# BURSA TEKNİK ÜNİVERSİTESİ

# PITTSBURGH KÖPRÜLERİ VERİ SETİYLE SINIFLANDIRMA YAPILMASI

**VERİ MADENCİLİĞİ DERSİ PROJESİ**

**Kerem Ersu**
**18360859059**

**Öğretim Üyesi: Erdem Yavuz**

**BAHAR DÖNEMİ 2022**

**İÇİNDEKİLER**

**Sayfa**

## ŞEKİL LİSTESİ

**Sayfa**

# ÖZET

Bu raporda Pittsburgh köprüleri veri setine eksik değer incelemesi, ayrık değer incelemesi, model oluşturulması, cross validation işlemi uygulanması gibi aşamalar uygulanacaktır. Son olarak da oluşturulan modelin başarısı çeşitli metrikler ışığında gözlemlenecektir.

# 1. GİRİŞ

Projede, UCI Machine Learning sitesinden alınan Pittsburgh Köprüleri (Pittsburgh Bridges) veri seti alınarak analizler yapıldı. Veriler, Pittsburghta bulunan köprülerin özelliklerini ele almaktadır. Bu veriler, köprünün lokasyonu, yapıldığı malzeme, yapılma amacı, köprünün büyüklüğü gibi özellikleri içermektedir.

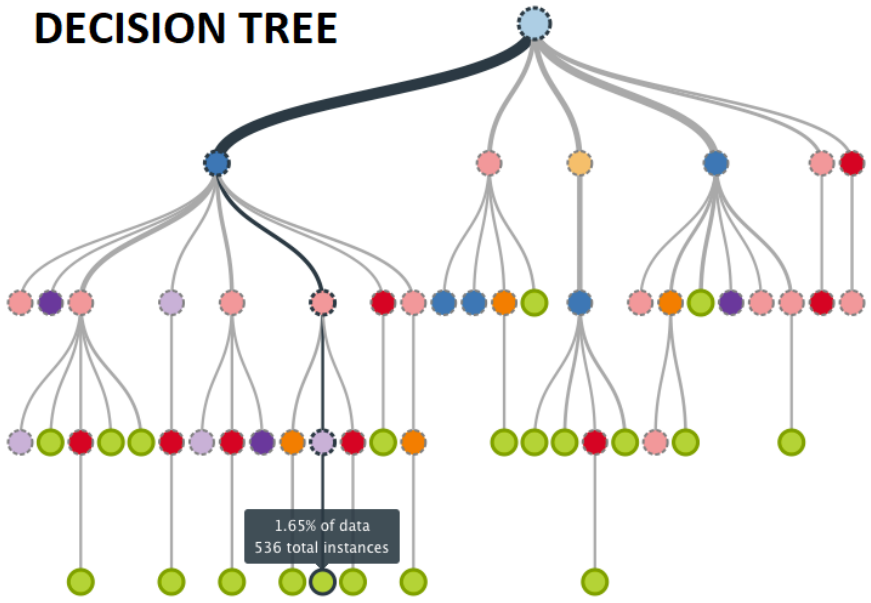
Bu veri seti kullanılarak analiz edilen verilerin açıklanması, görselleştirilmesi, eksik verilerin doldurulması daha sonrasında ise karar ağacı tekniği (decision tree method) kullanılarak veri sınıflandırma modellemesi yapılmıştır.

# 2. KULLANILAN YÖNTEMLER

## 2.1 Karar Ağacı Metodolojisi (Decision Tree Method)

Projede, modelleme yapılması için veri madenciliği sınıflandırma yöntemlerinden biri olan karar ağacı metodu (decision tree method) kullanılmıştır.

Karar ağaçları, önceden tanımlanmış bir hedef değişkene sahiptir. Yapıları itibariyle en tepeden en aşağı inen bir strateji sunmaktadırlar. Bir karar ağacı, çok sayıda kayıt içeren bir veri kümesini, bir dizi karar kuralları uygulayarak daha küçük kümlere bölmek için kullanılan bir yapıdır. Yani basit karar verme adımları uygulanarak, büyük miktarlardaki kayıtları, çok küçük kayıt yapılarına bölerek kullanılan bir yapıdır.




**Şekil 1.1: Karar Ağacı Yapısı**

Bu yöntemin avantajları; anlama ve yorumlama açısından kolay yapılardır, bu yapılar görselleştirilebilirdir. Veri hazırlığına az oranda ihtiyaç duyar, fakat bu modelkayıp değerleri desteklemez. Ağacın maliyeti logaritmiktir. Hem sayısal hem de kategorik veri işleyebilir. Çok çıktılı problemleri ele alabilir gibi avantajlara sahiptir.

Avantajlarının yanında bazı dezavantajlar da bulunur. Bunlardan bazıları ise aşırı karmaşık ağaçların üretilebilmesi ve ezbere öğrenme yani overfitting yaşanabilmesidir. Aşırı karmaşık ağaçların oluşması durumunda ağaç dallanması takip edilmesi çok zorlaşabilir. Aynı şekilde overfitting durumunda ise parametre kısıtlama ve budama gibi yöntemler kullanılabilir.

Karar ağaçları düğümleri, alt düğümlere bölmek için birden fazla algoritma kullanır. Bu düğümlerin oluşturulması, düğümlerin homojenliğini arttırır. Yani düğümün saflığı hedef değişkenlere göre artar.

Algoritma seçimi, hedef değişkenin tipine dayanır. Karar ağaçlarında sık kullanılan algoritmalar; kategorik değişkenler için entropi, gini, sınıflandırma hatası; sürekli değişkenler için ise en küçük karalere ayırma yöntemi gibi yöntemlerdir.

## 2.2 Kullanılan Yazılım Teknolojileri

Projenin modellenmesinde Python programlama dili ve bu dilin çeşitli kütüphaneleri kullanılmıştır.

### 2.2.1 Python Programlama Dili

Python 1991 yılında Guido Van Rossum tarafından geliştirilen bir high-level programlama dilidir. Gerek kod yazarken tanıdığı özgürlük gerekse kolay syntaxı sayesinde, günümüzde en popüler programlama dilleri arasında yer alan Python gün geçtikçe daha da popülerleşmekte.
“Python nedir?” sorusuna kısa bir cevap verecek olursak; neredeyse her amaç için kullanılabilen Python, obje yönelimli, yorumlanabilir ve dinamik bir programlama dilidir.

