İnönü Üniversitesi Sağlık Hizmetleri Meslek Yüksekokulu Dergisi
ISSN: 2147-7892, Cilt 10, Sayı 1 (2022) 356-371
doi: 10.33715/inonusaglik.1022065

Journal of Inonu University Health Services Vocational School
ISSN: 2147-7892, Volume 10, Issue 1 (2022) 356-371
doi: 10.33715/inonusaglik.1022065

Original Article / Araştırma Makalesi

AZ VERİ SETLİ ÇALIŞMALARINDA DERİN ÖĞRENME VE DİĞER SINIFLANDIRMA ALGORİTMALARININ KARŞILAŞTIRILMASI: AGONİST VE ANTAGONİST LİGAND ÖRNEĞİ

Comparison of Deep Learning and Other Classification Algorithms in Small Dataset Studies: Example of Agonist and Antagonist Ligand

Fatih Mehmet AVCU[1]
[1]*İnönü Üniversitesi, Enformatik Bölümü, Malatya*

Geliş Tarihi / Received: *11.11.2021* ***Kabul Tarihi / Accepted:*** *17.01.2022*

ÖZ

Makine öğrenme algoritmaları günümüzde hemen hemen tüm bilim dallarında kullanılmaktadır. Özellikle sınıflandırma algoritmaları fen ve sağlık bilimleri açısından oldukça popüler bir konudur. Derin öğrenme, diğer algoritmalar gibi makina öğrenme tekniklerinden biridir. Günümüzde işlemci hızlarının artması nedeni ile tekrar popüler olmuştur. Özellikle grafik işlemci tabanlı hesaplamalar bu konuyu popüler yapmıştır. Bu çalışmanın amacı, kimyasal veri tabanlarından elde edilen veriler ile literatürde iyi bilinen, dopamin reseptörlerine bağlanan agonist ve antiagonist moleküllerini makine öğrenme algoritmaları ile sınıflandırmaktır. Çalışmanın amacı ayrıca veri sayısı az olan durumlarda sınıflandırma yaparken doğru bir sınıflandırma için derin öğrenme algoritmasının kullanımını önermektir. Algoritmanın eğitmek için, Python kütüphanelerinden Scikit-learn ve Tensorflow-Keras kullanılmıştır. Sınıflandırma işlemi popüler makine öğrenme algoritmaları ile kıyaslanmış ve sonuçlar bir tablo olarak sunulmuştur.

Anahtar kelimeler: Agonist-antagonist, Derin öğrenme, Ligant, Makine öğrenmesi, Sınıflandırma.

ABSTRACT

Machine learning algorithms are used in almost all branches of science today. In particular, classification algorithms are a very popular subject in terms of science and health sciences. Deep learning is one of the machine learning techniques like other algorithms Today, it has become popular again due to the increase in processor speeds. Particularly graphics processor-based calculations have made this subject popular. The aim of this study is to classify the agonist and antiagonist molecules that bind to dopamine receptors, which are well known in the literature, with the data we obtained from chemistry databases, with machine learning algorithms. The aim of the study is also to suggest the use of a deep learning algorithm for an accurate classification when classifying in cases where the number of data is small. Scikit-learn and Tensorflow-Keras from Python libraries were used for training the algorithm. The classification process has been compared with popular machine learning algorithms and the results have been presented as a table.

