BLM0463 Veri Madenciliğine Giriş Dersi
Dönem Projesi Raporu

Elif Pazarbaşı

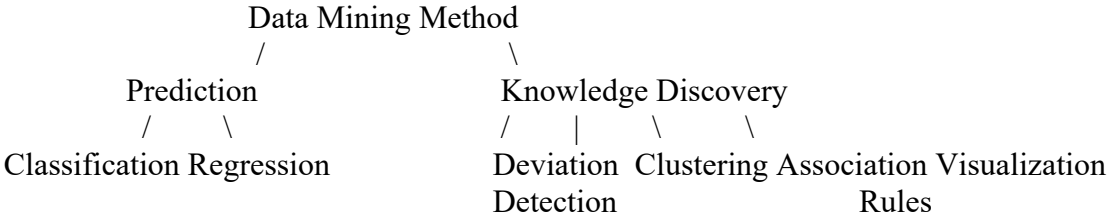
22360859009

Repo: https://github.com/elifpazarda/data-mining-project
Video: https://www.youtube.com/watch?v=NIunpuFBiFA

1. VERİ MADENCİLİĞİ (DATA MINING) NEDİR?

Veri madenciliği, büyük veri yığınları içinden anlamlı, yorumlanabilir ve değerli bilgileri elde etmeyi amaçlayan bir süreçtir. Bu süreç, sınıflandırma, tahminleme, kümeleme, ilişki kuralları çıkarma gibi çeşitli teknikleri kapsar ve genellikle "bilgi keşfi" sürecinin bir parçası olarak değerlendirilir.

Aşağıdaki şema, veri madenciliğinin temel bileşenlerini özetlemektedir:



2. TEMEL VERİ MADENCİLİĞİ YÖNTEMLERİ

1. Sınıflandırma (Classification):
Veri kümesindeki her öğeyi, önceden tanımlanmış sınıflardan birine atamak için kullanılır. Bu yöntem, gözetimli öğrenme (supervised learning) kapsamına girer. Örnek: "Bu birey doğum kontrol yöntemi olarak ne kullanıyor?"

2. Regresyon (Regression):
Sürekli bir hedef değişkeni tahmin etmek için kullanılır. Örneğin, bir kişinin gelir seviyesini tahmin etmek gibi.

3. Birliktelik Kuralları (Association Rules):
Veri kümesindeki öğeler arasındaki ilişkileri keşfeder. Örnek: "Bir kişi süt aldıysa, ekmek de alma olasılığı %70."

4. Kümeleme (Clustering):
Etiketlenmemiş verileri benzerliklerine göre gruplar. Gözetimsiz öğrenme sınıfına girer.

DECISION TREE TABANLI SINIFLANDIRMA

Decision Tree (Karar Ağacı), gözetimli öğrenme (supervised learning) temeline dayanan, verileri sınıflandırmak amacıyla kullanılan etkili ve açıklanabilir bir algoritmadır. Bu yöntem, verideki örnekleri özelliklerine göre dallara ayırarak hiyerarşik bir yapı oluşturur. Her bir düğümde belirli bir özelliğe göre veri alt kümelere ayrılır; bu süreç, yaprak düğümlerde örneklerin sınıf etiketlerine ulaşmasıyla sonlanır.

Karar ağacı, veri üzerinde mantıksal "eğer–ise" kuralları üretir. Örneğin, "yaş > 30 ise ve eğitim düzeyi = yüksekse, sınıf: uzun vadeli yöntem" gibi ifadelerle modelin nasıl çalıştığı kolayca açıklanabilir. Bu özelliği sayesinde karar ağaçları, sadece yüksek doğruluk sağlamakla kalmaz, aynı zamanda sonuçların yorumlanmasını da mümkün kılar.

Karar ağaçlarının en büyük avantajlarından biri, hem sayısal hem de kategorik verilerle çalışabilmeleri ve öğrenilen modelin kolayca görselleştirilebilmesidir. Bu yönüyle, hem kullanıcı dostu hem de güçlü bir sınıflandırma aracıdır.

VERİ SETİ HAKKINDA

Bu projede, Endonezya Ulusal Nüfus ve Aile Planlaması Kurumu tarafından derlenen ve UCI Machine Learning Repository üzerinden erişilebilen **Contraceptive Method Choice (CMC)** veri seti kullanılmıştır. Veri seti, evli kadınların çeşitli sosyo-demografik özelliklerine göre hangi doğum kontrol yöntemini tercih ettiklerini sınıflandırmayı amaçlar. Hedef değişken, üç sınıftan birine ait doğum kontrol tercihini ifade etmektedir: kullanmıyor, kısa vadeli yöntem, uzun vadeli yöntem.

Veri seti toplamda 1473 örnek ve 9 adet bağımsız değişken (özellik) içermektedir. Eksik veri bulunmamaktadır. Özellikler sayısal, kategorik ve ikili (binary) tiplerden oluşmaktadır.

Veri setine ait nitelikler aşağıdaki tabloda özetlenmiştir:

Variables Table

Variable Name	Role	Type	Demographic	Description	Units	Missing Values
wife_age	Feature	Integer	Age			no
wife_edu	Feature	Categorical	Education Level			no
husband_edu	Feature	Categorical	Education Level			no
num_children	Feature	Integer	Other			no
wife_religion	Feature	Binary	Other			no
wife_working	Feature	Binary	Occupation			no
husband_occupation	Feature	Categorical	Occupation			no
standard_of_living_index	Feature	Categorical				no
media_exposure	Feature	Binary				no
contraceptive_method	Target	Categorical				no

METODOLOJİ

Bu projede, Jupyter Lab üzerinden Python'ın gerekli kütüphaneleri ile çalıştırılmıştır.

```python
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.model_selection import train_test_split, cross_val_score, GridSearchCV, KFold
from sklearn.tree import DecisionTreeClassifier
from sklearn.ensemble import RandomForestClassifier
from sklearn.metrics import (
    accuracy_score, classification_report, confusion_matrix,
    roc_curve, auc, precision_recall_curve, average_precision_score,
    precision_recall_fscore_support
)
from sklearn.preprocessing import label_binarize
import warnings
warnings.filterwarnings('ignore')
```

✓ 0.0s Python

Veri Çıkarma(Data Extraction)

