|  |  |
| --- | --- |
| MAKİNE ÖĞRENMESİ RAPORU |  |
|  |  |
| ŞEREFHAN DİŞEK  180101062 |  |

# MAKİNE ÖĞRENMESİ VE MODELLERİ

Makine öğrenimi, bilgisayar sistemlerinin veri kullanarak deneyim kazanmasını ve kendilerini geliştirmesini sağlayan yapay zeka alanının bir alt dalıdır. Bu teknoloji, karmaşık veri setlerinden desenler ve ilişkiler çıkarmak için algoritmalar kullanır ve bu bilgileri tahminlerde bulunmak, kararlar vermek veya görevleri otomatikleştirmek için kullanır. Makine öğrenimi, geniş bir uygulama yelpazesine sahiptir ve otomatik sürücüden sağlık hizmetlerine, finansal tahminlerden doğal dil işleme uygulamalarına kadar birçok alanda kullanılabilir.

Makine öğrenimi modelleri, veri setlerini analiz etmek, öğrenmek ve sonuçlar üretmek için kullanılan matematiksel ve istatistiksel tekniklerdir. Bu modeller, veri setlerindeki kalıpları tanımlamak ve gelecekteki olayları tahmin etmek için kullanılır. Başlıca makine öğrenimi modelleri arasında destek vektör makineleri (SVM), karar ağaçları, k-NN (k en yakın komşu), doğrusal regresyon, lojistik regresyon, derin öğrenme ve daha fazlası bulunmaktadır.

Bu rapor, makine öğrenimi konseptlerini ve yaygın olarak kullanılan makine öğrenimi modellerini tanıtmayı amaçlamaktadır. İlerleyen bölümlerde, her bir modelin özellikleri, avantajları ve dezavantajları hakkında daha ayrıntılı bilgi verilecektir.



# QSAR BIOCONCENTRATION CLASSES VERİ SETİ

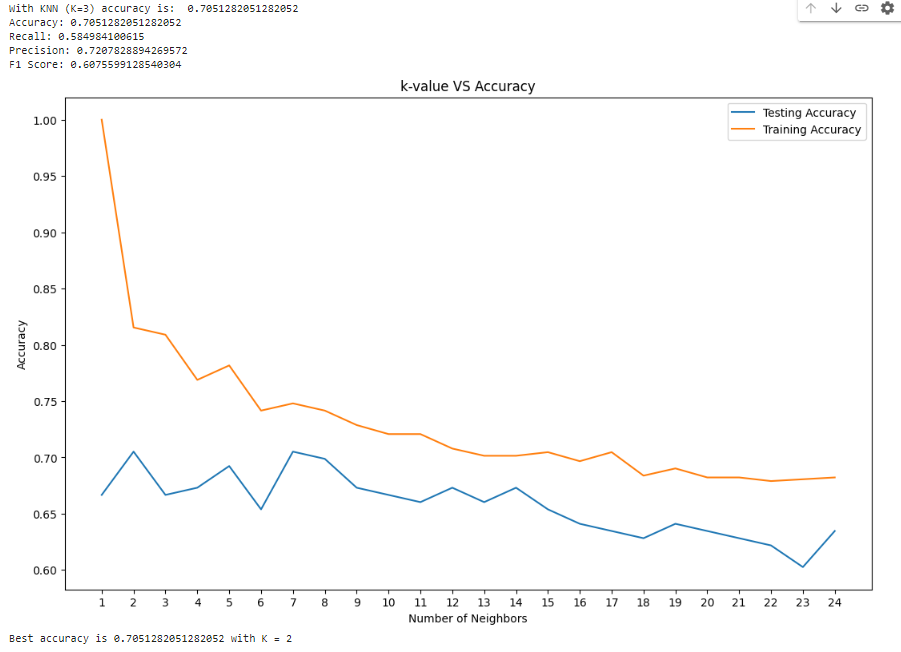
# Bu projede QSAR Bioconcentration Classes veri seti kullanılmıştır. Bu veri seti QSAR modelleme için manuel olarak seçilen Biyokonsantirasyon faktörünün ve mekanik sınıfların veri kümesidir. Çok değişkenli dataset özelliklerine sahip olup, sınıflandırma ve regresyon konuları için kullanılır. Veri seti 879 instance ve 14 feature içermektedir. Veri seti, gereksiz veya tekrarlayan özelliklerin kaldırılmasıyla önceden işlenmiştir. Ardından, veriler normalleştirilmiştir. X ve y, özellikler ve hedef değişkenler olarak ayrılmıştır.

