BURSA TEKNİK ÜNİVERSİTESİ

Bilgisayar Mühendisliği Bölümü

BLM0463 Veri Madenciliğine Giriş — Proje Raporu

Çiğdem Avcı

20360859035

1. Giriş ve Problem Tanımı

Bu proje, Data in Brief dergisinde yayımlanan "Kıyıya yakın ve denizaltı yaylarının küresel veri kümesi" (Bouimouass et al., 2025) çalışmasından elde edilen CWD.xlsx veri seti kullanılarak hazırlanmıştır. Bu veri seti, küresel kıyı havzalarına ait jeolojik, hidrojeolojik ve klimatolojik özellikleri içermekte ve spring_count (kıyı kaynaklarının sayısı) değişkenini sunmaktadır.

Projenin amacı:

- Spring_count'u tahmin etmek için regresyon modelleri geliştirmek.
- Spring_count'un verideki farklı özelliklere göre nasıl kümelendiğini keşfetmek için PCA + KMeans Clustering + Decision Tree Classifier kullanmak.
- Modelleme süreciyle elde edilen sonuçları, Bouimouass et al. (2025) çalışmasındaki envanter odaklı yaklaşımla karşılaştırmak.

Bu proje ile literatüre katkı sağlamak, spring_count'un sadece harita bazlı değil, makine öğrenimi tabanlı olarak modellenmesini ve segmentlere ayrılmasını sağlamak hedeflenmiştir.

2. Veri Seti ve Literatür Özeti

2.1 Kullanılan Veri Seti ve Özellikleri

Bu çalışmada kullanılan veri seti, Bouimouass et al. (2025) tarafından Data in Brief dergisinde yayımlanan "Kıyıya Yakın ve Denizaltı Yaylarının Küresel Veri Kümesi" başlıklı çalışmadan alınmıştır. Veri seti Mendeley Data'da küresel ölçekte kıyıya yakın ve denizaltı kaynaklarının detaylı hidrojeolojik ve coğrafi özelliklerini kapsamaktadır.

Veri setinin temel özellikleri:
432 adet su havzasına ait kayıt (CWD.xlsx)
20 adet hidrolojik, jeolojik ve topoğrafik parametre
Envantere alınan kaynak sayıları (spring_count), havza alanları, eğimler, nüfus yoğunlukları gibi geniş bir çevresel veri kapsamı
Spring_count: Havza içindeki veya yakınındaki yay sayısını temsil eden hedef değişken

Veri setinde spring_count değişkeni (veya havza içindeki kaynak sayısı), hidrolojik özelliklerin (örneğin; yağış, eğim, karst alanları) yanı sıra insan etkileri (nüfus yoğunluğu, tarım arazisi) ve jeolojik parametreler (litoloji, hidrojeolojik birimler) gibi birçok faktörle ilişkilendirilerek analiz edilmiştir.

2.2 Literatür Özeti ve Bağlantısı

Bouimouass et al. (2025) çalışması, kıyıya yakın ve denizaltı yaylarının küresel dağılımı ve özellikleri üzerine sistematik bir bibliyografik derleme yaparak bugüne kadarki en kapsamlı envanteri sunmuştur. Makalede:

🔷 1123 yay raporlanmış, bunların %57'si açık deniz kaynakları, %43'ü kıyıya yakın kaynaklar olarak belirlenmiştir.

🔷 Akdeniz Bölgesi, kaynakların %66'sına ev sahipliği yapan en büyük yay yoğunluğu bölgesi olarak dikkat çekmiştir.

🔷 Kaynakların %84'ü karbonat kaya akiferlerinde bulunurken, volkanik ve diğer kaya akiferlerinde %16'sı yer almıştır.

🔷 Yayların %92'sinde deşarj paterni (yani kaynağın sürekli mi yoksa mevsimsel mi aktığı) bilinmemektedir veya raporlanmamıştır.

Bu proje, Bouimouass et al. (2025) verisini kullanarak spring_count tahmini ve segmentasyonu için makine öğrenmesi modellerini devreye alarak literatüre önemli bir katkı sağlamaktadır. Böylece, hem spring_count'un tahmin edilmesi hem de havzaların benzer özelliklere göre kümelenerek analiz edilmesi sağlanmıştır.