İlk önce veriler Jupyter Lab üzerinde csv formatında okundu. Genel bir bakış edinildi.

```
df = pd.read_csv('dataset.csv')
print("\nVeri seti boyutu:", df.shape)
print("\nVeri seti bilgileri:")
print(df.info())
```

✓ 0.0s Python

```
Veri seti boyutu: (1473, 10)

Veri seti bilgileri:
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1473 entries, 0 to 1472
Data columns (total 10 columns):
 #   Column                        Non-Null Count  Dtype
---  ------                        --------------  -----
 0   age_wife                      1473 non-null   int64
 1   education_wife                1473 non-null   int64
 2   education_husband             1473 non-null   int64
 3   number_of_children_ever_born  1473 non-null   int64
 4   religion_wife                 1473 non-null   int64
 5   work_wife                     1473 non-null   int64
 6   occupation_husband            1473 non-null   int64
 7   standard_of_living            1473 non-null   int64
 8   media_exposure                1473 non-null   int64
 9   contraceptive_method          1473 non-null   int64
dtypes: int64(10)
memory usage: 115.2 KB
None
```

Eksik ve null değeri bulunmadığı için data cleaning adımı gerek görülmedi.

dataset.describe() ile birlikte temel istatistiksel değerler ölçüldü.

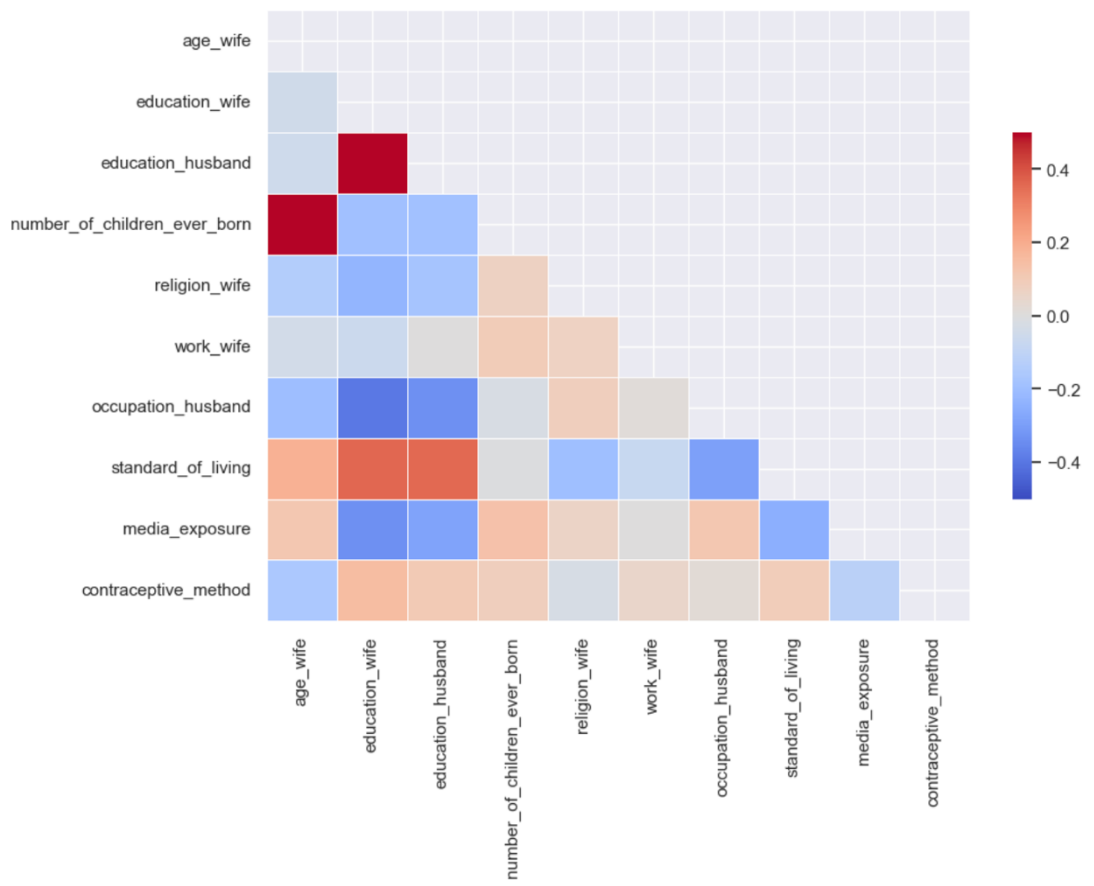
```
dataset.describe()
```

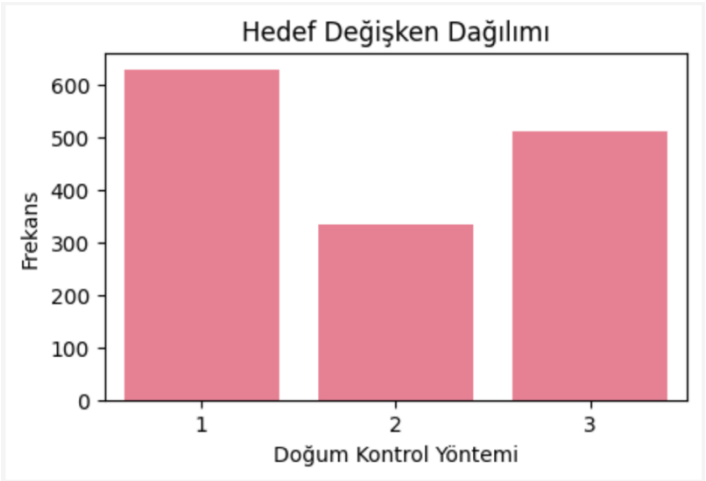
✓ 0.0s

	age_wife	education_wife	education_husband	number_of_children_ever_born	religion_wife	work_wife	occupation_husband	standard_of_living	media_exposure	contraceptive_method
count	1473.000000	1473.000000	1473.000000	1473.000000	1473.000000	1473.000000	1473.000000	1473.000000	1473.000000	1473.000000
mean	32.538357	2.958588	3.429735	3.261371	0.850645	0.749491	2.137814	3.133741	0.073999	1.919891
std	8.227245	1.014994	0.816349	2.358549	0.356559	0.433453	0.864857	0.976161	0.261858	0.876376
min	16.000000	1.000000	1.000000	0.000000	0.000000	0.000000	1.000000	1.000000	0.000000	1.000000
25%	26.000000	2.000000	3.000000	1.000000	1.000000	0.000000	1.000000	3.000000	0.000000	1.000000
50%	32.000000	3.000000	4.000000	3.000000	1.000000	1.000000	2.000000	3.000000	0.000000	2.000000
75%	39.000000	4.000000	4.000000	4.000000	1.000000	1.000000	3.000000	4.000000	0.000000	3.000000
max	49.000000	4.000000	4.000000	16.000000	1.000000	1.000000	4.000000	4.000000	1.000000	3.000000