### 2.2.2 Pandas Kütüphanesi

Pandas, veri işlemesi ve analizi için Python programlama dilinde yazılmış olan bir yazılım kütüphanesidir. Bu kütüphane temel olarak zaman etiketli serileri ve sayısal tabloları işlemek için bir veri yapısı oluşturur ve bu şekilde çeşitli işlemler bu veri yapısı üzerinde gerçekleştirilebilir olur. Yazılım ücretsizdir ve bir çeşit BSD ile lisansına sahiptir. Yazılım ismini bir ekonometri terimi olan veri panelinden almıştır. Bir veri paneli birçok zaman aralığı içinde farklı gözlemlerin işlenebildiği yapıyı tarif eder.

### 2.2.3 NumPy Kütüphanesi

NumPy, Python programlama dili için büyük, çok boyutlu dizileri ve matrisleri destekleyen, bu diziler üzerinde çalışacak üst düzey matematiksel işlevler ekleyen bir kitaplıktır. NumPy'nin atası Numeric, ilk olarak Jim Hugunin tarafından diğer birkaç geliştiricinin katkılarıyla oluşturuldu. 2005 yılında Travis Oliphant, Numarray'in özelliklerini kapsamlı değişikliklerle Numeric'e dahil ederek NumPy'yi yarattı. NumPy açık kaynaklı bir yazılımdır ve birçok katkıda bulunanlara sahiptir.

### 2.2.4 Matplotlib Kütüphanesi

Matplotlib; veri görselleştirmesinde kullandığımız temel python kütüphanesidir. 2 ve 3 boyutlu çizimler yapmamızı sağlar. Matplotlib genelde 2 boyutlu çizimlerde kullanılırken, 3 boyutlu çizimlerde başka kütüphanelerden yararlanılır.

### 2.2.5 Seaborn Kütüphanesi

Seaborn, Matplotlib kütüphanesi tabanlı, istatiksel bir Python veri görselleştirme kütüphanesidir. Seaborn kullanıcılara istatiksel görselleştirmeler yapmaları için high-level (yüksek seviyeli) bir arayüz sunar. Tamamen açık kaynak olan bu kütüphanenin Github reposunu inceleyebilir ve destek verebilirsiniz.

### 2.2.6 Scikit-Learn

Scikit-learn, veri bilimi ve machine learning için en yaygın kullanılan Python paketlerinden biridir. Birçok işlemi gerçekleştirmenizi sağlar ve çeşitli algoritmalar sağlar. Scikit-learn ayrıca sınıfları, yöntemleri ve işlevleri ile kullanılan algoritmaların arka planıyla ilgili belgeler sunar.

Ayrıca, modellerinizi test etmek için kullanabileceğiniz birkaç veri kümesi de sağlar.

### 2.2.7 MissingNo Kütüphanesi

Bu kütüphane, veri ön işleme kısmının önemli bir parçası olan eksik verilerin bulunması kısmında kullanılan bir kütüphanedir.

## 3. AŞAMALAR

### 3.1 Veri Analizi

Pittsburgh Köprüleri veri seti UCI Machine Learning Repositoryden alınmıştır. Bu veri setindeki örnek sayısı 108'dir. Ayrıca 13 özniteliğe sahiptir.

| Öznitelik | Türü |
|---|---|
| Id | Sayısal |
| River | Kategorik |
| Location | Sayısal |
| Erected | Kategorik |
| Purpose | Kategorik |
| Length | Kategorik |
| Lanes | Kategorik |
| Clear_g | Kategorik |
| T_or_d | Kategorik |
| Material | Kategorik |
| Span | Kategorik |

| | |
|---|---|
| Rel_l | Kategorik |
| Type | Kategorik |

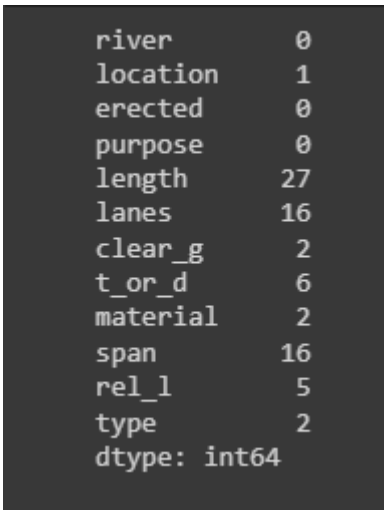
Özniteliklerden 2 tanesi sayısal, geriye kalan 11 tanesi ise kategoriktir.