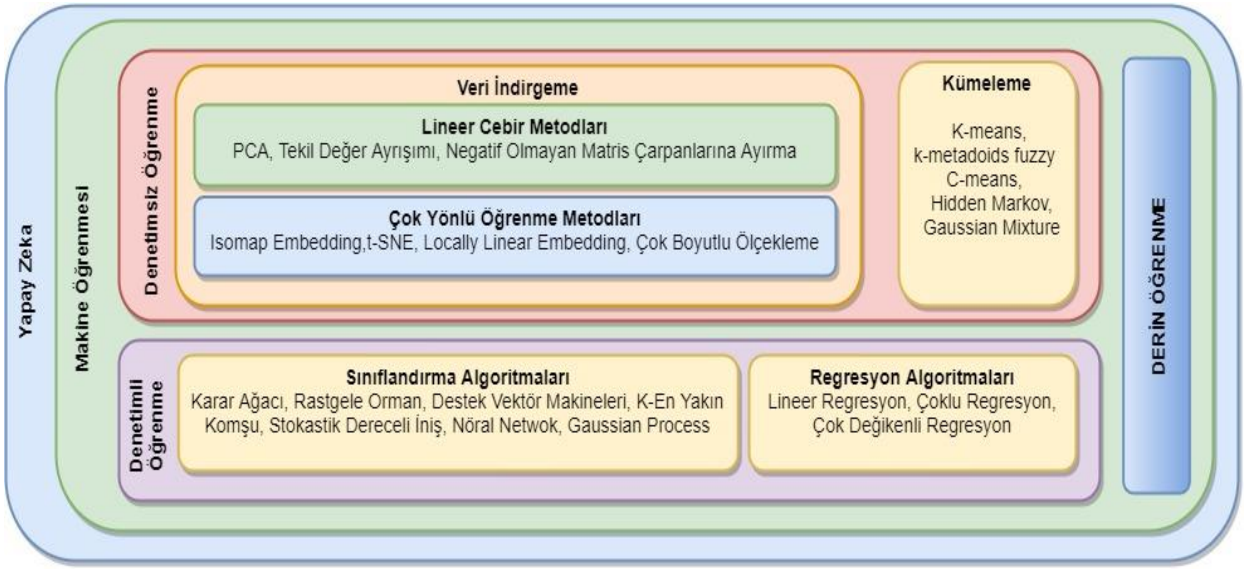
Keywords: Agonist-antagonist, Classification, Deep learning, Ligand, Machine learning.

Fatih Mehmet AVCU✉, fatih.avcu@inonu.edu.tr
İnönü Üniversitesi, Enformatik Bölümü, Malatya

Bu makaleye atıf yapmak için(How to cite this article): Avcu, F. M. (2022). Az Veri Setli Çalışmalarında Derin Öğrenme Ve Diğer Sınıflandırma Algoritmalarının Karşılaştırılması: Agonist Ve Antagonist Ligand Örneği. İnönü Üniversitesi Sağlık Hizmetleri Meslek Yüksekokulu Dergisi, 10(1), 356-371. doi: 10.33715/inonusaglik.1022065

GİRİŞ

Makine öğrenmesi bilgisayar bilimleri başta olmak üzere diğer bilim dalları için de çok hızlı büyüyen bir uygulamadır. Bilgisayarlar, yazılımlar aracılığı ile veri toplayarak öğrenebilir ve karar verme performanslarını geliştirebilirler (Ding, Tong, Zhang, ve Yang, 2008; Drouhard, Sabourin, ve Godbout, 1996). Makine öğrenimi (ML), Yapay Zeka'nın (AI) bir parçasıdır (Şekil 1) amacı, matematik ve istatistik metotlarını kullanarak model oluşturmaktır. Model oluşturmak için giriş verisini temel alan bilgisayar, yeni gelmiş veriye uyan modeli uygulayıp karar verilmesine olanak sağlar. Birçok alanda ve çeşitli konularda makine öğrenimi yöntemleri önerilmekte ve kullanılmaktadır. Son yıllarda, farklı alanlarda çalışan araştırmacılar, verileri kümelemek, sınıflandırmak ve tahmin etmek için makine öğrenimi yaklaşımını kullanmaya yönelmişlerdir (Karakaplan ve Avcu, 2021; Kumar ve Singh, 2018; Sekeroglu, 2004).



