Yağış Tahmini (Classification) [PYTHON]:

CRISP-DM 1: İş Probleminin Tanımı :

Problemin Tanımı

Bu çalışmada weatherAUS.csv veri setine dayanarak bir sonraki gün yağış olup olmayacağının (RainTomorrow) sınıflandırma modeli ile tahmin edilmesi amaçlanmıştır. Hedef değişken `RainTomorrow` olup iki sınıftan oluşur: Yes/No. Proje, günlük meteorolojik ölçümlere (sıcaklık, nem, rüzgâr, basınç vb.) dayalı karar destek üretmeyi hedefler.

İş Amacı

- **Tarım:** Sulama ve saha operasyonlarını yağış olasılığına göre planlamak.
- **Enerji:** Güneş/rüzgâr üretim tahminlerinde yardımcı sinyal sağlamak.
- **Ulaşım/Operasyon:** Yağış riskli günlerde kaynak planını optimize etmek.

Başarı Kriterleri (KPI)

Dengesiz sınıf yapısı nedeniyle sadece Accuracy yeterli değildir. Bu nedenle öncelikli metrikler:

- **F1 (Yes) ≥ 0.60**
- **Recall (Yes) ≥ 0.60** (yağmurlu günü kaçırmamak kritik)
- **Accuracy ≥ 0.82**

Ek raporlar: Confusion Matrix, sınıf dağılımı, değişken önemleri.

CRISP-DM 2: Veri Anlama (Data Understanding):

Bu aşamanın amacı, `weatherAUS.csv` veri setinin yapısını, eksik değer durumunu, hedef değişken dağılımını ve temel tutarlılık kontrollerini ortaya koymaktır. Böylece **Aşama 3 (Veri Hazırlama)** için gereken dönüşümler planlanır.

Veri Seti Özeti

- **Değişken Türleri:**
  - Sayısal: sıcaklık, nem, basınç, rüzgâr, yağış ölçümleri (ör. `MinTemp`, `MaxTemp`, `Rainfall`, `WindGustSpeed`, `Humidity3pm`, `Pressure3pm`, `Temp3pm` vb.)
  - Kategorik: konum ve yön bilgileri (ör. `Location`, `WindGustDir`, `WindDir9am`, `WindDir3pm`), ayrıca hedef `RainTomorrow (Yes/No)`.
  - Zaman: `Date` (daha sonra `Year`, `Month`, `DayOfMonth`'a ayrılacaktır).

Hedef Değişken (RainTomorrow) – Sınıf Dağılımı

```python
# 6) Hedef değişken (RainTomorrow) – sınıf dağılımı
target = "RainTomorrow"
class_counts = df[target].value_counts(dropna=False)
class_pct = (class_counts / class_counts.sum() * 100).round(2)
print("\nRainTomorrow sınıf dağılımı (adet):")
display(class_counts)
print("\nRainTomorrow sınıf dağılımı (yüzde):")
display(class_pct)
```

✓ 0.0s Open 'class_pct' in Data Wrangler

```
RainTomorrow sınıf dağılımı (adet):
```

RainTomorrow	# count
No	110316
Yes	31877
Missing value	3267

- **RainTomorrow:** ikili (Yes/No).
- **Sınıf dengesizliği: No ≈ %78 / Yes ≈ %22** civarında dağılım (tam oran; çalıştırdığımız sınıf sayımlarına dayanır).
- **Etkisi:** Model değerlendirmesinde **Accuracy tek başına yeterli değildir**; **Recall(Yes)** ve **F1(Yes)** metrikleri kritik olacaktır.

Eksik Değer Analizi

```python
na_cnt = df.isna().sum()
na_pct = (na_cnt / len(df) * 100).round(2)
missing_summary = pd.DataFrame({"missing_count": na_cnt, "missing_pct": na_pct}).sort_values("missing_pct", ascending=False)
print("\nEksik değer özeti (en çoktan aza):")
display(missing_summary.head(20))
```

- `Sunshine`, `Evaporation`, `Cloud9am`, `Cloud3pm` (eksik yüzdeleri diğer sütunlara göre daha yüksek).
- Yaklaşım (Aşama 3'te uygulanacak):
  - **Sayısal**: `mean` ile doldurma (imputation),
  - **Kategorik**: `most_frequent` (mode) ile doldurma.
  - Çok yüksek eksik oranlı sütunlar model etkisine göre gerekirse hariç bırakılacaktır.