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 108 entries, 0 to 107
Data columns (total 12 columns):
 #   Column    Non-Null Count  Dtype
---  ------    --------------  -----
 0   river     108 non-null    object
 1   location  107 non-null    float64
 2   erected   108 non-null    object
 3   purpose   108 non-null    object
 4   length    81 non-null     object
 5   lanes     92 non-null     object
 6   clear_g   106 non-null    object
 7   t_or_d    102 non-null    object
 8   material  106 non-null    object
 9   span      92 non-null     object
 10  rel_l     103 non-null    object
 11  type      106 non-null    object
dtypes: float64(1), object(11)
memory usage: 10.2+ KB
None
```

**Şekil 3.1: Değişkenlerin Tip Bilgileri**

| | river | location | erected | purpose | length | lanes | clear_g | t_or_d | material | span | rel_l | type |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | M | 3.0 | CRAFTS | HIGHWAY | MEDIUM | 2 | N | THROUGH | WOOD | SHORT | S | WOOD |
| 1 | A | 25.0 | CRAFTS | HIGHWAY | MEDIUM | 2 | N | THROUGH | WOOD | SHORT | S | WOOD |
| 2 | A | 39.0 | CRAFTS | AQUEDUCT | MEDIUM | 1 | N | THROUGH | WOOD | MEDIUM | S | WOOD |
| 3 | A | 29.0 | CRAFTS | HIGHWAY | MEDIUM | 2 | N | THROUGH | WOOD | SHORT | S | WOOD |
| 4 | M | 23.0 | CRAFTS | HIGHWAY | MEDIUM | 2 | N | THROUGH | WOOD | MEDIUM | S | WOOD |
| 5 | A | 27.0 | CRAFTS | HIGHWAY | SHORT | 2 | N | THROUGH | WOOD | MEDIUM | S | WOOD |
| 6 | A | 28.0 | CRAFTS | AQUEDUCT | MEDIUM | 1 | N | THROUGH | IRON | SHORT | S | SUSPEN |
| 7 | M | 3.0 | CRAFTS | HIGHWAY | MEDIUM | 2 | N | THROUGH | IRON | SHORT | S | SUSPEN |
| 8 | A | 39.0 | CRAFTS | AQUEDUCT | MEDIUM | 1 | N | DECK | WOOD | MEDIUM | S | WOOD |
| 9 | A | 29.0 | CRAFTS | HIGHWAY | MEDIUM | 2 | N | THROUGH | WOOD | MEDIUM | S | WOOD |

**Şekil 3.2: Veri Setinin İlk 10 Verisi**




**Şekil 3.3: Eksik Değer Analizi**

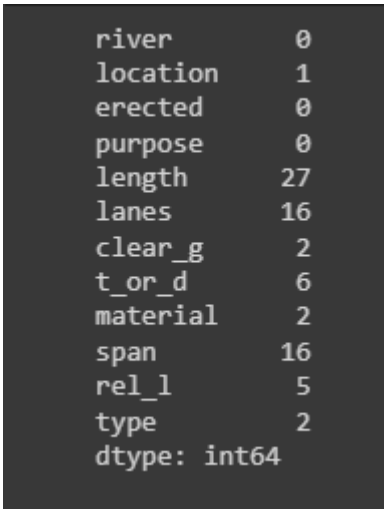