Şekil 1. Yapay Zekanın Şematik Olarak Gösterilmesi

Sınıflandırma problemlerinde denetimsiz öğrenme yöntemlerini de kullanılmaktadır. Ancak denetimli yöntemlerde girdi verisi ile çıktı arasında ilişki kurulması nedeniyle sınıflandırma problemlerinde daha popülerlerdir. Denetimli öğrenmede girdi verisi ile verilen doğru çıktı arasında bir ilişki kurulmaktadır. Sıklıkla kullanılan denetimli öğrenme teknikleri Lojistik Regresyon, Naive Bayes, Lineer Regresyon, Karar Ağacı, k-En Yakın Komşu algoritması (kNN), Destek Vektör Makineleri (SVM) ve çeşitli Yapay Sinir Ağları (ANN) algoritmaları olarak bilinir. Bu algoritmalar sonraki bölümde kısaca açıklanmış ve nasıl çalıştıkları şematik olarak gösterilmiştir.

Makine öğrenmesi tekniklerinin kullanıldığı birçok alan vardır. Çoğu uygulama, hesaplamalı öğrenme, doğal dil işleme ve görüntü tanıma üzerinedir. Makine öğrenimi teknikleri, bilgisayarları programlama ile karmaşık gerçek hayat problemlerini çözebilecek hale getirmek için kullanılır. Sistemler, verilerden öğrenebilen algoritmalar ile oluşturulabilir ancak modelin düzgün çalışabilmesi için veri sayısının yeterli olması gerekmektedir. Aksi halde model yanlış kurularak özellikle sınıflandırma için tahminler yanlış olabilmektedir.

Derin öğrenme yine bir makine öğrenme tekniği olmasına karşın veri sayısı az olan çalışmalarda araştırmacılara yardımcı olmaktadır.

İlişkili Çalışmalar

Sean ve arkadaşları makine öğrenme tekniklerini östrojen reseptörünün bağlanma tahmininde kullanmışlar (Russo, Zorn, Clark, Zhu, ve Ekins, 2018). Bu çalışmada, Random forest algoritması östrojen hormonu etkinliğinin tahmininde yeterli olduğunu söylemişlerdir. Richard Judson ve arkadaşları kimyasal tokstite için makine öğrenme yöntemlerini kullanmışlar. Başta Destek Vektör Makineleri ve Yapay Sinir Ağları başta olmak üzere makine öğrenme algoritmalarının uygulamada iyi aday olduğunu yazmışlardır (Judson, Elloumi, Setzer, Li, ve Shah, 2008). S. De Vito ve arkadaşları çok sensörlü kimyasal cihazlarının kalibre edilmesi için Destek Vektör Makineleri algoritmasının tutarlı sonuçlar verdiğini belirtmişlerdir (De Vito vd., 2018). Eni Mineraller ve arkadaşları ilaca bağlı karaciğer hasarlarını öngörmek için makine öğrenme yöntemlerini kullanmışlar ve en iyi performansı Bayers modelinin gösterdiğini belirtmişlerdir (Minerali, Foil, Zorn, Lane, ve Ekins, 2020). Andreas Mayr (Mayr vd., 2018) ve arkadaşları ilaç hedefi tahmini için makine öğrenme tekniklerini kıyaslamışlar, bu çalışmada rakiplerine oranla derin öğrenmenin önemli bir performans gösterdiğini belirtmişlerdir. Jerry L.Atwood ve diğerleri metal-organik nanokapsüllerin kristalleşme eğilimini incelemek için, hem başarılı hem de başarısız bir dizi deneyden veri setlerini eğiterek makine öğrenme algoritmalarını test etmişler ve XGBoost Algoritmasının yüksek doğrulukta tahmin sağladığını raporlamışlardır (Xie vd., 2020). Garrett B.Goh ve arkadaşları derin öğrenmenin özellikle hesaplamalı kimya alanında diğer makine öğrenme algoritmalarından üstün olduğunu belirtmişlerdir (Goh, Hodas, ve Vishnu, 2017). Aguiar ve arkadaşları kristalografik tahmin için kırınım ve kimya verileri kullanarak makine öğrenme algoritmalarını kullanmışlar ve derin öğrenme algoritmasının sınıflandırmada %85'in üzerinde doğruluğa sahip olduğunu söylemişlerdir (Aguiar, Gong, ve Tasdizen, 2020). Z. Pinar Gumus ve arkadaşları zeytinyağının coğrafi kökenini belirlemek için farklı

sınıflandırma yöntemlerini denemişler ve random forest algoritmasının iyi sonuç verdiğini ortaya koymuşlardır (O. Gumus, Yasar, Z. P. Gumus, ve Ertas, 2020).