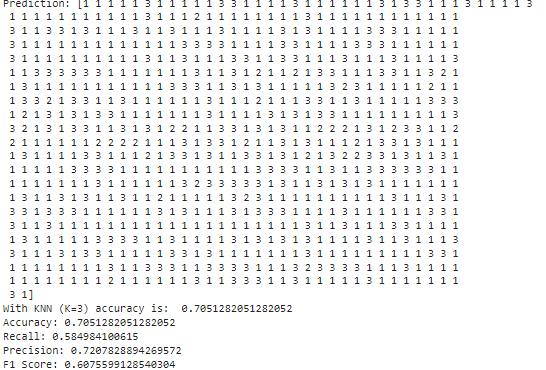
# KNN MODELİ

İlk olarak, tüm veri seti KNN algoritması ile eğitilmiştir. Daha sonra, veri seti train-test olarak ayrılmıştır. Eğitim veri seti üzerinde model tekrar eğitilmiş ve test veri seti üzerinde performansı değerlendirilmiştir.

Modelin performansını değerlendirmek için Accuracy, Recall, Precision ve F1 skoru gibi temel performans metrikleri kullanılmıştır. Bu metrikler, modelin sınıflandırma yeteneğini ölçmek için kullanılır.

KNN modelinin karmaşıklığı, K değerine bağlı olarak değişir. Bu nedenle, farklı K değerleri için modelin eğitim ve test doğruluğu karşılaştırılmış ve optimal K değeri belirlenmiştir. Grafik üzerindeki eğriler, K değeri arttıkça modelin karmaşıklığının nasıl değiştiğini gösterir.





# DECISION TREE

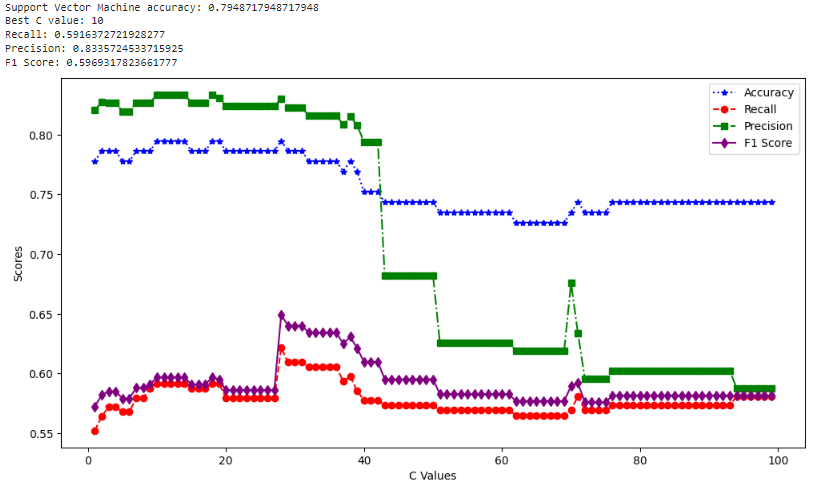
# Decision Tree modeli, DecisionTreeClassifier sınıfı kullanılarak oluşturulmuştur. Model, eğitim verisi üzerinde eğitilmiştir. Model oluşturulurken kullanılan parametreler arasında criterion=”entropy” (bilgi kazancı hesaplama yöntemi olarak entropy kullanılması) ve max\_depth=3 (ağacın maksimum derinliği) bulunmaktadır.

# Model, test veri seti üzerinde değerlendirilmiş ve çeşitli performans metrikleri kullanılarak performansı ölçülmüştür. Bu metrikler arasında Accuracy, Recall, Precision ve F1 skoru bulunmaktadır. Ayrıca, confusion matrisi de değerlendirme sürecinde kullanılmıştır.

# 

# RBF SVM

Amacımız, veri setindeki sınıflandırma görevini en iyi şekilde yerine getiren SVM modelini bulmaktır. Modelin performansı değerlendirildiğinde, elde edilen sonuçlar belirli bir C değeri için Accuracy, Recall, Precision ve F1 skoru açısından en iyi değerlerin bulunduğunu göstermektedir. Ancak, modelin daha fazla geliştirilmesi veya ayarlanması gerekebilir. Özellikle, farklı C değerlerinin yanı sıra diğer parametrelerin etkisi de değerlendirilmelidir. Bu nedenle, modelin daha fazla iyileştirilmesi için farklı parametre ayarları denenebilir veya farklı sınıflandırma algoritmaları kullanılabilir.



# NAIVE BAYES

# Naive Bayes, özelliklerin (feature) sınıflandırma işlemi sırasında birbirlerinden bağımsız oldukları varsayımına dayanan bir sınıflandırma yöntemidir. Bu varsayım gerçek hayatta her zaman doğru olmasa da, birçok durumda iyi sonuçlar verir.

BernoulliNB, özellikle binary (0-1) özelliklere sahip veriler için uygun olan bir Naive Bayes alt türüdür. Bizim örneğimizde, verilerin ölçeklendirilmesi sonrasında binary özelliklere dönüştüğünü varsayabiliriz.