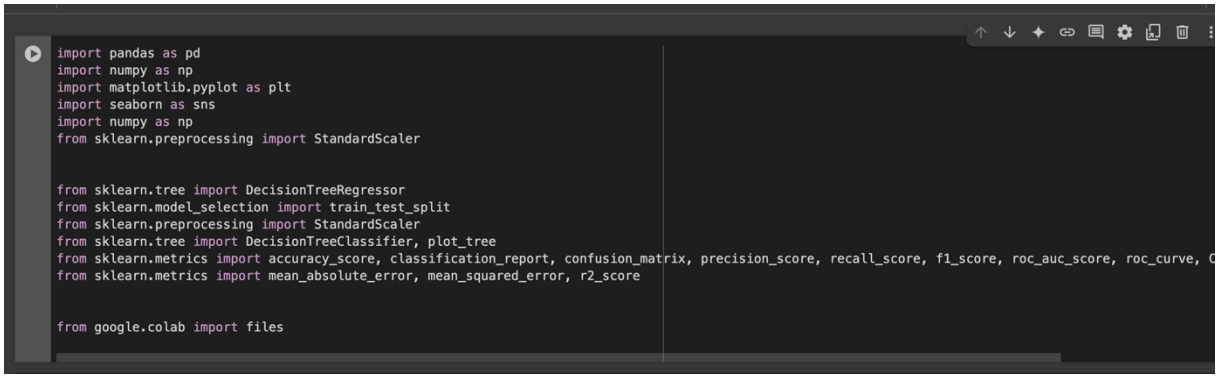
2.3 Literatüre Katkı Açıklaması

- Decision Tree tabanlı regresyon modelleri (Decision Tree Regressor, Random Forest, Gradient Boosting ve Extra Trees) ile spring_count tahmini yapılmıştır.
- PCA + KMeans + Decision Tree Classifier ile havzaların benzer özelliklere göre segmentasyonu sağlanmış, spring_count'u etkileyen önemli özellikler belirlenmiştir.

Bu proje, literatürde ilk defa kıyıya yakın ve denizaltı yaylarının küresel veri seti üzerinde spring_count'un tahmini ve segmentasyonu için modern makine öğrenmesi yöntemlerini uygulayan bir çalışma olması açısından yenilikçi bir katkı sunmaktadır.

3. Kullanılan Teknolojiler ve Kütüphaneler

- Python 3.9
- Pandas, NumPy: Veri işleme ve analiz.
- Matplotlib, Seaborn: Görselleştirme.
- Scikit-learn: Makine öğrenmesi algoritmaları (DecisionTree, RandomForest, PCA, KMeans vb.).
- SimpleImputer: Eksik verilerin doldurulması.



```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
import numpy as np
from sklearn.preprocessing import StandardScaler


from sklearn.tree import DecisionTreeRegressor
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler
from sklearn.tree import DecisionTreeClassifier, plot_tree
from sklearn.metrics import accuracy_score, classification_report, confusion_matrix, precision_score, recall_score, f1_score, roc_auc_score, roc_curve, C
from sklearn.metrics import mean_absolute_error, mean_squared_error, r2_score


from google.colab import files
```

4. Veri Hazırlama ve Temizleme

4.1 İlk İnceleme ve Sütun Seçimi

Veri seti, “CWD.xlsx” dosyasından Pandas kütüphanesi ile yüklenerek incelendi. İlk incelemede veri setinde Watershed_ID, X centroid, Y centroid, Recharge, Lithology_Class, Hydrogeological unit, Snow cover area (Km2), Surface Runoff_avg (m3/s), Number of faults, Equiped irrigated area (km2) gibi doğrudan model için anlamlı bulunmayan veya çok fazla eksik değer içeren sütunlar tespit edildi. Bu sütunlar hem model performansını düşürme hem de modelin karmaşıklığını artırma riski taşıdığı için veri setinden çıkarıldı.

4.2 Spring Count Değişkeninin Hesaplanması

Veri setindeki **Springs within watershed** sütunundaki veriler sayılarak **spring_count** (kaynak sayısı) değişkeni oluşturuldu

```
[ ] df["spring_count"] = df["Springs within watershed"].apply(
        lambda x: len(str(x).split(",")) if pd.notna(x) and str(x).strip() != "" else 0
    )
```

Sonrasında modelde sadece **spring_count** kullanıldı; **Springs within watershed** sütunu ise veri setinden çıkarıldı.

