

## BİL 470/570 Ödev 1

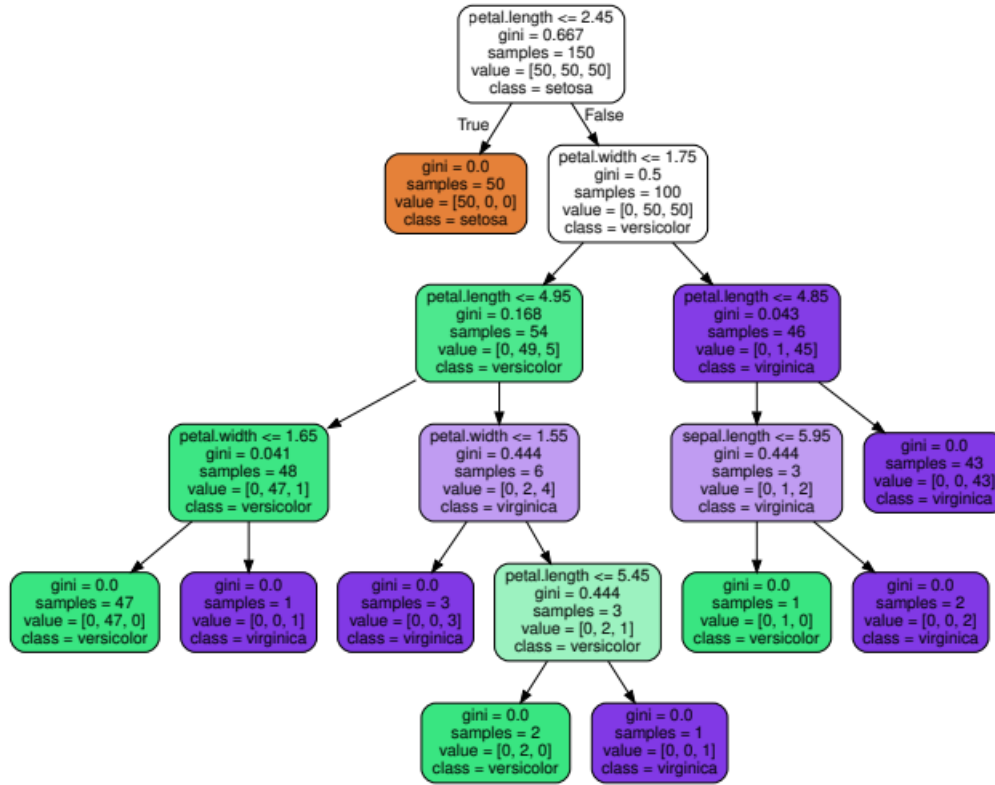


Figure 1: IRIS veri kümesinde eğitilmiş DecisionTreeClassifier'in görselleştirilmesi

### Amaç:

Bu ödevde, iris veri kümesi üzerinde bir keşifsel veri analizi (exploratory data analysis - EDA) gerçekleştireceksiniz. Bunun yanında herhangi bir kütüphaneden yararlanmadan bir karar ağacı sınıflandırıcısı implamente edeceksiniz, bu sınıflandırıcısı kullanarak iris veri kümesi üzerinden eğitim yapacak ve sınıflandırmanın sonucunu yorumlayacaksınız.

## 1 GÖREVLER

### 1.1 EDA

Iris veri seti, makine öğrenmesi literatüründe en iyi bilinen veri setidir. 150 örnek, 3 sınıfın her biri için 50 örnek içerir. <https://www.kaggle.com/datasets/uciml/iris> üzerinden veri setini indirebilirsiniz. İlk olarak sizden beklenen; sınıf etiketleri veri kümesinin son sütununda tür adları olarak verilen setosa, versicolor, virginica tür adlarını sırasıyla 0,1,2 olmak üzere tam sayılara çevirmeniz.

