En popüler veri setlerinden “iris” veri setidir.



İris veri seti; taç yaprak uzunluğu ve genişliği, çanak yaprak uzunlugu ve genişliği bilgileri verilen 3 İris bitki türüne (*Iris versicolor*, *Iris virginica* ve *Iris setosa*) ait, her bir türden 50 adet olmak üzere toplamda 150 adet örnek içeren bir veri setidir. Bu proje ile amacımız yaprak boyutları verilen bitkinin hangi türe ait olduğunu tahmin etmek. Anlaşılacağı üzere bu bir sınıflandırma problemidir. Artık problemi belirlediğimize göre adım adım projemizi oluşturmaya başlayalım:

1. Gerekli kütüphanelerin import edilmesi
2. Veri setinin yüklenmesi ve hazırlanması
3. Veri setinin incelenmesi
4. Veri setinin görselleştirilmesi
5. Algoritmaların uygulanması ve sonuçlarının degerlendirilmesi
6. Uygun algoritmanın seçilmesi ve tahmin yapılması

**1.Adım : Gerekli kütüphanelerin import edilmesi**

Burada veri seti işlemleri için [Pandas](https://pandas.pydata.org/) , görselleştirme işlemleri için [Matplotlib](https://matplotlib.org/) , makine öğrenimi algoritmalarını uygulamak ve sonuçlarını değerlendirmek için [Sklearn](http://scikit-learn.org/) kütüphanelerini import edeceğiz.

import matplotlib.pyplot as plt  
import pandas as pd  
from pandas.plotting import scatter\_matrix  
from sklearn import model\_selection  
from sklearn.discriminant\_analysis import LinearDiscriminantAnalysis  
from sklearn.linear\_model import LogisticRegression  
from sklearn.metrics import accuracy\_score  
from sklearn.metrics import classification\_report  
from sklearn.metrics import confusion\_matrix  
from sklearn.naive\_bayes import GaussianNB  
from sklearn.neighbors import KNeighborsClassifier  
from sklearn.svm import SVC  
from sklearn.tree import DecisionTreeClassifier

**2. Adim : Veri setinin yüklenmesi ve hazırlanması**

Pandas kütüphanesi ile veri setimizi yükledik , bağımlı(tür ismi) ve bağımsız(yaprak uzunluk ve genişlikleri) değişkenleri belirledik ve son olarak veri setimizi eğitim ve test verileri olmak üzere ikiye ayırdık.

# Veri setinin yüklenmesi  
iris\_dataset = pd.read\_csv(’iris.csv’)  
  
# Bağımlı ve bağımsız değişkenlerin oluşturulması  
X = iris\_dataset.values[:, 0:4]  
Y = iris\_dataset.values[:, 4]  
  
# Veri kümesinin eğitim ve test verileri olarak ayrılması  
X\_train, X\_test, Y\_train, Y\_test = model\_selection.train\_test\_split(X, Y, test\_size=0.20, random\_state=7)

**3.Adım : Veri setinin incelenmesi**

Veri setinin daha iyi anlaşılması için farklı yönleriyle ele alalım :

* Veri setindeki tüm satırlar

print("Veri setinin içeriği")  
print(iris\_dataset)

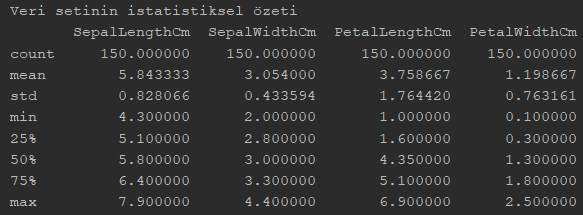
* Veri setinin boyutu

print("Veri setinin boyutu")  
print(iris\_dataset.shape)

**Output**: Veri setinin boyutu  
(150, 6)

* Veri setinin istatiksel özeti

print("Veri setinin istatistiksel özeti")  
print(iris\_dataset.describe())



* Veri setinin tür değişkenine göre dağılımı

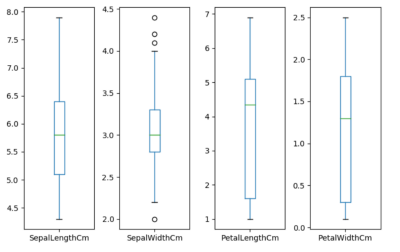
print("verilerin tür değişkenine göre dağılımı")  
print(iris\_dataset.groupby(’Species’).size())