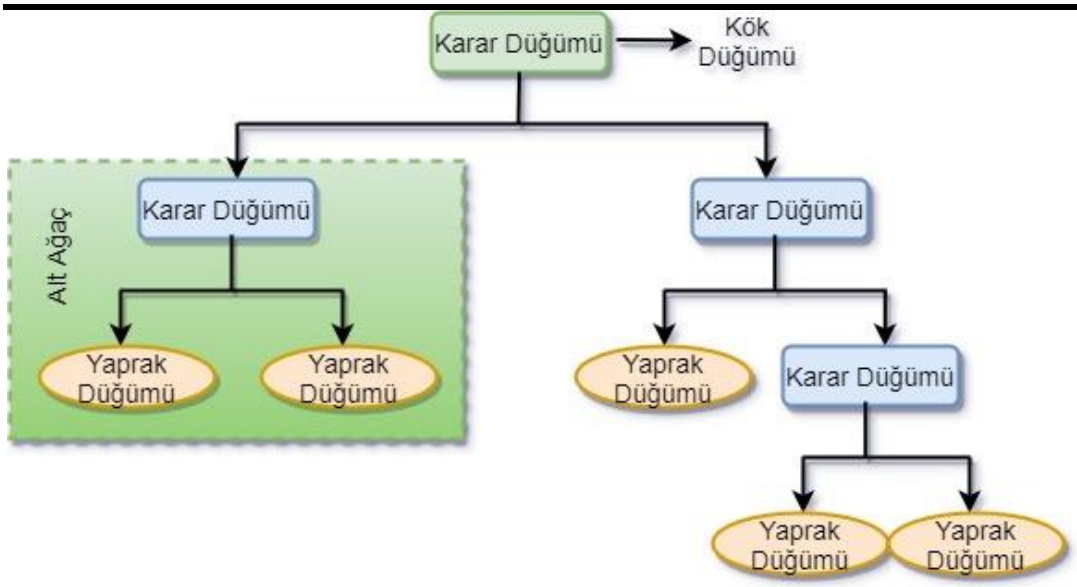
Makine Öğrenmesi

Makine öğrenmesi, bir problemi o probleme ait veriye göre modelleyen, matematiksel ve istatistiksel çıkarımlar yaparak tahminde bulunan bilgisayar algoritmalarının genel adıdır. Karar Ağacı, Random Forest, SVM, K-En Yakın Komşu, SGD, Naive Bayes, Gaussian Process Algoritması gibi pek çok makine öğrenme algoritması geliştirilmiştir. Bu algoritmalar tahmin ve kestirim, kümeleme ve sınıflandırma problemlerinde sıklıkla kullanılırlar.

Karar Ağacı Algoritması

Bir denetimli öğrenme tekniği olan Karar Ağacı Algoritması, hem sınıflandırma (Friedl ve Brodley, 1997) hem de regresyon (Tso ve Yau, 2007) problemleri için tercih edilen bir yöntemdir. Dalların karar düğümlerini yaprakların ise sonuç düğümlerini temsil ettiği bu algoritma şekil 2'de gösterilmiştir.