Daha sonra, bu veri kümesinde bir EDA gerçekleştirin. EDA, **veri seti özetinden**, özniteliklerin (features) ve hedefin (Target) **korelasyon matrisinden** (correlation matrix) ve öznitelikler için oluşturulacak **pair-plots** içermelidir. Bu konuda daha fazla ayrıntı vermek istiyorsanız ek şeyler ekleyin.

Sonuçlarınızı bu bölümde göstermek için **pandas, numpy, seaborn, matplotlib** gibi kütüphanelerini kullanabilirsiniz. İsterseniz ek kütüphaneler eklemekten çekinmeyin.

## 1.2 DecisionTreeClassifier

Derste öğrendiğiniz DecisionTreeClassifier'ı **herhangi bir kütüphane kullanmadan** implamente edin. Ağacın düğümlerini bölmek için Gini Saflığını (**Gini Impurity**) (Eq. 1) kullanın. Karar ağacının maksimum derinliğini (max\_depth) sınıfı çağırırken overfit üstesinden gelmek için bir argüman olarak alın, çünkü sonsuz derinlikli karar ağacı sınıflandırıcısı bildiğiniz gibi eğitim verilerini her zaman overfit eder.

$$G = 1 - \sum_{k=1}^n p_k^2$$

Formül 1: Gini Impurity

$p_k$ , k sınıfına ait örneklerin oranıdır.

Yukarıda belirtilen sınıflandırıcının imzası aşağıdaki gibi olacaktır:

- **DecisionTreeClassifier(max\_depth=5)**

Verilerin ilk %80'ini kullanarak Sınıflandırıcıyı eğitin ve kalan verilerle sınıflandırıcıyı test edin.

DecisionTreeClassifier'ı implamente ederken **herhangi bir kütüphane kullanamazsınız**. Vektörler için list yapısını ve 2D inputlar için list of list kullanabilirsiniz.

## 1.3 Sonuçlar

Train ve Test için ayrı ayrı;

- confusion matrix görüntüle
- F1-Scor
- Accuracy
- Precision
- Recall değerlerini hespla
- Receiver Operating Characteristic (**ROC**) eğrisini çizin ve altındaki alanı (AUC) hesaplayın
- Sonuçlar hakkında yorum yapın

## 2 Gönderme

3 dosya göndereceksiniz 2

1. **Python dosyası (dt.py):**

DecisionTreeClassifier implamentasyonunu içeri. **Bu dosyada herhangi bir kitaplık kullanamazsınız.**

2. **Notebook dosyası (report.ipynb):**

3 kısım içerir;

- (1) Iris veri setinin keşifsel veri analizi (EDA)
- (2) Sınıflandırıcının eğitimi ve
- (3) Sonuçların yorumlanması.

Adımları açıklamak için markdown syntax'ını kullanabilir, modeli eğitmek için python kodu yazabilirsiniz,

3. **Rapor (report.pdf):**

İlgili rapor.ipynb dosyasının PDF dışa aktarımıdır.

Bu dosya not defteri dosyasıyla aynı içeriğe sahip olmalıdır.

Bu dosyayı **File > Download as > .pdf** jupyter notebook menüsünden indirin.

### 3 Geliştirme Ortamı

Bu derste Python3 kullanılacaktır.

- Önerim bu ders için Anaconda kurulumunu gerçekleştirmeniz.  
<https://www.geeksforgeeks.org/how-to-install-anaconda-on-windows/>  
Burada detaylı kurulum anlatılmaktadır. Bu şekilde conda environmentlarını kullanabilirsiniz. Bunun yanında Conda python içermektedir.
- Bir IDE seçin: VSCode veya Anaconda Spyder
- [Jupyter Notebook](#) indirin. Eğer Anaconda indirdiyseniz bu adımı atlayabilirsiniz.

**Bireysel bir çalışma olmalıdır. Grup şeklinde yapılmamalıdır. Eğer çalışmanızın orijinallikinden şüphe edilirse demoya çağrılacaksınız. Beraber yapıldığı anlaşılırsa çalışmadan 0 alınacaktır.**