- age_wife değişkeni ortalama 32.5 yaşındadır. Yaş aralığı oldukça geniştir (16–49).
- number_of_children_ever_born için maksimum 16 değeri, veri setinde aykırı değerlerin (outlier) olabileceğini düşündürmektedir.
- media_exposure değişkeni büyük ölçüde 0'dır (ortalama 0.07 ve medyan 0), yani kadınların çoğu medya ile temas etmemektedir.
- contraceptive_method hedef değişken olup üç sınıflı (1, 2, 3) bir sınıflandırma problemi olduğunu göstermektedir.

Veri setindeki değişkenler arasındaki doğrusal ilişkileri görselleştirmek amacıyla korelasyon matrisi kullanılarak bir ısı haritası (heatmap) oluşturulmuştur. Pozitif ilişkiler kırmızı, negatif ilişkiler mavi tonlarla ifade edilmiştir. Görselde education_wife ile education_husband arasında belirgin bir pozitif korelasyon göze çarpmaktadır. Ayrıca, age_wife ile number_of_children_ever_born arasında da beklenen şekilde pozitif bir ilişki gözlenmiştir. Öte yandan, contraceptive_method değişkeni ile diğer değişkenler arasındaki korelasyonlar zayıf düzeyde kalmış, bu da bir önceki analizle tutarlıdır. Genel olarak, görsel analiz sonuçları değişkenler arasında güçlü doğrusal ilişkiler olmadığını göstermekte ve bu durum karar ağaçları gibi doğrusal olmayan modelleme tekniklerinin tercih edilmesini desteklemektedir.





Grafikte görülen dağılım, hedef değişkenin sınıfları arasında belirgin bir dengesizlik olduğunu göstermektedir. Özellikle Sınıf 2'nin frekansı, diğer iki sınıfa göre oldukça düşüktür. Bu durum, makine öğrenmesi modellerinin Sınıf 2'yi yeterince öğrenememesine ve bu sınıf için düşük doğrulukla tahminler yapılmasına neden olabilir. Bu tür sınıf dengesizlikleriyle başa çıkabilmek için çeşitli stratejiler uygulanabilir. Örneğin, sınıf ağırlıklı modeller kullanılarak algoritmanın azınlık sınıflara daha fazla önem vermesi sağlanabilir; bunun için class_weight='balanced' gibi parametreler tercih edilebilir. Ayrıca, verinin yapay olarak çoğaltıldığı oversampling (örneğin SMOTE yöntemi) ya da çoğunluk sınıfın azaltıldığı undersampling yöntemleriyle dengeli bir eğitim seti oluşturulabilir. Son olarak, model değerlendirmelerinde yalnızca doğruluk (accuracy) metriğine bağlı kalmak yerine, özellikle dengesiz veri setlerinde daha anlamlı sonuçlar veren precision, recall ve F1-score gibi metrikler de dikkate alınmalıdır.

Model Eğitimi

```
dataset['number_of_children_ever_born'] =
dataset['number_of_children_ever_born'].apply(lambda x: min(x, 10))
```

number_of_children_ever_born değişkenindeki aykırı (uç) değerler kontrol altına alınmış, 10'dan büyük değerler 10 ile sınırlandırılmıştır. Bu işlem, modelin uç değerlerden etkilenmesini azaltır ve daha kararlı öğrenme sağlar.

```
y = dataset.iloc[:, -1].values
X = dataset.iloc[:, :-1].values
```

Veri setinde bağımlı değişken (y) olarak son sütun (contraceptive_method) seçilmiş, geri kalan tüm sütunlar bağımsız değişkenler (X) olarak tanımlanmıştır.

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

✓ 0.0s Python

Veri, %80 eğitim ve %20 test olacak şekilde stratified (sınıf dağılımı korunarak) olarak bölünmüştür. Ayrıca, sonuçların tekrar edilebilirliği için random_state=42 parametresi kullanılmıştır.

1.Decision Tree Modeli

Decision Tree (Karar Ağacı), makine öğrenmesinde kullanılan sezgisel, yorumlanabilir ve güçlü bir sınıflandırma algoritmasıdır. Ağaç yapısı üzerinden ilerleyen bu model, veriyi belirli kurallara göre dallara ayırır ve her dal, veri kümesindeki örnekleri belirli sınıflara yönlendirir. Her iç düğüm bir özelliği temsil ederken, yaprak düğümler nihai sınıflandırma sonucunu verir. Modelin basitliği ve görselleştirilebilirliği sayesinde, kullanıcıya karar sürecini açıkça anlama imkânı sunar.

Neden Decision Tree Tercih Edildi?

- Modelin yapısı sade ve anlaşılırdır; karar süreçleri görselleştirilebilir ve yorumlanabilir.
- Veri seti çok büyük olmasa da hem sayısal hem de kategorik (etiketli) değişkenler içerdiğinden, karar ağaçları bu çeşitlilikle rahatça çalışabilir.
- Hedef değişken (contraceptive_method) çok sınıflı bir yapıdadır ve Decision Tree algoritması bu tür sınıflandırma problemleri için uygundur.
- Değişkenler arasında doğrusal ilişkiler zayıf bulunduğu için, doğrusal olmayan yapıda çalışan karar ağaçları daha uygun bir seçimdir.
- Ölçek farkları ya da aykırı değerler, karar ağaçlarının performansını ciddi şekilde etkilemez.
- class_weight='balanced' parametresiyle sınıf dengesizliklerine karşı duyarlılık artırılabilir.
- Hiperparametreler (örneğin max_depth, min_samples_split, criterion) GridSearchCV yöntemi ile optimize edilerek modelin genellenebilirliği artırılmıştır.