Şekil 3.3'te de görüldüğü üzere location verisinde 1, length verisinde 27, lanes verisinde 16, clear_g verisinde 2, t_or_d verisinde 6, material verisinde 2, span verisinde 16, rel_l verisinde 5 ve son olarak type verisinde 2 adet eksik veri bulunmaktadır.

## 3.2 Veri Ön İşleme

Analiz sonucunun kalitesini artırmak ve algoritmaları uygulayabilmek amacıyla ön veriye ön işleme prosedürleri uygulanır.

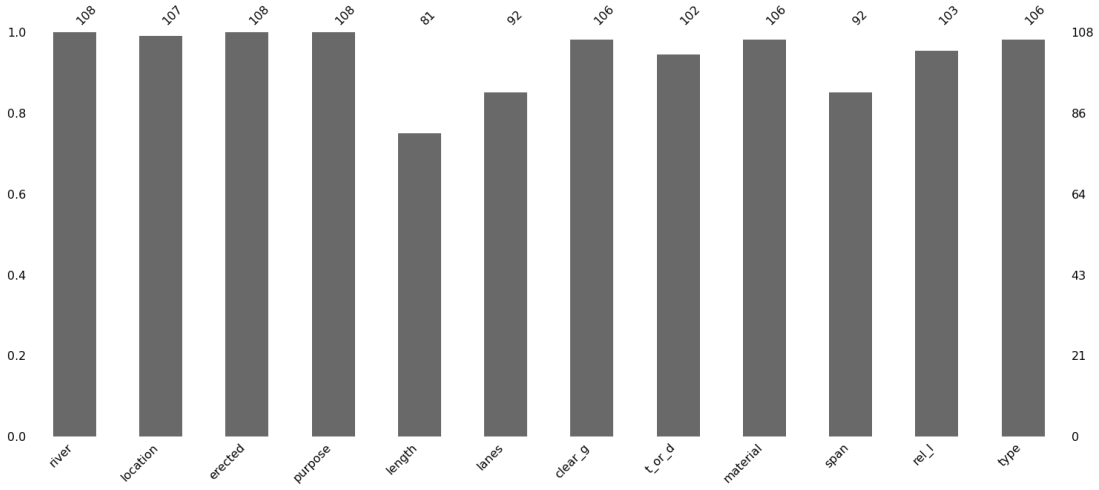
Projede kullanılan ön işleme teknikleri ise eksik değer analizinin yapılması, çıkan eksik değerlerin doldurulması, numerik ve kategorik ayrık değerlerin belirlenmesi ve baskılanması, oversampling, encoding gibi işlemlerdir.

### 3.2.1 Eksik Değer Analizi

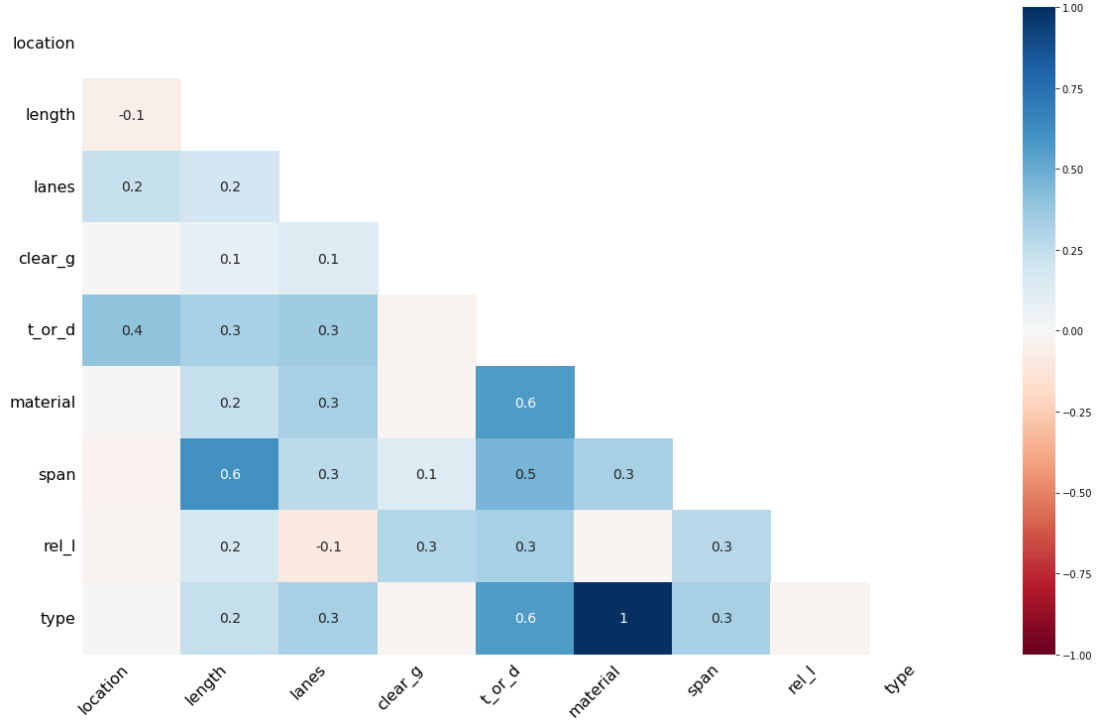



**Şekil 3.4: Eksik Değerler**

Şekil 3.4’te de görüldüğü üzere location verisinde 1, length verisinde 27, lanes verisinde 16, clear_g verisinde 2, t_or_d verisinde 6, material verisinde 2, span verisinde 16, rel_l verisinde 5 ve son olarak type verisinde 2 adet eksik veri bulunmaktadır.




**Şekil 3.5: Eksik Verilerin Grafik Olarak Gösterilmesi**




**Şekil 3.6: Eksik Değerlerin Korelasyonu**

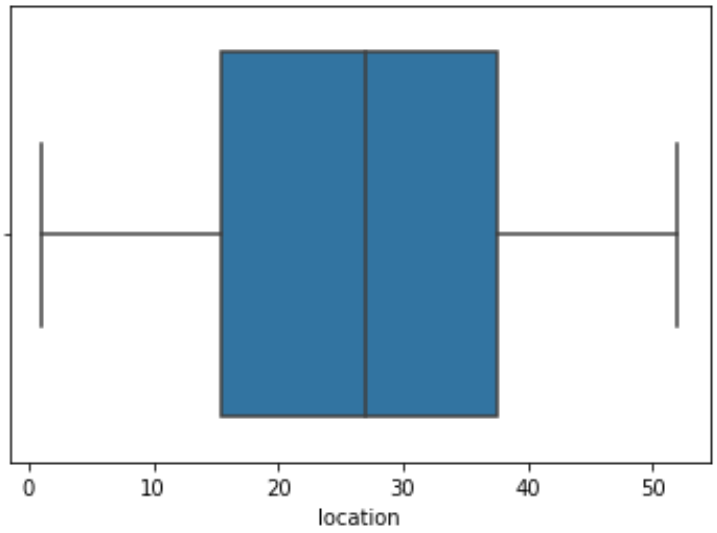
Bu veriler algoritmaların uygulanması ve analiz kalitesini artırabilmek amacıyla doldurulmuştur. Doldurma işlemi sayısal veriler için medyan ile, kategorik verilerde ise mod yardımıyla yapılmıştır.

### 3.2.2 Aykırı Değer Analizi

Veriden sağlıklı çıkarımlar yapmayı güçleştiren önemli engellerden bir tanesi de veri setindeki aykırı (outlier) değerlerdir. Aykırı değerler, diğer gözlemlerden kayda değer derecede uzak olan gözleme aykırı veya uç değer denir. Aykırı gözlemler, veri setinin geri kalanından farklı davranır ve bu nedenle dikkat çeker. Aykırı gözlemlerin tespit edilip, gözden geçirilmesi ve duruma göre müdahale edilmesi gerekmektedir. Aykırı değerler:

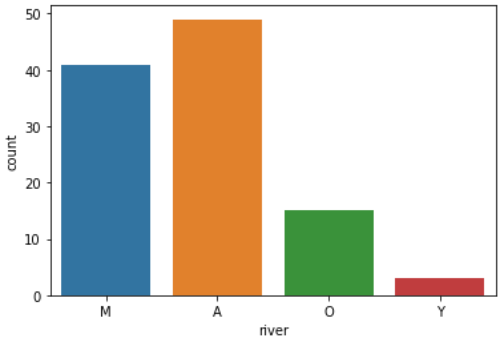
- Verilerin dağılımını ve ortalama, medyan vs. gibi veriyi temsil eden istatistikleri etkiler.
- Modellerden elde edilen sonuçlara etki eder.
- İstatistiksel testlerin gücünü düşürür.

#### 3.2.2.1 Numerik Bağımsız Değişkenlerin Analizi

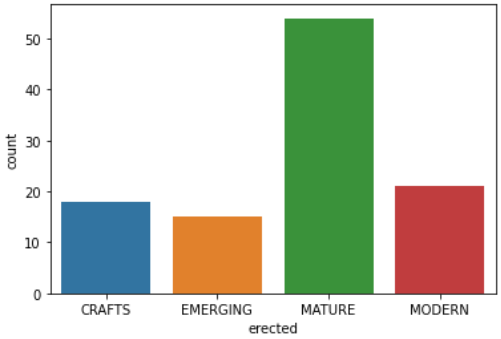