Özelliklerin ölçeklendirilmesi gerçekleştirildi (min-max normalization).

Train-test split yöntemi kullanılarak veri seti eğitim ve test kümelerine ayrıldı. Eğitim seti modelin öğrenmesine, test seti ise modelin performans değerlendirmesine kullanıldı.

BernoulliNB sınıflandırıcı örneği oluşturuldu.

Eğitim verileri modelin öğrenmesi için kullanıldı (fit methodu).

Test verileri kullanılarak sınıf tahminleri yapıldı (predict methodu).

Accuracy , Recall , Precision ve F1 Skoru hesaplandı.



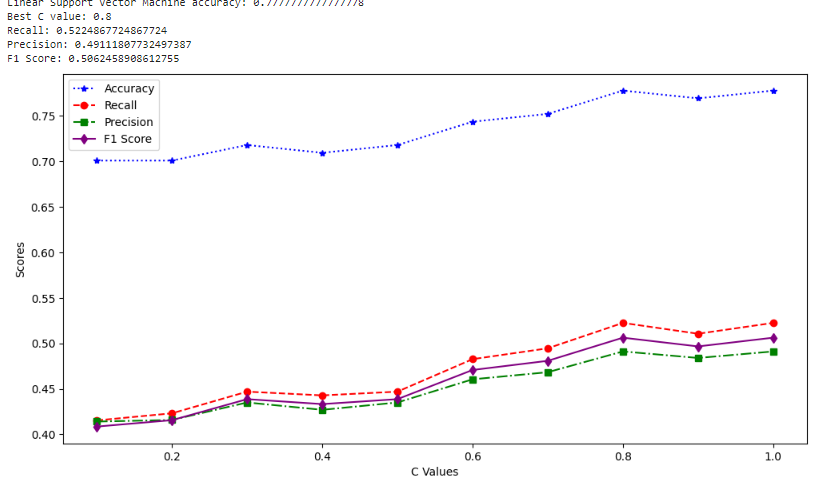
# LSVM

# Özelliklerin ölçeklendirilmesi gerçekleştirildi (min-max normalization).

Train-test split yöntemi kullanılarak veri seti eğitim ve test kümelerine ayrıldı. Eğitim seti modelin öğrenmesine, test seti ise modelin performans değerlendirmesine kullanıldı.

Kod, en iyi C değeri, bu değer ile elde edilen Accuracy , Recall , Precision ve F1 Skoru değerlerini çıktı olarak verir.

Ayrıca, C değerlerine göre performans metriklerinin değişimini gösteren bir grafik çizilir. Bu grafik, modelin karmaşıklığı ile performans arasındaki ilişkiyi görselleştirmeye yardımcı olur.



# RANDOM FOREST

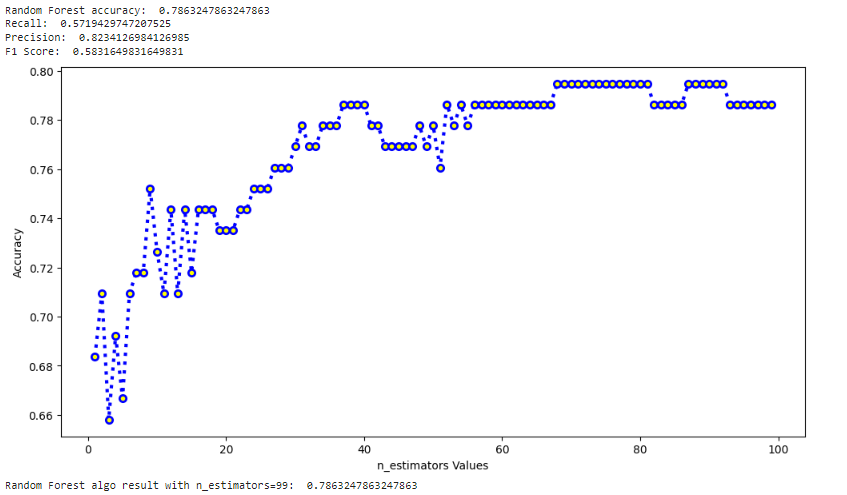
# Random Forest, birden fazla karar ağacının bir araya getirilmesiyle oluşturulur. Her karar ağacı bağımsız olarak eğitilir ve nihai tahmin, tüm karar ağaçlarının tahminlerinin çoğunluğuna göre yapılır. Bu yaklaşım, tek bir karar ağacının aşırı öğrenme (overfitting) problemine karşı daha dayanıklı bir model oluşturur.

Bu çalışmada, Random Forest modelinin performansını değerlendirmek ve en iyi karar ağacı sayısı değerini bulmak amaçlanmıştır.