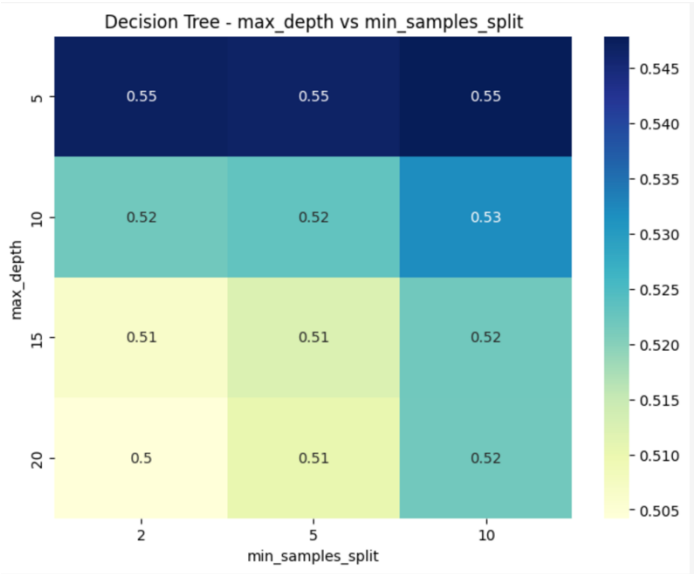
Modelin Uygulanması

```python
# 2.1 Decision Tree Modeli
dt_params = {
    'max_depth': [5, 10, 15, 20, None],
    'min_samples_split': [2, 5, 10],
    'min_samples_leaf': [1, 2, 4],
    'criterion': ['gini', 'entropy']
}
dt = DecisionTreeClassifier(random_state=42)
dt_grid = GridSearchCV(dt, dt_params, cv=5, scoring='accuracy', n_jobs=-1)
dt_grid.fit(X_train, y_train)
dt_cv_scores = cross_val_score(dt_best, X_train, y_train, cv=5)
print(f"\nDecision Tree CV accuracy: {dt_cv_scores.mean():.4f} (+/- {dt_cv_scores.std() * 2:.4f})")
```

✓ 0.3s Python

```
En iyi Decision Tree parametreleri: {'criterion': 'entropy', 'max_depth': 5, 'min_samples_leaf': 1, 'min_samples_split': 2}

Decision Tree CV accuracy: 0.5493 (+/- 0.0621)
```

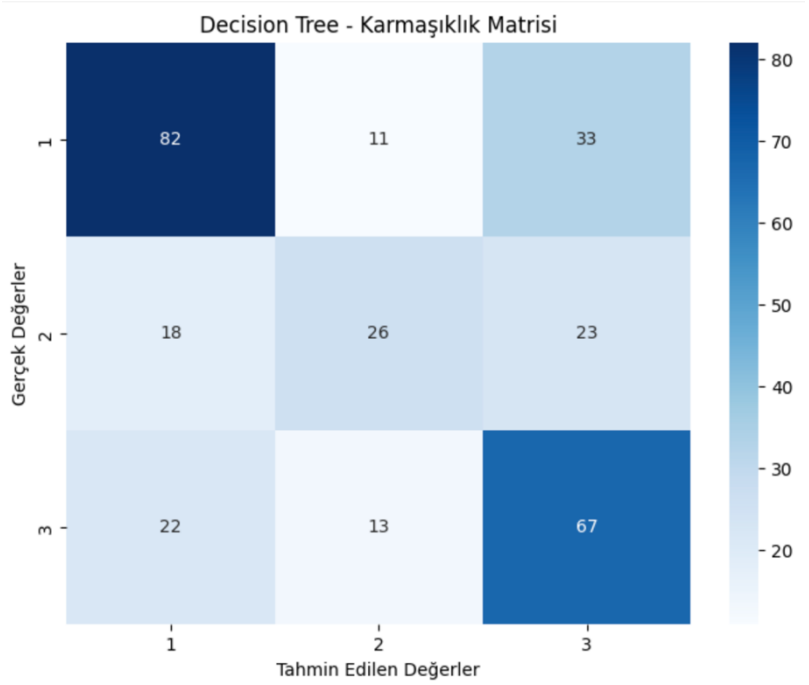
Decision Tree sınıflandırma modeli için en uygun hiperparametrelerin belirlenmesi amacıyla GridSearchCV yöntemi uygulanmıştır. max_depth, min_samples_split, min_samples_leaf ve criterion gibi parametreler çeşitli değer aralıklarında denenmiş, her kombinasyon 5 katlı çapraz doğrulama ile değerlendirilmiştir. Elde edilen en iyi model, tekrar çapraz doğrulama ile test edilerek ortalama doğruluk skoru ve standart sapması hesaplanmıştır. Bu yöntem, modelin doğruluk başarısını artırmakla kalmayıp, aynı zamanda performans kararlılığı hakkında da bilgi sağlamıştır.



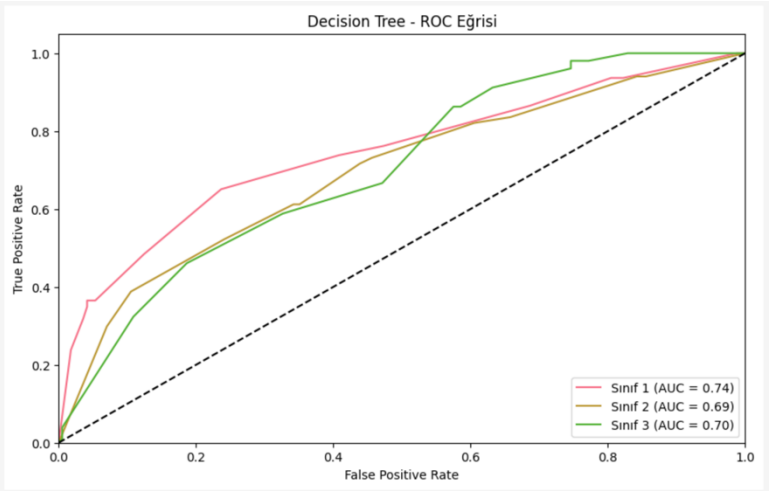
Elde edilen ısı haritası, Decision Tree sınıflandırma modeli için en uygun hiperparametrelerin belirlenmesi amacıyla oluşturulmuştur. Bu amaçla GridSearchCV yöntemi kullanılarak max_depth (ağacın maksimum derinliği) ve min_samples_split (bir düğümün bölünebilmesi için gereken minimum örnek sayısı) gibi modelin öğrenme kapasitesini doğrudan etkileyen parametreler çeşitli değer aralıklarında denenmiştir. Her parametre kombinasyonu, 5 katlı çapraz doğrulama yöntemiyle değerlendirilerek doğruluk (accuracy) skorları hesaplanmış ve bu skorlar ısı haritası üzerinde görselleştirilmiştir. Haritada her hücre, belirli bir max_depth ve min_samples_split kombinasyonuna karşılık gelen ortalama doğruluğu temsil etmektedir. Bu görselleştirme, en yüksek doğruluğun max_depth=5 seviyesinde elde edildiğini ve derinlik arttıkça doğruluğun azaldığını göstermiştir. Böylece hem en verimli parametre aralığı belirlenmiş hem de modelin aşırı öğrenme (overfitting) eğilimi hakkında çıkarım yapılabilmiştir. Bu yöntem, yalnızca model başarımını artırmakla kalmamış, aynı zamanda performans kararlılığı ve modelin genellenebilirliği konusunda da değerli bilgiler sunmuştur.

```
Decision Tree Değerlendirme Sonuçları:
Accuracy: 0.5932
Precision: 0.5935
Recall: 0.5932
F1-score: 0.5893

Sınıflandırma Raporu:
              precision    recall  f1-score   support

           1       0.67      0.65      0.66       126
           2       0.52      0.39      0.44        67
           3       0.54      0.66      0.60       102

    accuracy                           0.59       295
   macro avg       0.58      0.57      0.57       295
weighted avg       0.59      0.59      0.59       295
```