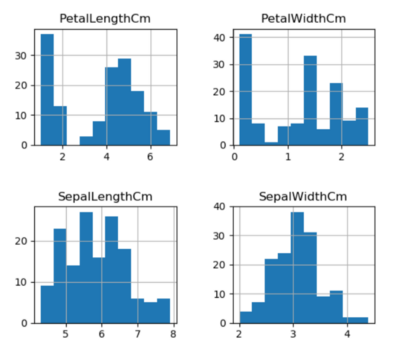
**Output**: verilerin tür değişkenine göre dağılımı  
Iris-setosa 50  
Iris-versicolor 50  
Iris-virginica 50

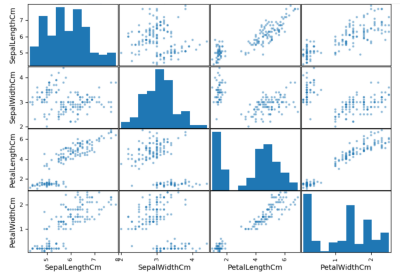
**4.Adım : Veri setinin görselleştirilmesi**

Görselleştirme işlemi veriyi daha iyi tanımamızı ve hangi modellerin veri setimize daha uygun oldugunu öngörebilmemizi sağlar . Makine öğrenimi alanında henüz yeniyseniz sadece grafiklere bakarak veri setiniz için en uygun modele karar verebilmeyi şimdilik beklemeyin. Çünkü bunu diyebilmek farklı türlerden bir çok veri seti ile çalışmış olmayı gerektirir, yani biraz daha zamana ve tecrübeye ihtiyacınız var.

# kutu grafigi  
iris\_dataset.plot(kind=’box’, subplots=True, sharex=False, sharey=False)  
plt.show()  
  
# histogram  
iris\_dataset.hist()  
plt.show()  
  
# scatter plot matrix  
scatter\_matrix(iris\_dataset)  
plt.show()







Kutu grafigi-Histogram-Scatter plot matrix

**5.Adım : Algoritmaların uygulanması ve sonuçlarının değerlendirilmesi**

Bu adımda veri setimize en uygun modeli arıyoruz bu sebeple farklı modelleri uygulayıp, ‘[cross validation](http://scikit-learn.org/stable/modules/cross_validation.html)’ sonuçlarını karşılaştırarak en doğru sonucu veren modeli seçeceğiz.

# Modellerin listesinin olusturulmasi  
models = [  
 (’LR’, LogisticRegression()),  
 (’LDA’, LinearDiscriminantAnalysis()),  
 (’KNN’, KNeighborsClassifier()),  
 (’DT’, DecisionTreeClassifier()),  
 (’NB’, GaussianNB()),  
 (’SVM’, SVC())  
]  
  
# Modeller için 'cross validation' sonuçlarının yazdırılması  
results = []  
names = []  
for name, model in models:  
 kfold = model\_selection.KFold(n\_splits=10, random\_state=7)  
 cv\_results = model\_selection.cross\_val\_score(model, X\_train, Y\_train, cv=kfold, scoring="accuracy")  
 results.append(cv\_results)  
 names.append(name)  
 print("%s: %f (%f)" % (name, cv\_results.mean(), cv\_results.std()))

**Output**:  
LR: 0.966667 (0.040825)  
LDA: 0.975000 (0.038188)  
KNN: 0.983333 (0.033333)  
DT: 0.966667 (0.040825)  
NB: 0.975000 (0.053359)  
SVM: 0.991667 (0.025000)

Sonuçları değerlendirmek için accuracy (doğruluk ) değerlerine bakıyoruz. Accuracy değeri en yüksek olan modelin veri setimiz için en uygun model olduğunu varsayıyoruz. Sonuçları incelediğimizde SVM in en yüksek accurucay değerine sahip olduğunu görüyoruz ve bu sebeple sonraki adımda SVM modelini kullanarak tahmin yapacagiz.

Not : Eğitim ve test kümeleri random olarak bölündüğü için siz uyguladığınızda daha farklı sonuçlar alabilirsiniz. Nerede hata yaptik diye düşünmenize gerek yok :).

**6.Adım : Uygun algoritmanın seçilmesi ve tahmin yapılması**

SVM modelini X\_train ve Y\_train verilerini kullanarak eğiteceğiz. Modelimizi eğittikten sonra artık test için ayırdığımız verilerin türünü tahmin edebiliriz. Tahmin değerleri(predictions) ile gerçek test değerlerini (Y\_test ) karşılaştırarak algoritmamızın doğruluğu hakkında bir fikir edinebiliriz.

svc = SVC()  
svc.fit(X\_train, Y\_train)  
predictions = svc.predict(X\_test)  
  
print('accuracy degeri :', accuracy\_score(Y\_test, predictions))  
print(confusion\_matrix(Y\_test, predictions))  
print(classification\_report(Y\_test, predictions))

