot Encoding ile elde edilen sonuçlar, projenin en iyi performansına ulaşmıştır (**%87.6 doğruluk**).

**Feature Selection (Özellik Seçimi)**

Model eğitiminde hangi değişkenlerin daha baskın olduğunu görmek için **özellik önem sıralaması (feature importance)** incelenmiştir.

**One-Hot Encoding ile:**

En baskın özellikler adaya ait değişkenler olmuştur (**island\_Torgensen, island\_Biscoe, island\_Dream**).

**Label Encoding ile:**

Tek bir “island” özelliği baskın çıkmış, ardından cinsiyet, yaşam evresi ve diyet gelmiştir.

**Değerlendirme:**

Her iki yöntemle de en kritik özelliğin **penguenin yaşadığı ada** olduğu görülmüştür. Bu, coğrafi konumun tür ayrımında güçlü bir belirleyici olduğunu göstermektedir. One-Hot Encoding ile özellikler daha ayrıntılı işlendiği için, modelin performansı bir miktar daha yüksek çıkmıştır.

**4. Final Model ve Tahmin (Prediction)**

metin, ekran görüntüsü, yazı tipi, tasarım içeren bir resim

Yapay zeka tarafından oluşturulmuş içerik yanlış olabilir.Yapılan karşılaştırmalar sonucunda, en iyi performansı **XGBoost modeli** ve **One-Hot Encoding** ile elde ettiğimiz için çalışmanın devamında bu kombinasyon tercih edilmiştir.

Model, diğer yöntemlere göre daha doğru sonuçlar verdiği için **dosyaya kaydedilmiştir**. Böylece model her defasında yeniden eğitilmeden doğrudan yüklenip kullanılabilmektedir.

Tahmin adımında, terminal üzerinden kullanıcıdan **girdi değerleri (örneğin gaga uzunluğu, kanat uzunluğu, vücut ağırlığı, ada bilgisi vb.)** alınmıştır. Kaydedilen model dosyası okunarak bu veriler işlenmiş ve model ilgili penguenin türünü (**Adelie, Gentoo, Chinstrap**) tahmin etmiştir.Tahminlerimiz için girdiğimiz verileriPalmerpenguins\_original.csv dosyasından yapmaktayız

**5. Tahmin Sonuçları ve Değerlendirme**

Eğitilen model, terminal üzerinden girilen verilerle test edilmiştir. Tahminlerde bazı doğru sonuçlar elde edilmiş olsa da, özellikle **Adelie türü** için hatalı sınıflandırmalar dikkat çekmiştir.

* **Doğru tahminler (+):** Çoğunlukla Chinstrap (Dream adası) ve Gentoo (Biscoe adası) türlerinde yüksek doğruluk elde edilmiştir.
* **Yanlış tahminler (-):** Adelie türünde hatalar yoğunlaşmıştır. Model bazı Adelie bireylerini Gentoo veya Chinstrap olarak tahmin etmiştir.

**Veri Dağılımının Etkisi**

Original CSV incelendiğinde:

* **Adelie türü** → üç adada da (Torgensen, Biscoe, Dream) gözlemlenmektedir.
* **Gentoo türü** → yalnızca Biscoe adasında bulunmaktadır.
* **Chinstrap türü** → yalnızca Dream adasında bulunmaktadır.

Bu dağılım, modelin özellikle **ada bilgisini aşırı derecede belirleyici** (dominant feature) olarak kullanmasına yol açmıştır. Sonuç olarak model, şu şekilde basit kurallar öğrenmiştir:

* Biscoe → Gentoo
* Dream → Chinstrap
* Torgensen → Adelie

Bu durumda, modelin **Underfitting** eğilimi gösterdiğini tahmin etmekteyim. **Underfitting (yetersiz öğrenme)** model verinin karmaşıklığını yakalayamıyor, basit kurallara sıkışıyor (ör. sadece ada bilgisine göre karar veriyor).

Bu durumu düzeltmek amacıyla ada sütunu baskılanılmaya çalışıldı ama model tahmin sırasında her sonucu gentoo türü olarak vermeye başladı.

Veriden island sütunu komple çıkarılmaya karar verildi ama bu sefer hem accuary, recall ve precision değerlerinde ciddi bir düşüş yaşandı hem de bu durum tahmin kısmında sağlıklı sonuçlar alamamamıza neden oldu.

**Sonuç**

Model genel olarak %87 doğruluk seviyesine ulaşsa da, Adelie türü üzerindeki hatalar ve ada bilgisinin aşırı baskınlığı önemli bir sınırlılık yaratmaktadır. Daha dengeli bir model için:

* Gerçek biyolojik farklılıkları (gaga, kanat, kilo ölçüleri) daha ön plana çıkaracak yöntemler denenebilir,
* Daha geniş ve çeşitlendirilmiş veri setleri ile model yeniden eğitilebilir.