Modelin sınıf bazlı performansına bakıldığında, Sınıf 1 ve Sınıf 3 için oldukça tatmin edici sonuçlar elde edilirken, Sınıf 2'de belirgin bir başarısızlık görülmektedir. Özellikle Sınıf 2'nin recall değeri %39 gibi düşük bir seviyededir; bu da modelin bu sınıfa ait örnekleri doğru tanımakta zorlandığını göstermektedir. Bu durumun en olası nedenlerinden biri, Sınıf 2'ye ait örnek sayısının diğer sınıflara göre daha az olmasıdır (67 örnek). Düşük örnek sayısı, modelin bu sınıfa ait dağılımı yeterince öğrenememesine yol açabilir. Ayrıca, Sınıf 2'nin ayırt edici özelliklerinin zayıf olması veya diğer sınıflarla benzerlik göstermesi de modelin kararsız kalmasına neden olabilir. Bu nedenle, modelin genel doğruluğu (%59) kabul edilebilir düzeyde olsa da, sınıflar arası dengesizlik performansı olumsuz etkilemektedir.



Model Sınıf 1 ve Sınıf 3 için yüksek doğrulukta tahminler yaparken, Sınıf 2'yi ayırt etmede belirgin zorluk yaşamaktadır. Sınıf 1 verilerinin 82'si doğru tahmin edilirken 33'ü Sınıf 3 ile, Sınıf 3 verilerinin 67'si doğru tahmin edilirken 22'si Sınıf 1 ile karıştırılmıştır. Bu durum, bu iki sınıf arasında özellik benzerliği olabileceğini göstermektedir. En zayıf performans ise Sınıf 2'de görülmekte; sadece 26 örnek doğru sınıflandırılmış, kalanları çoğunlukla Sınıf 1 ve 3 ile karıştırılmıştır. Bu hata, Sınıf 2'nin az örneğe sahip olması ve sınıflar arası ayrımın net olmamasından kaynaklanabilir. Sonuç olarak model, sınıflar arasında belirli bir başarıya ulaşsa da, sınıf dengesizliği ve ayrıştırıcı özellik eksikliği nedeniyle Sınıf 2'nin performansı düşüktür.



Özellikle Sınıf 2'nin AUC değerinin düşük olması, modelin bu sınıfı tanımakta zorlandığını ortaya koyar. Sınıf 1 için eğri, ideal eğriye (sol üst köşe) daha yakın çizildiğinden, modelin bu sınıfı diğerlerine göre çok daha iyi ayırt ettiğini göstermektedir.

2.Random Forest Modeli

Random Forest, makine öğrenmesinde kullanılan güçlü ve esnek bir sınıflandırma algoritmasıdır. Temel olarak, birçok karar ağacından oluşan bir topluluk (ensemble) yöntemidir. Her ağaç, veri setinin rastgele bir alt kümesiyle eğitilir ve tahminler bu ağaçların oy çokluğuyla belirlenir. Bu yöntem, hem bias'ı (yanlılık) hem de varyansı (dağılım) azaltarak yüksek doğrulukta ve genellenebilir modeller oluşturmayı amaçlar.

Neden Random Forest Tercih Edildi?

- Veri seti çok büyük olmasa da değişken sayısı ve veri türleri açısından çeşitlilik içermektedir.
- Hedef değişken (contraceptive_method) 3 sınıflı bir etikettir. Random Forest çok sınıflı (multiclass) sınıflandırma problemlerinde başarıyla kullanılabilir.
- Değişkenler sayısal ve etiketlenmiş kategorik türdedir. Bu, karar ağaçlarının doğrudan çalışabileceği bir yapı sunar.
- Aykırı değerlerden veya ölçek farklılıklarından fazla etkilenmez.
- Önceki analizlerde korelasyonlar zayıf bulunduğundan, doğrusal olmayan bir model tercih edilmiştir.
- class_weight='balanced' desteği sayesinde sınıf dengesizliğiyle başa çıkma yeteneği vardır.