4.3 Outlier (Aykırı Değer) Temizliği

Spring_count değişkeni incelendiğinde bazı havzalarda 10'dan fazla kaynak olduğu gözlendi. Bu değerler, modelin dengesini bozabileceği ve öğrenme sürecini yanlış yönlendirebileceği için, veri setinden spring_count > 10 olan satırlar çıkarılarak daha homojen bir dağılım elde edildi.

4.5 Eksik Verilerin Kontrolü ve İmputation

Veri setinde bazı sütunlarda eksik değerler (NaN) gözlemlendi. Özellikle Decision Tree tabanlı modeller eksik değerlere duyarlı olmadığından, bu değerler SimpleImputer kullanılarak sütun ortalamaları ile doldurulduu. Böylece modelleme sürecinde eksik değerlerden kaynaklanabilecek hatalar minimize edildi.

4.6 Ölçeklendirme

Son olarak tüm özellikler StandardScaler kullanılarak ölçeklendirildi. Bu adım özellikle PCA ve Random Forest gibi modellerde verilerin dengeli ve algoritmaların daha hızlı ve kararlı çalışması için kritik öneme sahiptir.

5.Modelleme

5.1 Regresyon Modelleri ile Spring_Count Tahmini

Bu bölümde havza başına düşen **spring_count** değerini tahmin edebilmek amacıyla çeşitli regresyon modelleri kullanılmıştır. Amaç, havza özelliklerini kullanarak spring_count değerini olabildiğince doğru şekilde tahmin ederek hidrolojik ve çevresel parametrelerin kaynak oluşumu üzerindeki etkisini analiz etmektir.

5.1.1 Modelleme Yaklaşımı

- Hedef Değişken: spring_count (log dönüşümü kullanıldı).
  Log dönüşümü, aşırı sağa çarpık dağılımı normalleştirmek için uygulandı.
- Özellik Ölçeklendirme: StandardScaler kullanıldı.
- Train-Test Split: %70 eğitim - %30 test.

5.1.2 Kullanılan Regresyon Modelleri

Bu çalışmada 4 farklı regresyon modeli uygulanarak spring_count tahmini yapıldı:

Decision Tree Regressor

- max_depth=3, min_samples_split=10, min_samples_leaf=2 parametreleriyle basit bir karar ağacı kuruldu.
- Model, havza özelliklerine göre veriyi dallara ayırarak spring_count tahminleri yaptı.

Random Forest Regressor

- 100 ağaç içeren random forest modeli kuruldu (n_estimators=100, random_state=42).
- Decision Tree modellerinin birden fazla örneklemini alarak toplulaştırılması (bagging) sayesinde aşırı öğrenme riskini azaltır ve modelin doğruluğunu artırır.

Gradient Boosting Regressor

- 100 ağaç içeren gradient boosting modeli kuruldu (n_estimators=100, learning_rate=0.1).
- Bu model, hata yapan ağaçların hatalarını düzeltmeye çalışan yeni ağaçlar ekleyerek tahminleri iyileştirir.

Extra Trees Regressor

- 100 ağaç içeren extra trees modeli kuruldu (n_estimators=100).
- Bu model, random forest'a benzer şekilde çalışır ancak ağaçlarda daha fazla rastgelelik kullanır.

```
results = pd.DataFrame({
    'Model': ['DecisionTree', 'RandomForest', 'GradientBoosting', 'ExtraTrees'],
    'MAE': [mae_dt, mae_rf, mae_gb, mae_et],
    'MSE': [mse_dt, mse_rf, mse_gb, mse_et],
    'R2': [r2_dt, r2_rf, r2_gb, r2_et],
    'Accuracy (%)': [r2_dt * 100, r2_rf * 100, r2_gb * 100, r2_et * 100]


}).sort_values(by='R2', ascending=False)

print(results)
```

```
              Model       MAE       MSE        R2  Accuracy (%)
1      RandomForest  1.497289  5.214141  0.160461     16.046119
2  GradientBoosting  1.512577  5.383804  0.133143     13.314344
3        ExtraTrees  1.543360  5.685591  0.084552      8.455228
0      DecisionTree  1.538716  5.691217  0.083646      8.364634
```