Seçilen n\_estimators değeri 99 olan bir Random Forest sınıflandırıcı örneği oluşturuldu.

Eğitim verileri modelin öğrenmesi için kullanıldı (fit methodu).

Test verileri kullanılarak sınıf tahminleri yapıldı (predict methodu).

, Accuracy, Recall, Precision ve F1 Score hesaplandı.

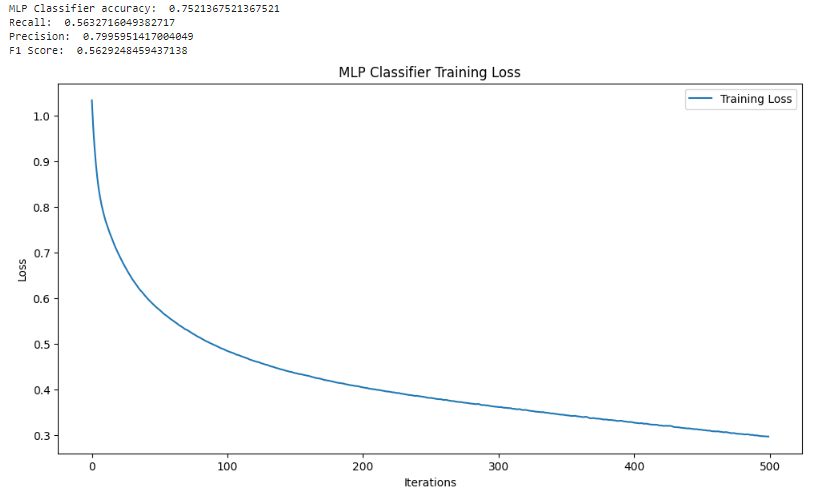
# MLP CLASSIFIER

# MLP, birbirine bağlı katmanlardan oluşan yapay sinir ağı mimarisine sahiptir. Bu katmanlar, giriş katmanı, gizli katmanlar ve çıkış katmanı olmak üzere farklılaşır. Gizli katmanlardaki nöronlar, giriş verilerini dönüştürerek lineer olmayan ilişkileri modellemeyi sağlar.

Bu çalışmada, MLP modelinin performansını değerlendirmek ve sınıflandırma görevindeki başarısını ölçmek amaçlanmıştır.

Özelliklerin ölçeklendirilmesi gerçekleştirildi (min-max normalization).

Train-test split yöntemi kullanılarak veri seti eğitim ve test kümelerine ayrıldı. Eğitim seti modelin öğrenmesine, test seti ise modelin performans değerlendirmesine kullanıldı.

Verilerin standart normalleştirme işlemi ("StandardScaler") uygulandı. Bu işlem, özelliklerin ortalama sıfır ve birim varyans değerine sahip olmasını sağlar.

# LOGISTIC REGRESSION

# Bağımsız değişkenler (x) ve bağımlı değişken (y) ayrıldı.

Eğitim ve test kümeleri oluşturmak için train\_test\_split fonksiyonu kullanıldı.

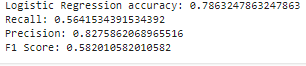
Sınıf etiketleri (y) kategorik olduğundan, LabelEncoder ile sayısal hale getirildi.

Lojistik Regresyon modeli max\_iter=1000 ve random\_state=42 parametreleri ile oluşturuldu.

Eğitim verileri (x\_train, y\_train\_encoded) modelin öğrenmesi için kullanıldı.

Test verileri (x\_test) kullanılarak sınıf tahminleri yapıldı (y\_pred).

Accuracy, Recall, Precision ve F1 Score hesaplandı. Bu sonuçlar, modelin sınıflandırma problemini ne kadar iyi çözdüğünü gösterir.



# NEURAL NETWORKS

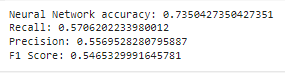
# 00 ve 50 nöronlu iki gizli katmana sahip bir Yapay Sinir Ağı modeli oluşturuldu.

relu' aktivasyon fonksiyonu, 'adam' çözücü ve 1000 iterasyon ile model eğitimi gerçekleştirildi.

Eğitim verileri (x\_train, y\_train\_encoded) modelin öğrenmesi için kullanıldı.

Test verileri (x\_test) kullanılarak sınıf tahminleri yapıldı (y\_pred).

Accuracy , Recall , Precision ve F1 Score hesaplandı.



# ADABOOST

50 adet karar ağacı içeren bir AdaBoost modeli oluşturuldu (n\_estimators=50).

Eğitim verileri (x\_train, y\_train) modelin öğrenmesi için kullanıldı.

Test verileri (x\_test) kullanılarak sınıf tahminleri yapıldı (y\_pred).

Accuracy, Recall , Precision ve F1 Score hesaplandı.