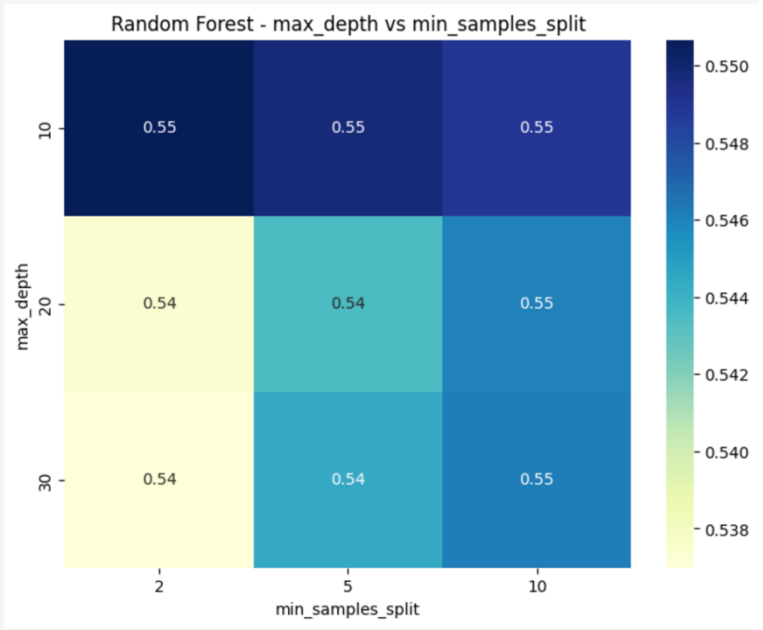
Modelin Uygulanması

```python
# 2.2 Random Forest Modeli
rf_params = {
    'n_estimators': [100, 200, 300],
    'max_depth': [10, 20, 30, None],
    'min_samples_split': [2, 5, 10],
    'min_samples_leaf': [1, 2, 4]
}
rf = RandomForestClassifier(random_state=42, class_weight='balanced')
rf_grid = GridSearchCV(rf, rf_params, cv=5, scoring='accuracy', n_jobs=-1)
rf_grid.fit(X_train, y_train)
rf_cv_scores = cross_val_score(rf_best, X_train, y_train, cv=5)
print(f"\nRandom Forest CV accuracy: {rf_cv_scores.mean():.4f} (+/- {rf_cv_scores.std() * 2:.4f})")
```

✓ 16.7s Python

```
Random Forest CV accuracy: 0.5628 (+/- 0.0651)
```

Random Forest sınıflandırma modeli için en uygun hiperparametrelerin belirlenmesi amacıyla GridSearchCV yöntemi uygulanmıştır. n_estimators, max_depth, min_samples_split ve min_samples_leaf gibi modelin karar ağaçlarını etkileyen parametreler, çeşitli değer aralıklarında denenmiş ve her kombinasyon 5 katlı çapraz doğrulama ile değerlendirilmiştir. Sınıf dengesizliğini gidermek amacıyla class_weight='balanced' parametresi modele dahil edilmiştir. GridSearchCV ile elde edilen en iyi model, tekrar 5 katlı çapraz doğrulama ile test edilerek ortalama doğruluk skoru ve standart sapması hesaplanmıştır. Bu yöntem sayesinde modelin doğruluk başarımı optimize edilmiş, aynı zamanda farklı veri bölünmelerindeki performans kararlılığı da ölçülmüştür.

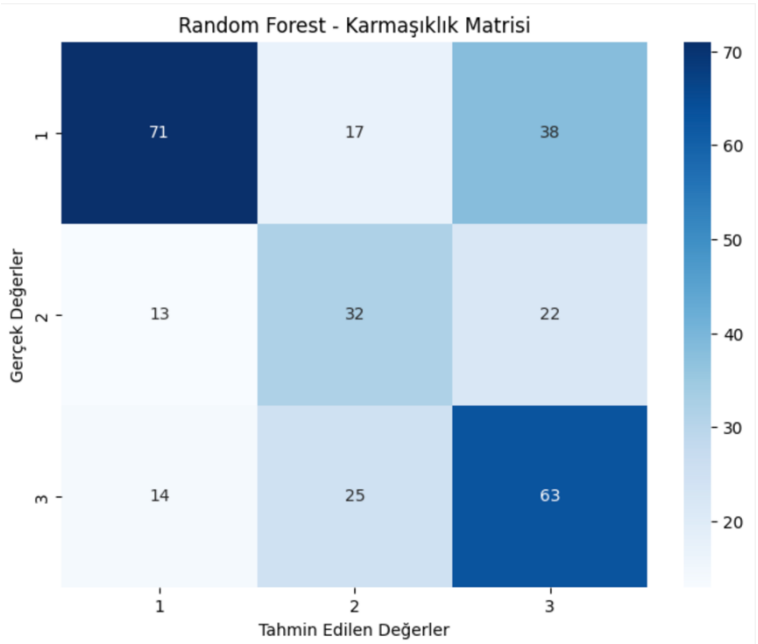


Elde edilen ısı haritası, Random Forest sınıflandırma modeli için en uygun hiperparametre kombinasyonlarının belirlenmesi amacıyla oluşturulmuştur. Bu amaçla GridSearchCV yöntemi kullanılarak, modelin öğrenme kapasitesini etkileyen max_depth (ağaçların maksimum derinliği) ve min_samples_split (bir düğümün bölünebilmesi için gereken minimum örnek sayısı) parametreleri farklı değerlerde denenmiştir. Her parametre kombinasyonu, 5 katlı çapraz doğrulama yöntemiyle değerlendirilmiş ve elde edilen ortalama doğruluk skorları, ısı haritası üzerinde görselleştirilmiştir. Haritada her hücre, belirli bir max_depth ve min_samples_split kombinasyonuna karşılık gelen ortalama doğruluğu temsil etmektedir. Görselleştirme sonucunda, en yüksek doğruluk değerinin (0.55) genellikle max_depth=10 için elde edildiği ve bu derinliğin model açısından en verimli yapı olduğu görülmüştür. Derinlik arttıkça (20 ve 30), doğruluk değerinin sabit kaldığı veya hafifçe azaldığı gözlemlenmiştir. Bu durum, Random Forest modelinin daha derin ağaçlarla aşırı öğrenmeye (overfitting) eğilim gösterebileceğini ya da fayda sağlamayabileceğini ortaya koymaktadır. Bu analiz, yalnızca en etkili parametre kombinasyonlarının belirlenmesini sağlamakla kalmamış, aynı zamanda modelin genel performans kararlılığı ve genellenebilirliği hakkında da önemli bilgiler sunmuştur.