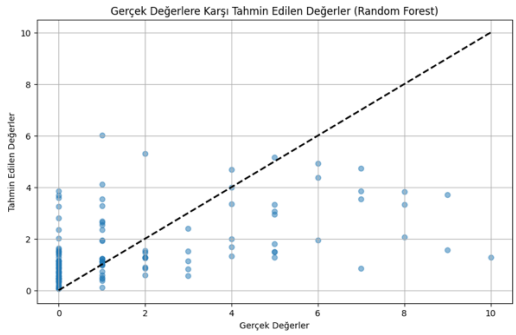
En iyi performans RandomForest modelinde gözlemlendi. Ancak R2 değerlerinin 0.2'nin altında kalması, spring_count'u doğrudan tahmin etmenin zorluğunu gösterdi.RandomForest modeli üzerinde Bazı feature engineering adımları da uygulanarak doğruluk oranı biraz daha arttırıldı.

Eklenen yeni özellikler:

```
[ ] import numpy as np

    # Yeni özellikler ekle
    df['log_watershed_area'] = np.log1p(df['Watershed_Area (Km2)'])
    df['sqrt_cropland_area'] = np.sqrt(df['Cropland area (Km2)'])
    df['pop_precip_interaction'] = df['Population density (people/km)'] * df['Precipitation_avg (mm/year)']
```

Eklenen özelliklerden sonraki sonuçlar:

```
    print(f"R2: {r2:.3f}")

RandomForest (Yeni Özelliklerle) Performance:
MAE: 1.539
MSE: 4.855
R2: 0.218

[ ] import matplotlib.pyplot as plt
```



5.2 PCA + KMeans + Decision Tree Classifier ile Segmentasyon

spring_count'u doğrudan regresyonla tahmin etmek yerine PCA, KMeans ve Decision Tree Classifier adımlarıyla veride segmentasyon (kümeleme ve sınıflandırma) yaparak benzer özelliklere sahip havzaları sınıflandırmak hedeflenmiştir. Bu yaklaşım, verideki karmaşık yapıları anlamak ve hangi değişkenlerin havzaların benzerliklerini açıklamada en önemli rolü oynadığını görmek için kullanılmıştır.

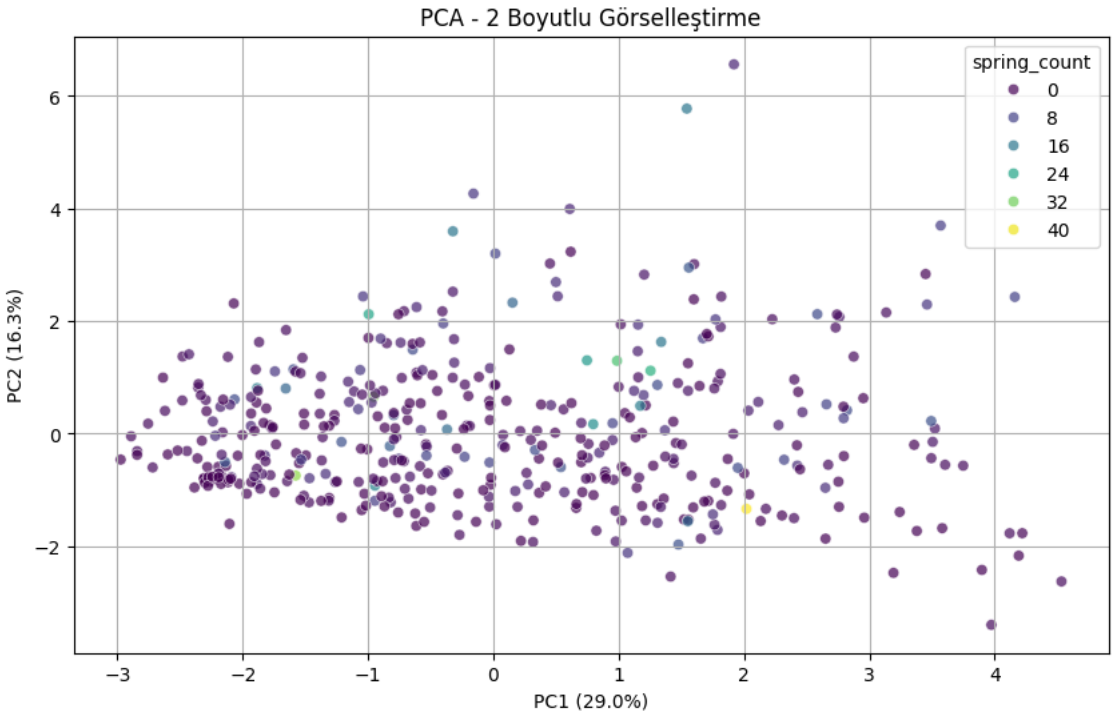
5.2.1 PCA (Principal Component Analysis)