**Şekil 3.7: Numerik Bağımsız Değişkenlerin Analizinin BoxPlotta Gösterilmesi**

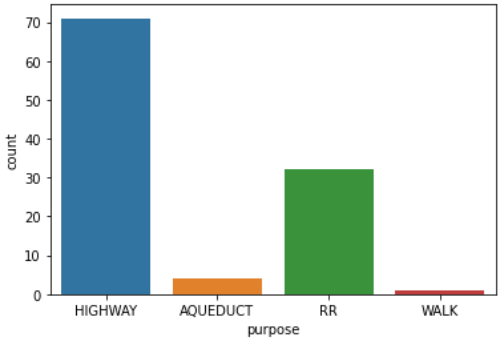
#### 3.2.2.2 Kategorik Bağımsız Değişkenlerin Analizi



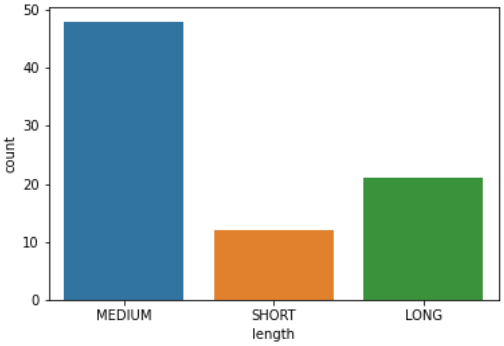

**Şekil 3.8: River Özniteliğinin İncelenmesi**



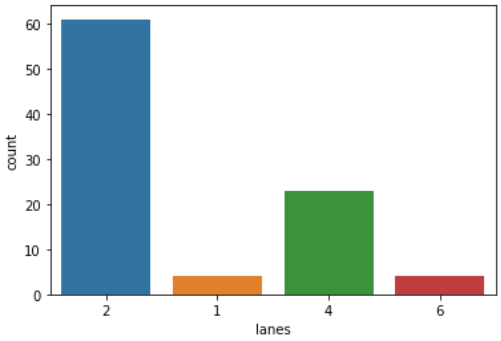

**Şekil 3.9: Erected Özniteliğinin İncelenmesi**



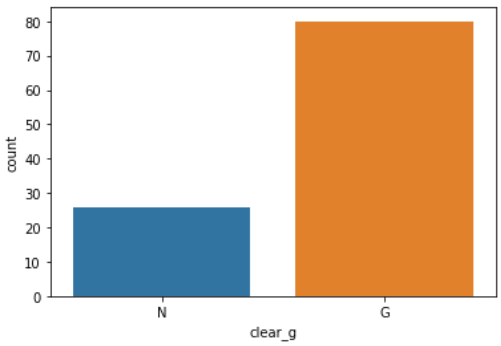

**Şekil 3.10: Purpose Özniteliğinin İncelenmesi**



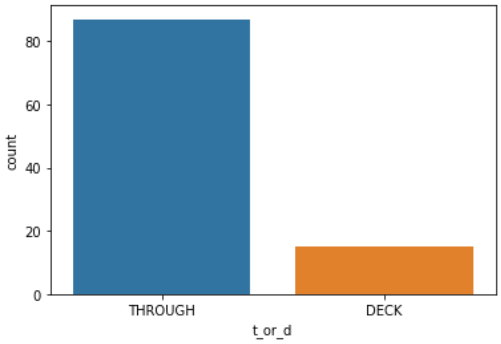

**Şekil 3.11: Length Özniteliğinin İncelenmesi**



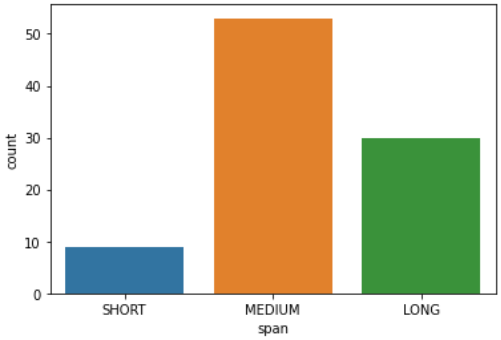

**Şekil 3.12: Lanes Özniteliğinin İncelenmesi**



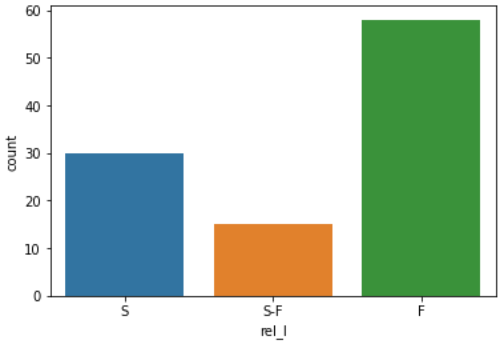

**Şekil 3.13: Clear-g Özniteliğinin İncelenmesi**



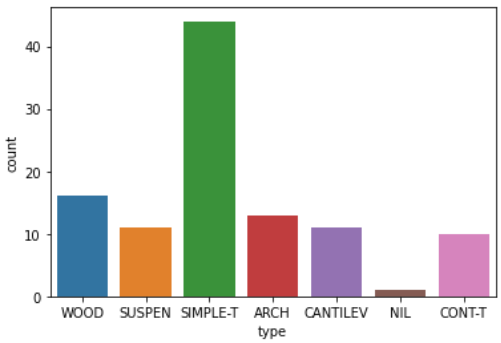

**Şekil 3.14: Through or Deck Özniteliğinin İncelenmesi**




**Şekil 3.15: Span Özniteliğinin İncelenmesi**




**Şekil 3.16: Rel_l Özniteliğinin İncelenmesi**




**Şekil 3.17: Type Özniteliğinin İncelenmesi**

### 3.2.3 Oversampling

Oversampling yani aşırı örnekleme, eşit sınıf dağılımları elde edilene kadar azınlık sınıfının örneklerini çoğaltır. Bu konudaki yöntemlerinin çoğu, azınlık sınıfının örneklerini kopyaladığından, aşırı öğrenme(overfiting) olma olasılığı artar. Ayrıca, yüksek düzeyde dengesiz dağılıma sahip büyük bir veri kümesi olması durumunda, aşırı örnekleme hesaplama açısından çok maliyetli olabilir.

Veri setinde veri sayısının azlığı nedeni sebebiyle oversampling yöntemi kullanılarak veri eklemesi yapılmıştır.

### 3.2.4 One Hot Encoding

One Hot Encoding, kategorik değişkenlerin ikili (binary) olarak temsil edilmesi anlamına gelmektedir. Bu işlem ilk önce kategorik değerlerin tamsayı değerleriyle eşlenmesini gerektirir. Daha sonra, her bir tamsayı değeri, 1 ile işaretlenmiş tamsayı indeksi dışında ki tüm değerleri sıfır olan bir ikili vektör olarak temsil edilir. Örneğin aşağıda 3 kategoride veri vardır apple,chicken ve broccoli. Bu alanlar binary olarak ayrıştırıldığında apple için ilk satır 1 iken diğerleri 0 olur. Diğer veri içinde aynı şekilde sayısal veriye çevirme işlemi devam eder.

