



**Ders Adı:** BLM4510 Yapay Zeka

**Öğrenci Adı:** 15011025 Muhammet Çeneli

**Proje konusu:** KNN ve LVQ Algoritmaları

Projede seçilen konu makine öğrenmesi algoritmaları olan knn ve lvq algoritmalarının aynı verisetleri üzerinde karşılaştırılmalarıdır. Knn tüm eğitim verisetini belleğe alarak yeni bir veri geldiğinde verinin tümüyle karşılaştırıp en yakın k tanesi içerisindeki en çok bulunan sınıfı seçer. Test verisi gelmeden önce eğitim işlemi olmadığından dolayı unsupervised algoritmadır. Lvq ise öncesinde eğitim işlemi yaparak vektörler oluşturur bu vektörler ağırlıkları temsil eder. Supervised algoritmadır.

### **KNN Algoritmasının Çalışması**

K-en yakın komşuları (KNN) algoritması, yeni veri noktalarının değerlerini tahmin etmek için 'özellik benzerliği' kullanır, bu da yeni veri noktasına, eğitim setindeki noktalara ne kadar yakın olduğuna bağlı olarak bir değer atanacağı anlamına gelir. Çalışmasını aşağıdaki adımlar yardımıyla anlayabiliriz:

**Adım 1** - KNN'nin ilk adımı sırasında, eğitimi ve test verilerini yüklenir.

**Adım 2** - Sonra, K değerini yani en yakın kaç nokta üzerinden hesaplama yapılacağı seçilir.

**Adım 3** - Test verilerindeki her nokta için aşağıdakileri yapılır:

**3.1** - Test verisi ile eğitim verisindeki her elemanın arasındaki mesafeler hesaplanır.

**3.2** - Uzaklık değerine göre, bu veriler artan düzende sıralanır.

**3.3** - Daha sonra, sıralı diziden en üstteki K satırları seçilir.

**3.4** - Bu satırların içerisindeki en çok bulunan sınıf sonuç olur.

### **LVQ Algoritmasının Çalışması**

**Adım 1** - Referans vektörleri başlangıçta aşağıdaki gibi tanımlanır:

**Aşama 1-a**- Verilen eğitim vektörleri setinden, ilk “m” (küme sayısı) eğitim vektörleri alınır ve bunlar ağırlık vektörleri olarak kullanılır. Kalan vektörler eğitim için kullanılabilir.

**Aşama 1-b** - İlk ağırlık ve sınıflandırma rastgele atanır.

**Aşama 1-c** - K-means yöntemi uygulanır.

**Adım 2** - Referans vektörü başlatılır.

**Adım 3** - Bu algoritmayı durdurma koşulu karşılanmazsa, adım 4-9 ile devam edilir.

**Adım 4** - Her eğitim vektörü için 5-6. adımlar uygulanır.

**Adım 5 -  $J = 1-m$  ve  $i = 1-n$  için mesafe hesaplanır.**

**Adım 6 - Uzaklığının minimum olduğu birim alınır.**

**Adım 7 - Yeni ağırlıklar hesaplanır.**

**Adım 8 - Öğrenme oranını azaltılır.**

**Adım 9 - Çıkış durumu kontrol edilir.**

Geliştirme Sürecinde KNN'in bir dezavantajının, tüm eğitim veri setine bağlı kalınması gerektiği olarak gözlemlenmiştir. KNN'nin veri kümenizdeki iyi sonuçlar verdiğini fark edilmiştir fakat tüm eğitim veri kümesini saklamak dezavantajdır. Bellek gereksinimlerinin azalması için LVQ kullanılabilir.

### Ekran Görüntüleri:

```
Administrator: C:\Windows\System32\cmd.exe
balance-scale dataset is calculating...
  success rate with KNN = 0.80
  success rate with LVQ = 0.86

iris dataset is calculating...
  success rate with KNN = 0.93
  success rate with LVQ = 0.91

abalone dataset is calculating...
  success rate with KNN = 0.50
  success rate with LVQ = 0.37

C:\Users\pc\Desktop\yapay zeka final>
```

```
Select Administrator: C:\Windows\System32\cmd.exe
balance-scale dataset is calculating...
  success rate with KNN = 0.86
  success rate with LVQ = 0.87

iris dataset is calculating...
  success rate with KNN = 1.00
  success rate with LVQ = 0.91

abalone dataset is calculating...
  success rate with KNN = 0.49
  success rate with LVQ = 0.37

C:\Users\pc\Desktop\yapay zeka final>
```

```
Select Administrator: C:\Windows\System32\cmd.exe
balance-scale dataset is calculating...
    success rate with KNN = 0.86
    success rate with LVQ = 0.86

iris dataset is calculating...
    success rate with KNN = 1.00
    success rate with LVQ = 0.93

abalone dataset is calculating...
    success rate with KNN = 0.49
    success rate with LVQ = 0.31

C:\Users\pc\Desktop\yapay zeka final>
```

Kaynaklar:

[https://en.wikipedia.org/wiki/Learning\\_vector\\_quantization](https://en.wikipedia.org/wiki/Learning_vector_quantization)

[https://www.tutorialspoint.com/artificial\\_neural\\_network/artificial\\_neural\\_network\\_learning\\_vector\\_quantization.htm](https://www.tutorialspoint.com/artificial_neural_network/artificial_neural_network_learning_vector_quantization.htm)

[https://www.tutorialspoint.com/machine\\_learning\\_with\\_python/machine\\_learning\\_with\\_python\\_knn\\_algorithm\\_finding\\_nearest\\_neighbors.htm](https://www.tutorialspoint.com/machine_learning_with_python/machine_learning_with_python_knn_algorithm_finding_nearest_neighbors.htm)

<https://archive.ics.uci.edu/ml/datasets.php>