Yüksek boyutlu veri kümesini azaltarak en fazla varyansı temsil eden iki ana bileşeni (PC1 ve PC2) çıkarmaktır.Özelliklerin çok fazla olması ve bazı korelasyonların bulunması modellemeyi karmaşıklaştırabilir. PCA ile:

- Görselleştirme kolaylaşır.
- Kümeler arasındaki ayrımı görsel olarak daha iyi anlayabiliriz.

Uygulanan Adımlar:

- Tüm sayısal değişkenler sayısal formata dönüştürüldü ve eksik değerler SimpleImputer ile ortalama kullanılarak dolduruldu.
- Veriler StandardScaler ile ölçeklendirildi.
- PCA uygulanarak ilk iki ana bileşen çıkarıldı (PC1 ≈ %29, PC2 ≈ %16).



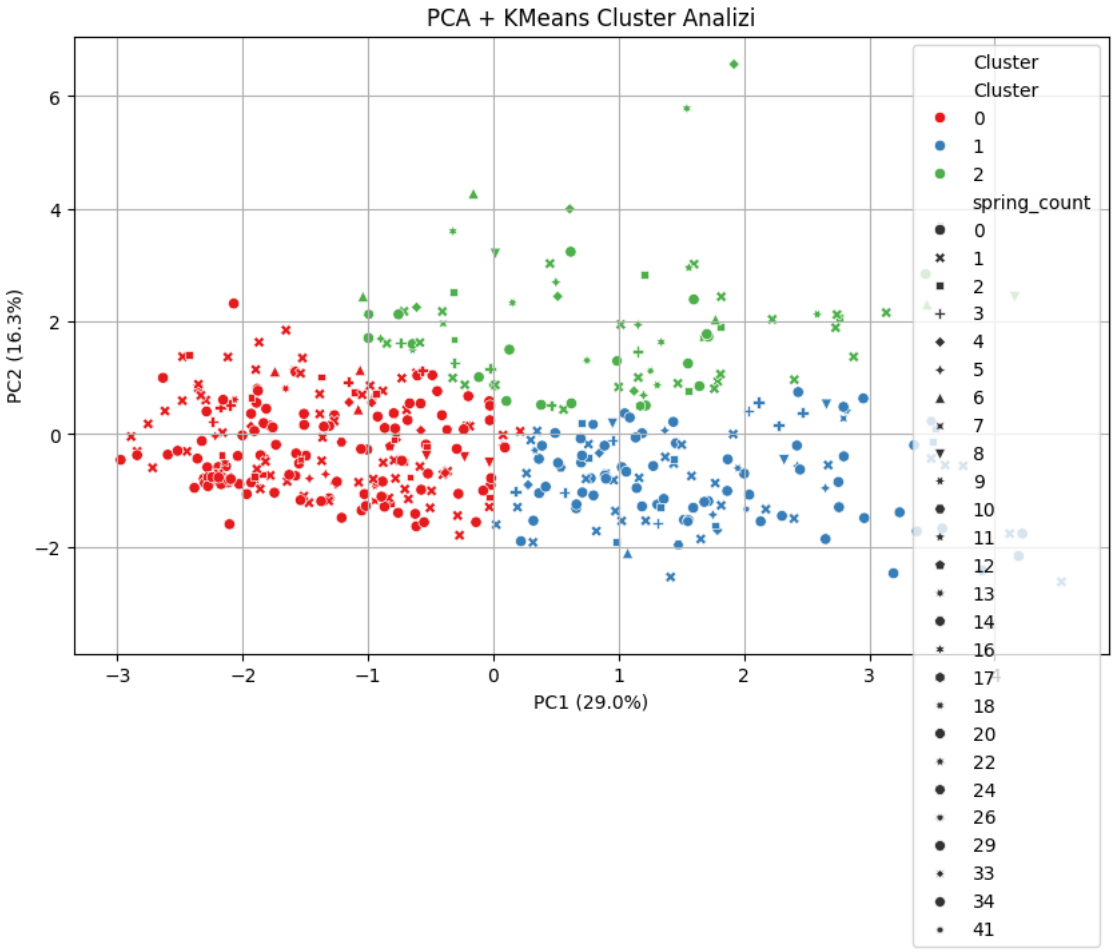
5.2.2 KMeans Clustering

PCA ile azaltılmış veride benzer havzaları aynı kümeye atayarak verideki yapıları ortaya çıkarmak.

Uygulanan Adımlar:

- PCA ile elde edilen iki boyutlu veriye KMeans (n_clusters=3) uygulandı.
- Küme sayısı (n_clusters=3) görselleştirmeye ve domain bilgisine dayanarak seçildi.
- Her havza, benzer özelliklerine göre bir kümeye atandı.

PCA görselleştirmesi ile desteklendi ve kümelerin PC1-PC2 düzleminde nasıl ayrıştığı gösterildi.



5.2.3 Decision Tree Classifier

KMeans ile belirlenen kümeleri sınıflandırmak ve hangi özelliklerin kümeleri ayırmada en önemli rolü oynadığını görmek için kullandığımız model.

Model Yapısı:

- Decision Tree Classifier (max_depth=5, min_samples_split=5) kullanıldı.
- Hedef değişken olarak KMeans ile elde edilen Cluster etiketleri kullanıldı.