Bir Karar ağacında, Karar Düğümü ve Yaprak Düğümü olmak üzere iki düğüm vardır. Karar düğümleri herhangi bir karar vermek için kullanılır ve birden fazla sonucu olabilir. Yaprak düğümleri ise karar düğümlerinin sonucudur. Kararlar verilen veri kümesinin özellikleri temelinde gerçekleştirilir. Verilen koşullara dayalı olarak bir soruna/karara ilişkin tüm olası çözümleri elde etmek için gerekli kadar karar düğümü eklenir. İşlemler Kök Düğümü ile başlayıp karar düğümleri ile devam eder ve sonunda yaprak düğümlerine ulaşılır. Elde edilen yapı bir ağaca benzediğinden bu algoritmaya karar ağacı algoritması denir. Ağacı oluşturan değişken sayısına göre tek değişkenli (ID3 ve C4.5) ya da çok değişkenli (CART) karar ağacı algoritmaları kullanılır. Bir karar ağacı düğüm noktasında soru sorar ve cevaba göre ağacı alt dallara böler.

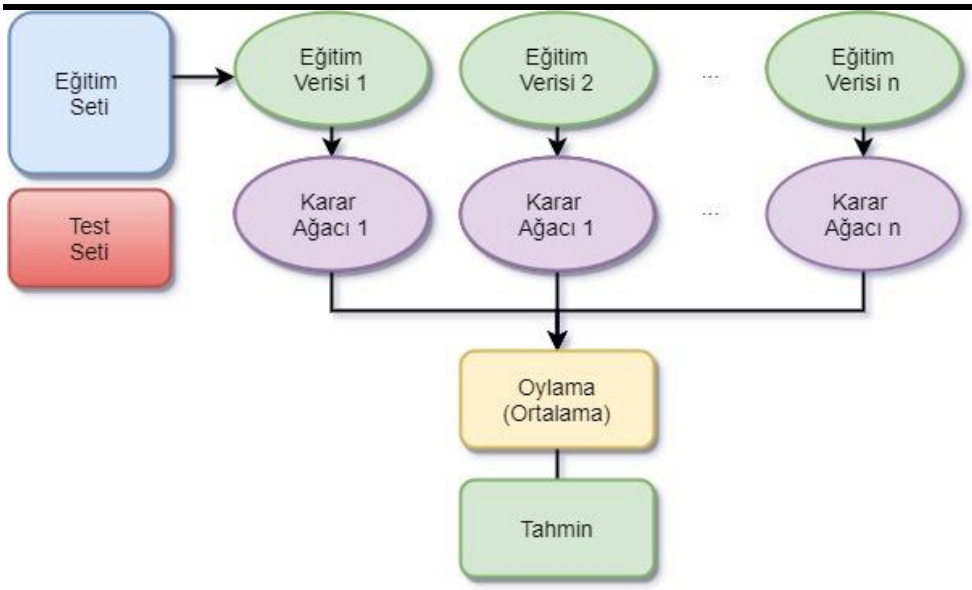


Şekil 2. Karar Ağacı Algoritması

Rastgele Orman (Random Forest) Algoritması

Rastgele orman, denetimli öğrenme tekniğine ait popüler bir makine öğrenme algoritmasıdır. Karar ağacı algoritması gibi sınıflandırma (Grömping, 2009) hem de regresyon (Pal, 2005) problemleri için kullanılabilir.