| | location | river_M | river_O | river_Y | erected_EMERGING | erected_MATURE | \ |
|---|---|---|---|---|---|---|---|
| 0 | 3.0 | 1 | 0 | 0 | 0 | 0 | |
| 1 | 25.0 | 0 | 0 | 0 | 0 | 0 | |
| 2 | 39.0 | 0 | 0 | 0 | 0 | 0 | |
| 3 | 29.0 | 0 | 0 | 0 | 0 | 0 | |
| 4 | 23.0 | 1 | 0 | 0 | 0 | 0 | |
| .. | ... | ... | ... | ... | ... | ... | |
| 103 | 24.0 | 0 | 0 | 0 | 0 | 0 | |
| 104 | 44.0 | 0 | 1 | 0 | 0 | 0 | |
| 105 | 7.0 | 1 | 0 | 0 | 0 | 0 | |
| 106 | 43.0 | 0 | 1 | 0 | 0 | 0 | |
| 107 | 28.0 | 0 | 0 | 0 | 0 | 0 | |

| | erected_MODERN | purpose_HIGHWAY | purpose_RR | purpose_WALK | ... | lanes_4 | \ |
|---|---|---|---|---|---|---|---|
| 0 | 0 | 1 | 0 | 0 | ... | 0 | |
| 1 | 0 | 1 | 0 | 0 | ... | 0 | |
| 2 | 0 | 0 | 0 | 0 | ... | 0 | |
| 3 | 0 | 1 | 0 | 0 | ... | 0 | |
| 4 | 0 | 1 | 0 | 0 | ... | 0 | |
| .. | ... | ... | ... | ... | ... | ... | |
| 103 | 1 | 1 | 0 | 0 | ... | 0 | |
| 104 | 1 | 1 | 0 | 0 | ... | 0 | |
| 105 | 1 | 1 | 0 | 0 | ... | 0 | |
| 106 | 1 | 1 | 0 | 0 | ... | 0 | |
| 107 | 1 | 1 | 0 | 0 | ... | 0 | |

| | lanes_6 | clear_g_N | t_or_d_THROUGH | material_STEEL | material_WOOD | \ |
|---|---|---|---|---|---|---|
| 0 | 0 | 1 | 1 | 0 | 1 | |
| 1 | 0 | 1 | 1 | 0 | 1 | |
| 2 | 0 | 1 | 1 | 0 | 1 | |
| 3 | 0 | 1 | 1 | 0 | 1 | |
| 4 | 0 | 1 | 1 | 0 | 1 | |
| .. | ... | ... | ... | ... | ... | |
| 103 | 1 | 0 | 1 | 1 | 0 | |
| 104 | 1 | 0 | 1 | 1 | 0 | |
| 105 | 1 | 0 | 1 | 1 | 0 | |
| 106 | 0 | 0 | 1 | 1 | 0 | |
| 107 | 0 | 0 | 1 | 1 | 0 | |

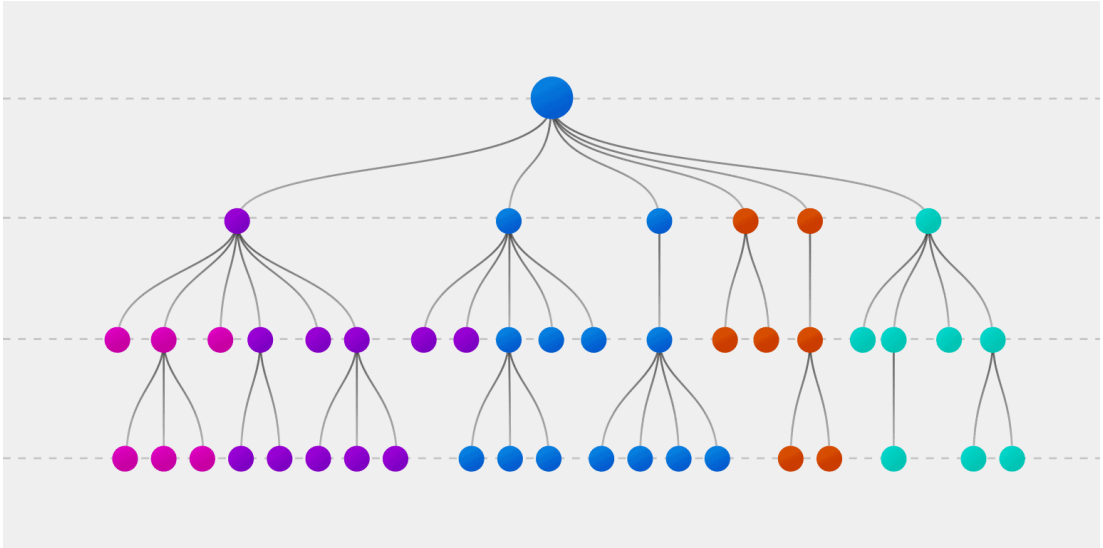
| | span_MEDIUM | span_SHORT | rel_l_S | rel_l_S-F |
|---|---|---|---|---|
| 0 | 0 | 1 | 1 | 0 |
| 1 | 0 | 1 | 1 | 0 |
| 2 | 1 | 0 | 1 | 0 |
| 3 | 0 | 1 | 1 | 0 |
| 4 | 1 | 0 | 1 | 0 |
| .. | ... | ... | ... | ... |
| 103 | 1 | 0 | 0 | 0 |
| 104 | 0 | 0 | 0 | 0 |
| 105 | 0 | 0 | 0 | 0 |
| 106 | 1 | 0 | 0 | 0 |
| 107 | 1 | 0 | 0 | 0 |

**Şekil 3.18: One Hot Encoding Sonrası Öznitelikler**

# 4. MODELLEME

## 4.1 Model Seçilmesi

Model olarak daha önce de bahsedilen karar ağacı metodu (decision tree method) kullanılmıştır.