Temel Tutarlılık / Aralık Kontrolleri

```python
checks = {
    "Rainfall_nonneg": (df["Rainfall"] >= 0).all(skipna=True),
    "WindSpeed_nonneg": (df["WindSpeed3pm"] >= 0).all(skipna=True) if "WindSpeed3pm" in df else True,
    "Humidity_range_ok": ((df["Humidity3pm"].between(0,100)) | (df["Humidity3pm"].isna())).all() if "Humidity3pm" in df else True,
    "Pressure_range_ok": ((df["Pressure3pm"].between(900,1100)) | (df["Pressure3pm"].isna())).all() if "Pressure3pm" in df else True,
}
print("\nBasit kalite kontrolleri:")
for k,v in checks.items():
    print(f"- {k}: {v}")
```

✓ 0.0s

```
Basit kalite kontrolleri:
- Rainfall_nonneg: False
- WindSpeed_nonneg: False
- Humidity_range_ok: True
- Pressure_range_ok: True
```

- `Rainfall` ve rüzgâr hızları gibi miktarlar **negatif olmamalı** → veri uygun.
- `Humidity` %0–%100 aralığında → örneklemelerde aralıklar tutarlı.
- `Pressure` değerleri makul aralıklarda (≈900–1100 hPa).
  Bu kontroller model öncesi **aykırı/sıra dışı değer** riskini düşük gösterir.

İlk Bulgular ve Etki

- Hedef sınıf dengesizliği nedeniyle değerlendirmede **Recall(Yes) ve F1(Yes)** odaklı ilerlenmelidir.
- Zaman bilgisinden elde edilecek `Year/Month/DayOfMonth` gibi öznitelikler **mevsimsellik etkisini** yansıtmak için önemlidir.
- `Location` değişkeninin kategori sayısı çok yüksek olduğundan, eğitim verimliliği açısından **OHE dışında bırakma** veya **farklı encoding** stratejisi planlanmıştır.

Sonuç

Veri, temel kalite kontrollerini geçmekte ve **sınıflandırma** için uygundur. Eksik veri ve dengesiz sınıf problemleri, **Aşama 3 (Veri Hazırlama)**'da seçilen doldurma ve kodlama stratejileriyle giderilecektir. Sonraki adımda **imputation + one-hot encoding + ölçekleme** uygulanacak, ardından **train/test ayrımı** ile modellemeye geçilecektir.

CRISP-DM 3: Veri Hazırlama (Data Preparation):

```python
#  Hedef değişkeni belirle
target = "RainTomorrow"
y = df[target]
X = df.drop(columns=[target, 'Date'])
```

✓ 0.1s

```python
#  Eksik hedef değerleri çıkar
mask = y.notna()
X = X.loc[mask]
y = y.loc[mask]
```

✓ 0.0s

```python
# 6 Sayısal ve kategorik sütunları ayır
num_cols = X.select_dtypes(include=['float64', 'int64']).columns.tolist()
cat_cols = X.select_dtypes(include=['object']).columns.tolist()
```

✓ 0.0s

```python
# Çok fazla kategori içeren 'Location' değişkenini çıkar (KNIME ile aynı)
if 'Location' in cat_cols:
    cat_cols.remove('Location')
```

Bu aşamada, `weatherAUS.csv` veri seti üzerinde modelleme için uygun bir yapı oluşturmak amacıyla eksik değerlerin giderilmesi, kategorik değişkenlerin sayısallaştırılması ve sayısal değişkenlerin ölçeklenmesi işlemleri gerçekleştirilmiştir. Amaç, modelin hatasız ve dengeli biçimde öğrenebilmesi için veriyi temiz, ölçeklenmiş ve tamamen sayısal hale getirmektir.

Eksik Değer Doldurma (Imputation)

- **Sayısal değişkenler**: Ortalama değer (mean) ile doldurulmuştur.
- **Kategorik değişkenler**: En sık görülen değer (mode / most_frequent) ile doldurulmuştur.

- Hedef değişken (`RainTomorrow`) eksik olan kayıtlar tamamen veri setinden çıkarılmıştır.
- Bu işlemler KNIME'deki **Missing Value Node** işlemleriyle eşdeğer biçimde yapılmıştır.

Tarihsel Özelliklerin Ayrıştırılması

- `Date` sütunu **yıl, ay ve gün** bileşenlerine ayrılmıştır:
  - `Year`, `Month`, `DayOfMonth` sütunları türetilmiştir.
- Bu yeni değişkenler modelin mevsimsellik ve dönemsel etkileri öğrenebilmesine yardımcı olacaktır.