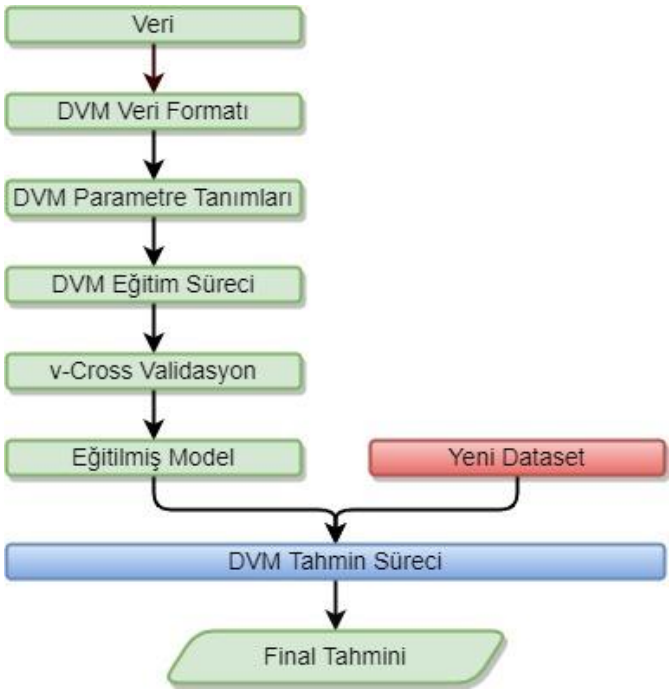
Adından da anlaşılacağı gibi, Random Forest, verilen veri kümesinin çeşitli alt kümelerinde bir dizi karar ağacı içeren ve bu veri kümesinin tahmin doğruluğunu iyileştirmek için ortalamayı alan bir sınıflandırıcıdır. Bu algoritma, tek bir karar ağacına güvenmek yerine, her ağaçtan tahmini alır ve tahminlerin çoğunluk oylarına dayanarak nihai çıktıyı tahmin eder. Karmaşık bir problemi çözmek ve modelin performansını iyileştirmek için birden fazla karar ağacının birleşmesiyle oluşturulur. Algoritmanın şeması Şekil 3'de gösterilmiştir. Veri kümesinin sınıfını tahmin etmek için birden çok ağacı birleştirildiğinden, bazı karar ağaçlarının doğru çıktıyı tahmin etmesi, bazılarının ise tahmin etmemesi mümkündür. Ancak birlikte, tüm ağaçlar doğru çıktıyı tahmin eder.



Şekil 3. Rastgele Orman Algoritması

Destek Vektör Makineleri Algoritması

V. Vladimir tarafından 1995 yılında geliştirilmiştir (Cortes ve Vapnik, 1995). Denetimli öğrenme modellerinden biri olan destek vektör makineleri (DVM), sınıflandırma ve regresyon analizi için kullanılır. Ancak literatürde çoğunlukla sınıflandırma algoritmalarına uygulandığı görülür (Furey vd., 2000). Algoritma kısaca, bir düzlem üzerine yerleştirilmiş eğitim verisi üzerinde noktaları ayırmak için bir doğru çizer ve bu doğrunun ayrılması düşünülen iki sınıfın noktaları için maksimum uzaklıkta olması planlanır. Algoritmanın akış şeması Şekil 4'de gösterilmiştir.