**Şekil 4.1: Decision Tree**

## 4.2 Model Doğrulaması (Validation)

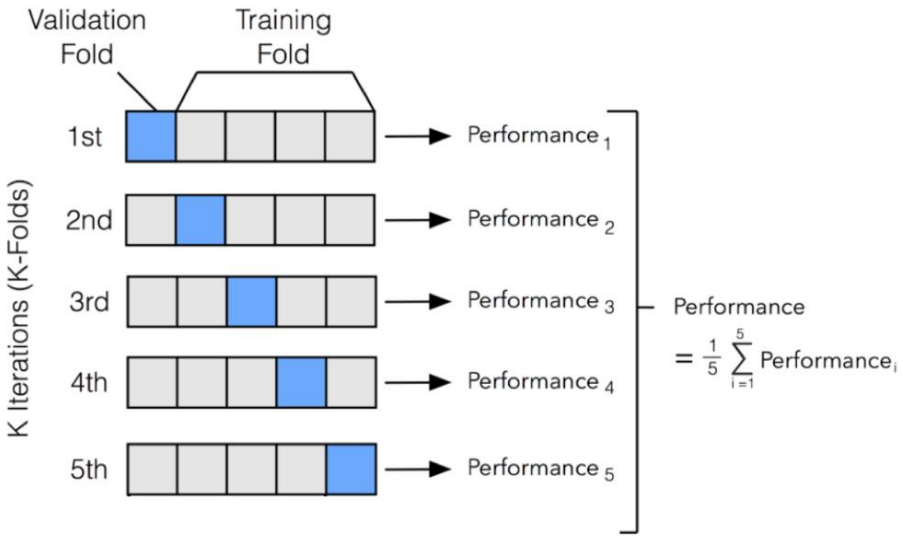
Sınıflandırma problemlerinde öncelikle veri seti train ve test setleri olarak ayrılır daha sonra ise train veri seti üzerinden model oluşturulur ve test veri seti üzerinde yapılan tahminleri test edilir. Ancak train/test ayrımında bazı problemler bulunabilir . Veri seti ayrımının rastgele yapılamamış olma ihtimali bulunmaktadır. Sadece belirli bir yaşta, belirli bir bölgeden, sadece kadın ya da erkekleri seçip onlar üzerinden model kurulmuş olunabilir. Bu da overfitting problemine sebep olacaktır. Bu problem ise modelin doğrulanması yani validasyon yapılması ile çözümlenebilir.

Bu projede kullanılan validasyon yöntemi ise K-Fold Cross Validation yöntemidir.

### 4.2.1 K-Fold Cross Validation

K-Folds Cross Validation'da veriler k farklı alt kümeye bölünür. Verileri eğitmek ve son alt kümeyi test verisi olarak bırakmak için k-1 adet alt küme kullanılır. k adet deney sonucunda ortaya çıkan ortalama hata değeri modelin geçerliliğini belirtir.

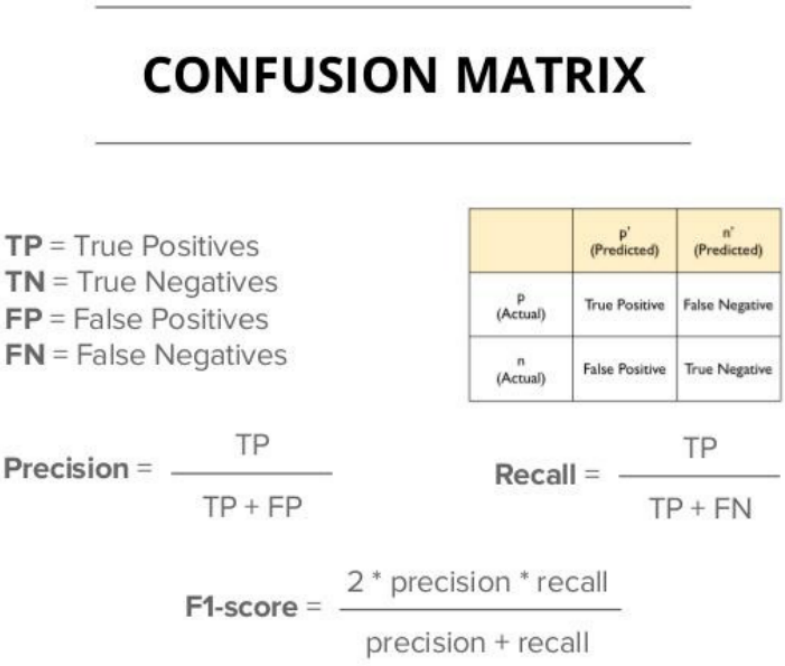
K değeri genellikle 3 ya da 5 olarak seçilmektedir. Bu değer 10 ya da 15 de seçilebilir ancak bu oldukça pahalı bir hesaplamaya ve zaman kaybına sebep olacaktır.




**Şekil 4.2: K-Fold Cross Validation**

## 4.3 Modelin Başarısının Ölçülmesi

Modelin başarılı tahminler yapıp yapmadığını ölçmek için çeşitli metrikler bulunur. Bu metriklerden bazıları accuracy, precision, f1 score, recall gibi metriklerdir.




**Şekil 4.3: Confusion Matrix**

| Metrik | Sonuç |
|---|---|
| Accuracy | 0.86 |
| Precision | 0.87 |
| F1 Score | 0.86 |
| Recall | 0.86 |

Modelin başarı oranı da tabloda görüldüğü gibidir. Yüzde 86-87 dolaylarında bir başarı elde edilmiştir.

### 4.3.1 Sonuçların Diğer Çalışmalarla Karşılaştırılması

| | precision | recall | f1-score | support |
|---|---|---|---|---|
| DECK | 0.85 | 0.73 | 0.79 | 15 |
| THROUGH | 0.96 | 0.98 | 0.97 | 87 |