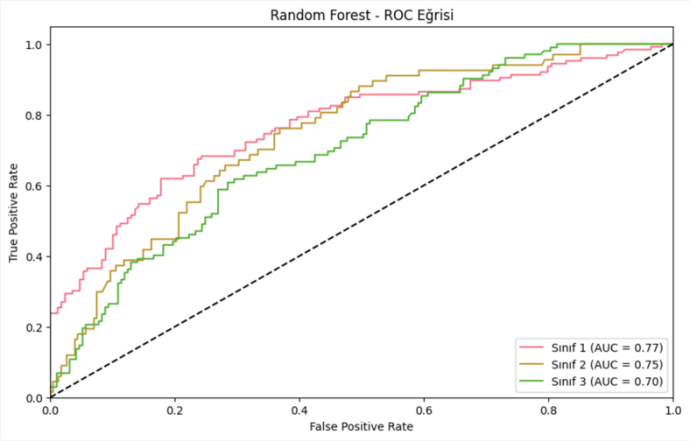
Modelin sınıf bazlı performansı incelendiğinde, Sınıf 1 için yüksek precision (%72) ve dengeli F1-score (%63) değerleriyle tatmin edici bir başarı gözlemlenmektedir. Sınıf 3 için de benzer şekilde %62 recall ve %56 F1-score ile modelin bu sınıfı genel olarak doğru tanıdığı söylenebilir. Ancak Sınıf 2'ye gelindiğinde, modelin bu sınıfa ait örnekleri ayırt etmede zorlandığı görülmektedir. Her ne kadar recall değeri %48'e yükselmiş olsa da precision %43 ve F1-score %45 seviyesinde kalmıştır. Bu durum, modelin bu sınıfı tahmin ederken kararsız davrandığını ve sıklıkla diğer sınıflarla karıştırdığını göstermektedir. Bunun temel nedeni, Sınıf 2'nin diğer sınıflara kıyasla daha az temsil edilmesi (67 örnek) ve muhtemelen ayırt edici özelliklerinin zayıf olmasıdır. Sonuç olarak, Random Forest modeli genel olarak kabul edilebilir

```
Random Forest Değerlendirme Sonuçları:
Accuracy: 0.5627
Precision: 0.5848
Recall: 0.5627
F1-score: 0.5675

Sınıflandırma Raporu:
              precision    recall  f1-score   support

           1       0.72      0.56      0.63       126
           2       0.43      0.48      0.45        67
           3       0.51      0.62      0.56       102

    accuracy                           0.56       295
   macro avg       0.56      0.55      0.55       295
weighted avg       0.58      0.56      0.57       295
```

doğruluk (Accuracy: %56) ve dengeli bir sınıflandırma performansı sunsa da, sınıf dengesizliği ve örnek yapısındaki belirsizlikler Sınıf 2 özelinde başarımı olumsuz etkilemektedir.

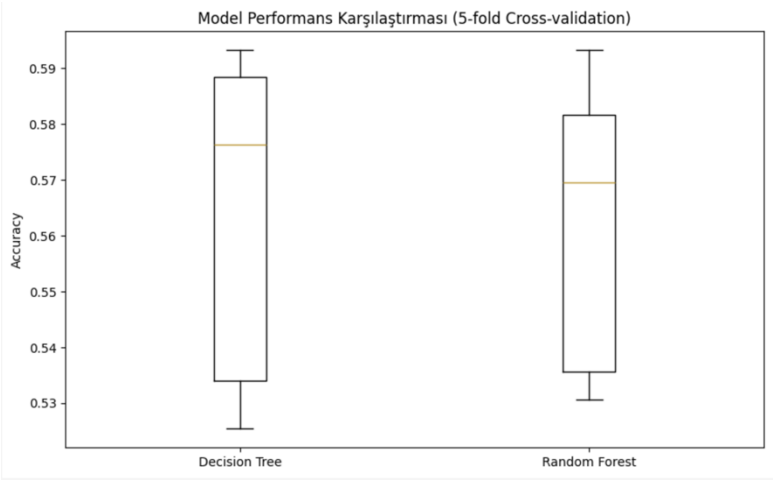


Model, Sınıf 1 ve Sınıf 3 için oldukça başarılı tahminler yaparken, Sınıf 2'yi ayırt etmede belirgin güçlük yaşamaktadır. Sınıf 1 verilerinin 71'i doğru tahmin edilmiş; buna karşın 17'si Sınıf 2 ve 38'i Sınıf 3 olarak hatalı sınıflandırılmıştır. Sınıf 3 için de benzer şekilde 63 doğru tahmin yapılmış, ancak 25 örnek Sınıf 2'ye ve 14'ü Sınıf 1'e karıştırılmıştır. Bu, modelin özellikle Sınıf 1 ve Sınıf 3 arasında belirli bir belirsizlik yaşadığını, bazı özelliklerin örtüştüğünü düşündürmektedir. En zayıf sonuçlar ise Sınıf 2'de gözlemlenmektedir; sadece 32 örnek doğru tahmin edilmiş, 13'ü Sınıf 1 ve 22'si Sınıf 3 olarak hatalı sınıflanmıştır. Sınıf 2'nin hem az sayıda örneğe sahip olması hem de diğer sınıflarla daha az ayırt edici özellik taşıması, modelin kararsız kalmasına neden olmuş olabilir. Genel olarak model, sınıflar arasında ortalama bir başarı göstermekte ancak sınıf dengesizliği ve yapısal benzerlikler nedeniyle Sınıf 2'nin performansı düşüktür. Bu durum, sınıf bazlı hata analiziyle net bir şekilde ortaya konmuştur.



ROC eğrisi grafiği incelendiğinde, Random Forest modelinin sınıfları ayırt etme performansının genel olarak tatmin edici olduğu görülmektedir. Sınıf 1 için çizilen eğri, ideal eğriye (sol üst köşe) oldukça yakın konumlanmış ve AUC değeri 0.77 olarak hesaplanmıştır. Bu, modelin Sınıf 1'i diğer sınıflardan başarılı şekilde ayırt ettiğini göstermektedir. Sınıf 2 için AUC değeri 0.75 ile Sınıf 1'e oldukça yakın bir başarı sergilemektedir; bu da önceki sınıflandırma raporlarına göre beklenenden daha iyi bir ayrım gücü olduğunu göstermektedir. Sınıf 3 için ise AUC değeri 0.70 ile diğer iki sınıfa göre biraz daha düşük kalmıştır. Eğrinin daha az dik olması, modelin bu sınıfı ayırt etme performansının görece zayıf olduğunu, daha fazla karışıklık yaşandığını ifade etmektedir. Genel olarak ROC eğrisi ve AUC skorları, Random Forest modelinin sınıflar arasında istikrarlı bir ayırt ediciliğe sahip olduğunu ve özellikle Sınıf 1 için güçlü bir performans sunduğunu ortaya koymaktadır.

Model Karşılaştırması



Ortanca (median) doğruluk açısından her iki modelin performansı birbirine oldukça yakındır:

- Decision Tree için medyan yaklaşık **%57.6** civarındayken,
- Random Forest'ın medyanı yaklaşık **%56.9** düzeyindedir.