- Bağımsız değişkenler PCA uygulanmadan önceki orijinal özellikler (spring_count ve springs within watershed hariç) olarak belirlendi.

Model Eğitim ve Testi

- PCA ve KMeans adımlarından sonra Decision Tree Classifier için veriler train-test split ile %70 eğitim ve %30 test olarak ayrıldı.

- SimpleImputer ile eksik veriler doldurulduu ve Decision Tree Classifier modeli bu verilerle eğitildi.
- Model performansı Accuracy, Precision, Recall, F1-Score ve Confusion Matrix ile değerlendirildi.

```
Decision Tree Classifier Performance:
Accuracy: 0.800

Classification Report:
              precision    recall  f1-score   support

           0       0.85      0.89      0.87        65
           1       0.73      0.75      0.74        40
           2       0.76      0.64      0.70        25

    accuracy                           0.80       130
   macro avg       0.78      0.76      0.77       130
weighted avg       0.80      0.80      0.80       130


Confusion Matrix:
[[58  4  3]
 [ 8 30  2]
 [ 2  7 16]]
```

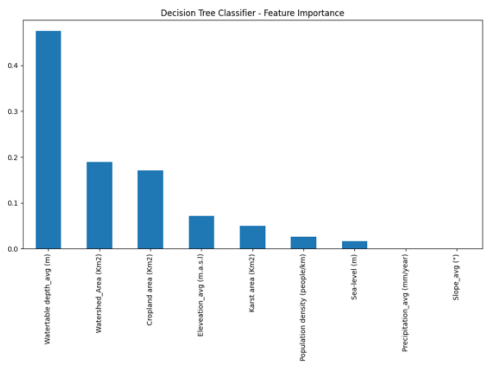
Decision Tree Classifier modeli test setinde **%80** doğruluk elde etti.

Bu sonuç, verideki kümelenmenin anlamlı olduğunu ve Decision Tree Classifier'ın kümeleri ayırt etmede başarılı olduğunu gösterdi.

Model feature importance hesaplayarak hangi değişkenlerin kümeleri ayırmada en etkili olduğunu belirledi.

- Watertable Depth Average
- Watershed Area
- Cropland Area

  değişkenlerinin en önemli özellikler olduğu görüldü.