Şekil 4. DVM Algoritması

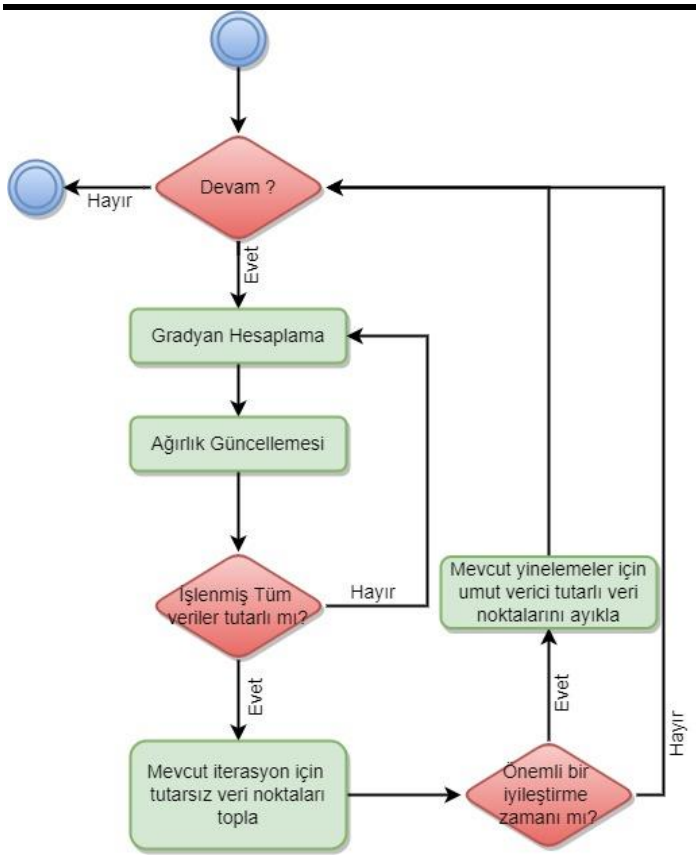
K-En Yakın Komşu Algoritması

1951 yılında Fix ve Hodges tarafından geliştirilmiş, benzerlik ve uzaklık özelliklerine dayanarak sınıflandırma yapmak için oluşturulmuş basit makine öğrenme tekniklerinden biridir (Altman, 1992). K-En Yakın Komşu yöntemi, sınıflandırma problemini çözen denetimli öğrenme yöntemleri arasında yer alır. Algoritmada gelen veri k komşu sayısına göre uzaklıklar belirlenir ve yeni verinin sınıfına karar verilir. Sınıf belirlenmesinde Öklid, Simple Matching, Jaccard ve Manhattan uzaklıkları kullanılabilir. KNN algoritmasının akış şeması Şekil 5' de verilmiştir.

Şekil 5. K-En Yakın Komşu Algoritması

Stokastik Dereceli İniş Algoritması

"Stokastik" kelimesi, rastgele bir olasılıkla bağlantılı bir sistem veya süreç anlamına gelir. Bu nedenle, Stokastik Dereceli İnişte (Stochastic gradient descent-SGD), her bir yineleme için tüm veri kümesi yerine rastgele birkaç örnek seçilir (Taddy, 2019). Dereceli İniş optimizasyonunda, toplu işlem tüm veri kümesi olarak alınır. Bununla birlikte, tüm veri kümesini kullanmak, en iyiye daha az gürültülü veya daha az rastgele bir şekilde ulaşmak için yararlıdır. Ancak veri kümelerimiz gerçekten büyük olduğunda bu bir sorundur. Veri kümemizde bir milyon örneğiniz olduğunu varsayalım. Tipik bir Dereceli İniş optimizasyon tekniği kullanıyorsanız, bu bir milyon verinin tamamını kullanmanız gerekir. Bu nedenle, hesaplaması çok zaman alıcı hale gelir. Böyle problem Stokastik Dereceli İniş ile çözülmektedir (Taddy, 2019). SGD'de, her bir yinelemeyi gerçekleştirmek için sadece tek bir örnek kullanır. Örnek rastgele karıştırılır ve yinelemenin gerçekleştirilmesi için seçilir. Algoritma Şekil 6'da gösterilmiştir.