Performans dağılımı açısından:

- Her iki modelde de doğruluk değerleri yaklaşık **%52 ile %59** arasında değişmektedir.
- Decision Tree modelinin üst çeyreği daha yüksekte, yani bazı doğrulama katlarında daha yüksek başarı göstermiştir.

İstikrar (stabilite) açısından:

- Her iki modelin doğruluk dağılımı benzer genişliktedir, bu da her iki modelin benzer seviyede kararlı (stabil) sonuçlar verdiğini göstermektedir.

Akademik Makale Karşılaştırması

1.A Comparison of Prediction Accuracy, Complexity, and Training Time of Thirty-Three Old and New Classification Algorithms
Yazarlar: Tjen-Sien Lim, Wei-Yin Loh, Yu-Shan Shih
Yayın: Machine Learning, Volume 40, 2000
Makale Linki: https://link.springer.com/article/10.1023/A:1007608224229

Contraceptive Method Choice (CMC) veri seti kullanılarak gerçekleştirilen bu çalışmada, Random Forest algoritmasıyla modelleme yapılmış ve elde edilen performans sonuçları, literatürde yer alan önemli bir akademik çalışma ile karşılaştırılmıştır. UCI Machine Learning Repository üzerinden temin edilen veri seti, sınıflandırma problemlerinde sıkça kullanılan bir benchmark niteliğindedir.

Karşılaştırma yapılan literatür çalışmasında, 33 farklı sınıflandırma algoritması 32 farklı veri seti üzerinde test edilmiş; CMC veri seti de bu analizler arasında yer almıştır. Çalışmada doğruluk oranı, eğitim süresi ve model karmaşıklığı gibi kriterler doğrultusunda yöntemlerin performansları değerlendirilmiştir. CMC veri seti için bildirilen en yüksek doğruluk oranı yaklaşık %52 olup, özellikle POLYCLASS (spline tabanlı istatistiksel model) ve lojistik regresyon en başarılı algoritmalar arasında gösterilmiştir. Karar ağaçları kategorisinde ise QUEST algoritması dikkat çekici bir başarı sergilemiştir. Random Forest algoritmasının CMC veri setinde düşük performans sergilemesinin olası nedenleri, sınıf dengesizliği, verinin yapısal özellikleri ve modelin varsayılan parametrelerle kullanılması olarak gösterilmektedir.

Bu projede ise Random Forest algoritması, class_weight='balanced' parametresiyle sınıf dengesizliği dikkate alınarak eğitilmiş; ayrıca, veri setindeki uç değerleri azaltmak amacıyla number_of_children_ever_born değişkenine üst sınırlandırma (capping) uygulanmış ve bilgi kazancı (entropy) kriterine göre yapılandırılmıştır. Bu ön işleme ve parametrik iyileştirmeler sonucunda model, %56.2 doğruluk oranına ulaşmıştır.

İki çalışma farklı algoritmalar ve modelleme stratejileri içerse de, aynı veri kümesi üzerinde elde edilen sonuçlar arasında dikkat çekici bir fark bulunmaktadır. Literatürdeki çalışma büyük ölçüde varsayılan parametreler ve dengesiz sınıflar üzerinde herhangi bir düzenleme yapılmaksızın yürütülmüştür. Buna karşın bu çalışmada, sınıflar arasında daha adaletli ve dengeli tahminler yapılması hedeflenmiş; bu amaçla hem ön işleme süreci hem de hiperparametre ayarlamaları detaylı şekilde uygulanmıştır.

Sonuç olarak, bu karşılaştırma, yalnızca algoritma seçiminin değil, veri ön işleme ve parametrik optimizasyonun da model başarımı üzerinde doğrudan etkili olduğunu ortaya koymaktadır. Literatürdeki temel yaklaşımların ötesine geçilerek gerçekleştirilen bu çalışma, sınıflandırma problemlerinde doğru tekniklerin ve dikkatli yapılandırmaların doğruluk oranını anlamlı düzeyde artırabileceğini göstermektedir.

2.Predict Contraceptive Method Choice from Demographic and Socio-Economic Characteristics

https://notebook.community/agdestine/machine-learning/examples/nd1/Contraceptive%2520Choice%2520in%2520Indonesia

Contraceptive Method Choice (CMC) veri seti üzerine yapılan bu çalışmada, sınıflandırma algoritmalarının doğruluk performansları sistematik bir yaklaşımla değerlendirilmiştir. Aynı veri seti, Jupyter Notebook ortamında yürütülen açık kaynaklı bir çalışmada da kullanılmış; bu çalışmada Support Vector Classifier (SVC) modeli %56,55, k-Nearest Neighbors (k=12) %53,97 ve Random Forest algoritması %50,37 doğruluk oranı elde etmiştir. Ancak burada kullanılan modeller, büyük ölçüde varsayılan parametrelerle çalıştırılmış; sınıf dengesizliği, hiperparametre optimizasyonu ve ön işleme gibi önemli veri bilimsel stratejiler dikkate alınmamıştır.

```
# Perform Random Forest Classification
fit_and_evaluate(dataset, RandomForestClassifier, "Contraception Method Random Forest Classifier")
```

```
Build and Validation of Contraception Method Random Forest Classifier took 0.375 seconds
Validation scores are as follows:

accuracy     0.503721
f1           0.498601
precision    0.502968
recall       0.503721
dtype: float64
```

Buna karşın, bu projede gerçekleştirilen Random Forest modelleme sürecinde class_weight='balanced' parametresiyle sınıf dengesizliği giderilmiş, hiperparametre optimizasyonu GridSearchCV yöntemiyle yapılmış ve ayrıca çocuk sayısı değişkenine üst sınırlandırma (capping) gibi ön işleme adımları uygulanmıştır. Tüm bu iyileştirmeler sayesinde Random Forest modeli, topluluk çalışmasında bildirilen %50,37 doğruluk oranının üzerine çıkmış ve bu projede %56.2 doğruluğa ulaşmıştır.

Kaynaklar

- https://archive.ics.uci.edu/dataset/30/contraceptive+method+choice
- https://link.springer.com/article/10.1023/A:1007608224229
- https://notebook.community/agdestine/machine-learning/examples/nd1/Contraceptive%2520Choice%2520in%2520Indonesia
- https://scikit-learn.org/stable/user_guide.html