| accuracy | | | 0.94 | 102 |
| macro avg | 0.90 | 0.86 | 0.88 | 102 |
| weighted avg | 0.94 | 0.94 | 0.94 | 102 |

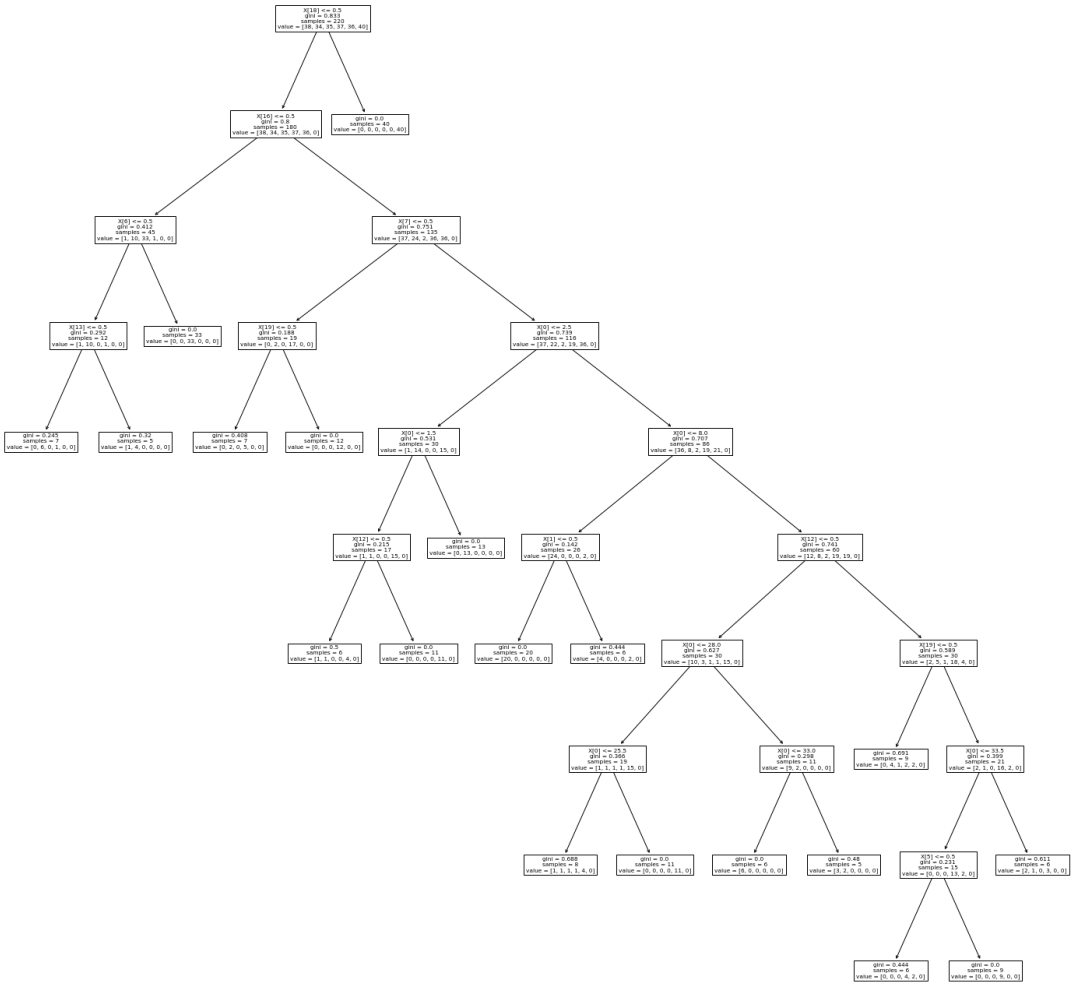
**Şekil 4.4: Karşılaştırma Yapılan Çalışmanın Sonuçları**

| | precision | recall | f1-score | support |
|---|---|---|---|---|
| 0 | 0.88 | 0.88 | 0.88 | 8 |
| 1 | 0.73 | 0.92 | 0.81 | 12 |
| 2 | 1.00 | 0.82 | 0.90 | 11 |
| 3 | 0.80 | 0.89 | 0.84 | 9 |
| 4 | 0.88 | 0.70 | 0.78 | 10 |
| 5 | 1.00 | 1.00 | 1.00 | 6 |
| accuracy | | | 0.86 | 56 |
| macro avg | 0.88 | 0.87 | 0.87 | 56 |
| weighted avg | 0.87 | 0.86 | 0.86 | 56 |

**Şekil 4.5: Bu Çalışmada Elde Edilen Sonuçlar**

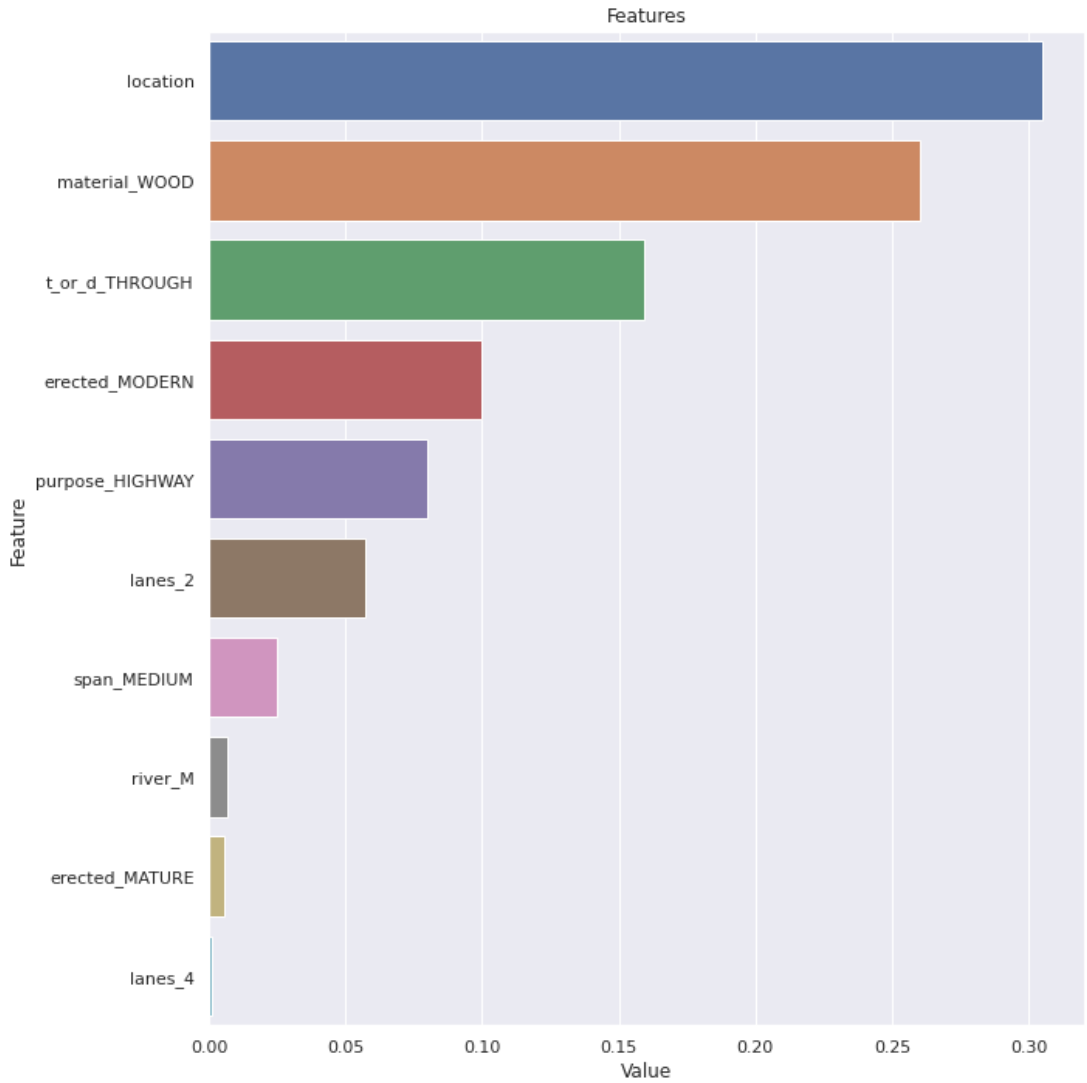
Şekillerde görüldüğü üzere internetten alınan çalışmada alınan sonuçlar, bu raporda incelenen çalışmaya göre daha başarılı olmuştur.

## 4.4 Karar Ağacının Görselleştirilmesi



**Şekil 4.6: Modelin Karar Ağacı**

## 4.5 Modelin Feature Importances Değerleri




**Şekil 4.7: Feature Importances**

## 5. SONUÇ

Bu projede Pittsburgh köprüleri veri seti analiz edilmiş, eksik veriler bulunmuş ve doldurulmuş, ayrık değerler incelenmiş, model oluşturulmuş ve bu model cross validationa uğratılmış, son olarak da modelin başarısı incelenmiştir.

Modelin oluşturulmasında Decision Tree Metodolojisi kullanılmıştır. Ayrıca cross validation işleminde de K-Fold Cross Validation seçimi yapılmıştır.

Modelin başarısına baktığımızda ise yüzde 86 – 87 civarlarında başarılı sonuçlar elde edilmiştir.

## 6. KAYNAKÇA


- https://archive.ics.uci.edu/ml/datasets/Pittsburgh+Bridges
- https://www.veribilimiokulu.com/bir-bakista-k-fold-cross-validation/
- https://womaneng.com/one-hot-encoding-nedir-nasil-yapilir/
- https://ravenfo.com/2021/02/11/aykiri-deger-analizi/
- https://bayramadali.wordpress.com/one-hot-encoding/
- https://bilgisayarkavramlari.com/2013/03/31/k-fold-cross-validation-k-katlamali-carpraz-dogrulama/
- https://womaneng.com/cross-validation-nedir-capraz-dogrulama-nedir/
- https://www.kaggle.com/code/heitornunes/predicting-5-design-properties-cart