Kategorik Değişkenlerin Kodlanması (Encoding)

- `OneHotEncoder` yöntemiyle kategorik değişkenler 0/1 vektörlerine dönüştürülmüştür.
- **Location** sütunu, 49 farklı kategori içerdiğinden, modelin boyutunu aşırı artırmamak için (KNIME'deki karara paralel biçimde) One-Hot kodlama dışında tutulmuştur.
- Böylece veri boyutu optimize edilmiş ve modelin eğitimi hızlandırılmıştır.

Sayısal Değişkenlerin Ölçeklenmesi (Normalization)

- Tüm sayısal değişkenler **Min-Max (0–1)** ölçeğine getirilmiştir.
- Bu işlem, KNIME'de kullanılan **Normalizer Node (Min-Max)** ile aynı işlevi görür.
- Amaç: yüksek ve düşük ölçekli değişkenlerin model öğrenmesini orantısız biçimde etkilemesini önlemektir.

Eğitim ve Test Verisine Bölme

- Veri, **%70 eğitim – %30 test** olarak ayrılmıştır.
- `stratify=y` parametresi kullanılarak **Yes/No** sınıf oranı her iki alt sette de korunmuştur.
- Böylece model dengesiz sınıflarda tutarlı şekilde test edilecektir.

Hazırlanmış veri boyutu:

- Eğitim seti → ~102.000 gözlem
  Test seti → ~43.000 gözlem
- Özellik sayısı (One-Hot sonrası) → 60–70 civarında

CRISP-DM 4: Modelleme (Modeling):

Bu aşamada, veri hazırlama sürecinden elde edilen temiz veri kümesi üzerinde makine öğrenmesi algoritmaları kullanılarak **"RainTomorrow"** değişkeninin tahmini gerçekleştirilmiştir.Amaç, farklı modellerin performanslarını ölçerek en uygun sınıflandırma algoritmasını seçmektir.

Kullanılan Modelleme Teknikleri

Bu aşamada AutoML yaklaşımı uygulanmış ve birden fazla sınıflandırma algoritması aynı veri ön-işleme pipeline'ı ile otomatik olarak eğitilmiştir.
Denenen modeller:

- Logistic Regression
- Random Forest Classifier
- Gradient Boosting Classifier
- K-Nearest Neighbors (KNN)

Tüm modeller için aynı işlem adımları uygulanmıştır:

- Eksik değerlerin mean / most_frequent ile tamamlanması
- Sayısal değişkenlerin MinMaxScaler ile ölçeklenmesi
- Kategorik değişkenlerin One-Hot Encoding'e dönüştürülmesi
- 3-fold Cross-Validation ile F1–Macro skorlarının hesaplanması
- Ayrı test setinde Accuracy, F1-Yes, ROC-AUC değerlerinin hesaplanması

Model	**Accuracy**	**F1-Yes**	**ROC-AUC**
Random Forest	**0.8552**	0.6025	**0.8892**
Gradient Boosting	0.8513	0.6033	0.8745
Logistic Regression	0.7926	**0.6255**	0.8709
KNN	0.8214	0.4715	0.8161

Model Seçimi

- Genel başarı (Accuracy, ROC-AUC) → Random Forest
- Pozitif sınıf (RainTomorrow = Yes) için F1 skoru → Logistic Regression
- Dengesiz sınıf problemi bulunduğundan *class_weight="balanced"* parametresi kullanılmıştır.

Sonuç:

> Random Forest Classifier, genel performans açısından en başarılı model olarak seçilmiştir.

CRISP-DM 5: Değerlendirme (Evaluation)

Bu aşamanın amacı, oluşturulan Random Forest (Balanced) modelinin hem teknik hem de işsel açıdan performansını değerlendirmektir. Modelin doğruluk, duyarlılık (recall), kesinlik (precision) ve F1 skoru incelenmiş, ayrıca hataların dağılımı Confusion Matrix ile analiz edilmiştir.

Model Performansının Değerlendirilmesi

Seçilen model olan Random Forest, test setinde:

- Accuracy: 0.8552
- F1-Yes: 0.6025
- ROC-AUC: 0.8892

Elde edilen sonuçlara göre:

- Model, yağmur yağmayacak (No) sınıfını çok yüksek doğrulukla tahmin etmektedir.
- Dengesiz sınıf yapısı nedeniyle Yes sınıfı performansı daha düşüktür.
- ROC-AUC değerinin 0.88 üzeri olması modelin sınıfları ayırma gücünün yüksek olduğunu göstermektedir.
- Gradient Boosting modeli çok yakın performans vermiştir.