6. Karşılaştırma ve Literatürle Bağlantı

Yöntem	Amaç	Başarı Düzeyi	Yorum
DecisionTree & RandomForest Regressor	spring_count tahmini	R2 ≈ 0.16–0.22	Direkt tahmin zayıf kaldı. Karmaşık etkileşimleri modellemek zor.
PCA + KMeans + DecisionTree Classifier	Segmentasyon ve sınıflandırma	Accuracy ≈ 0.80	Daha anlamlı segmentasyon sağlandı. Özelliklerin etkisi netleşti.

Bouimouass et al. (2025) çalışmasında sadece envanter verisi sağlanmıştı. Bu projede ise PCA + KMeans + Decision Tree kullanılarak segmentasyon ve feature importance analizi yapılarak yeni katkı sağlanmıştır.

7. Projenin Sonucu ve Genel Değerlendirme

Bu proje kapsamında Coastal Springs Dataset kullanılarak iki farklı yöntem ile spring_count değişkeninin analizi yapılmıştır:

Birinci Yöntem: Regresyon Modelleri (Decision Tree ve Random Forest Regressor)

- Spring_count değişkeni log1p dönüşümü ile normalize edilerek Decision Tree, Random Forest, Gradient Boosting ve Extra Trees Regressor modelleriyle tahmin edilmiştir.
- En iyi sonuç Random Forest Regressor modeli ile elde edilmiş olup; MAE: 1.539, MSE: 4.855 ve R^2: 0.218 olarak hesaplanmıştır.
- Bu sonuçlar, spring_count'un doğrudan tahmininde bazı hidrolojik ve coğrafi değişkenlerin tek başına yeterli olmadığını; modelin açıklama gücünün düşük kaldığını göstermektedir.

İkinci Yöntem: PCA + KMeans + Decision Tree Classifier ile Segmentasyon

- Spring_count'un doğrudan regresyonla tahmini yerine PCA ile boyut indirimi yapılmış ve veriler KMeans ile 3 kümeye ayrılarak benzer havzalar belirlenmiştir.

- Decision Tree Classifier kullanılarak kümelerin hangi değişkenler ile en iyi ayrıldığı analiz edilmiştir.
- Bu yöntem ile %80 accuracy değerine ulaşılmış, Precision, Recall ve F1-Score gibi metrikler de oldukça dengeli değerler göstermiştir.
- Özellikle Watershed_Area, Watertable Depth Average ve Cropland Area gibi değişkenler sınıflandırmada öne çıkan en etkili değişkenler olarak belirlenmiştir.
- PCA-KMeans yöntemi ile spring_count'un tahmin edilmesinden ziyade verilerin segmentasyonu başarıyla sağlanarak yönetilebilir gruplar ortaya konmuştur.

Genel Değerlendirme:

- Regresyon modelleri (özellikle Random Forest) ile spring_count'un tahmininde belirli bir başarı sağlanmış olsa da, verinin doğası gereği düşük R^2 değeri modelin yeterince açıklayıcı olmadığını göstermiştir.
- PCA + KMeans + Decision Tree Classifier yöntemiyle segmentasyon yaklaşımı, spring_count'un farklı hidrografik ve çevresel özelliklerle nasıl gruplandığını anlamada çok daha güçlü bir araç sunmuştur.
- Segmentasyon sayesinde su kaynaklarının yönetimi ve risk analizi gibi alanlarda bölgesel stratejiler geliştirilebilir; bu da projenin temel çıktılarından biridir.

Bu proje, Coastal Springs Dataset için ilk defa Decision Tree tabanlı ve PCA tabanlı analizlerin bir arada yapıldığı bir çalışma olup, hidrojeolojik faktörlerin spring_count üzerindeki etkilerini anlamak için hem sayısal hem görsel olarak zengin sonuçlar sağlamıştır. Bu çalışma, hem veri madenciliği alanında hem de su kaynakları yönetiminde önemli katkılar sunmaktadır.

GitHub: https://github.com/ccigdemavci/-data-mining

Youtube: https://www.youtube.com/watch?v=W8CA4k3qUnQ

Makale: https://www.sciencedirect.com/science/article/pii/S2352340925000514#refdata001

Dataset: https://data.mendeley.com/datasets/z33d3c5d8n/1