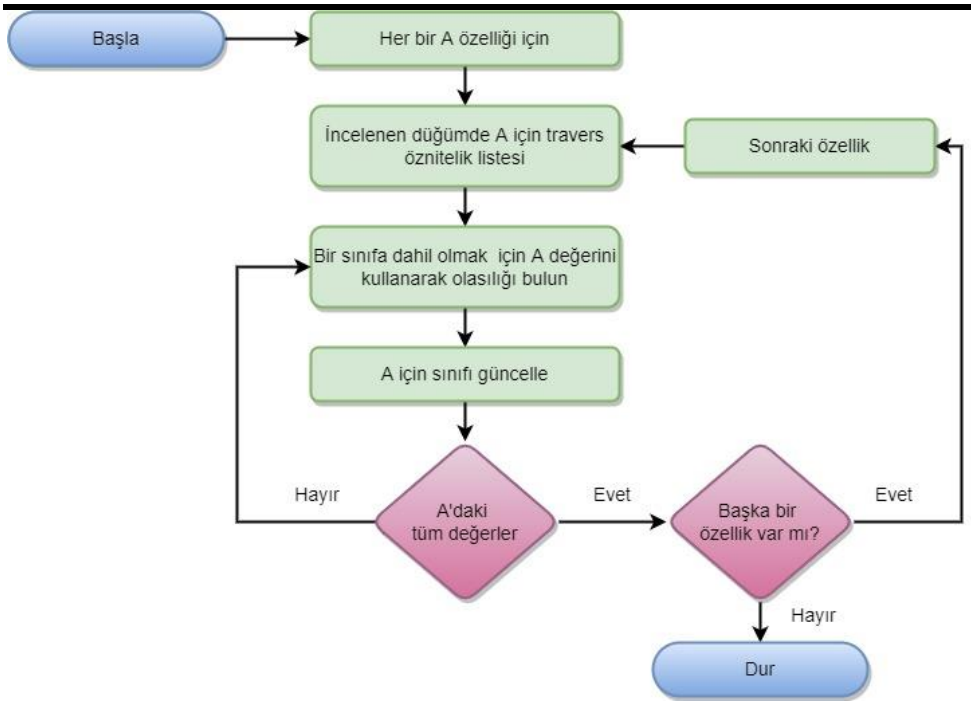


Şekil 6. SGD Algoritması

Naive Bayes Algoritması

Temeli Matematikçi Thomas Bayes'in koşullu olasılık hesaplama formülüne dayanır. Diğer algoritmalar gibi bu algoritmada sınıflandırma örneklerinde sıklıkla kullanılır. Algoritma geçmiş bilgilerin hangi sınıflara ait olduğu verildiğinde yeni gelen verinin hangi sınıfa dâhil olduğunun bulunmasında kullanılır. Kullanımı kolay olması nedeni ile sıklıkla kullanılan bir algoritmadır.

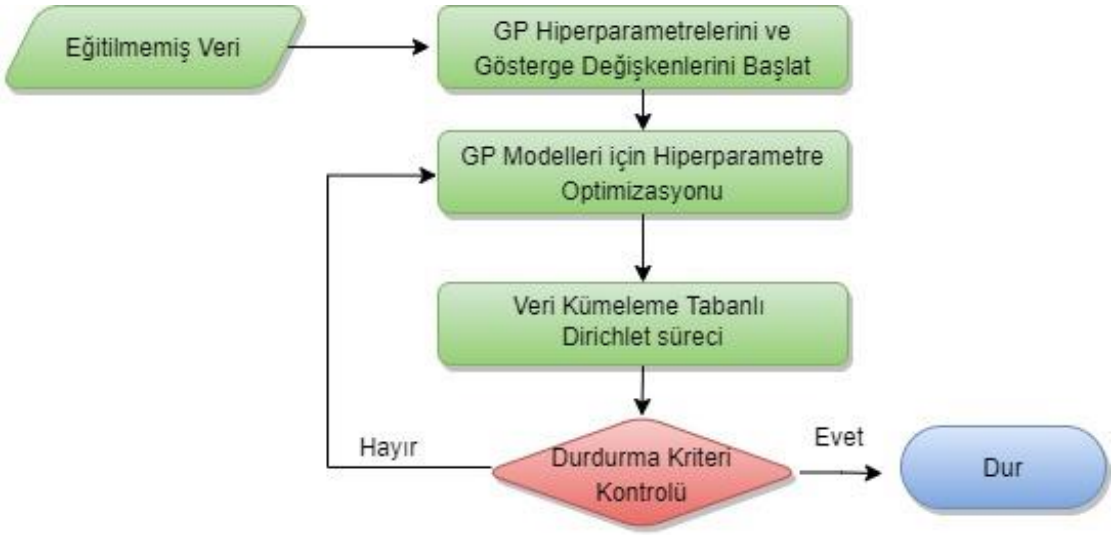
Bu makine öğrenme tekniğinde her bir değişken birbirinden bağımsız aynı zamanda eşit öneme sahip olarak varsayıldığı için daha kolay anlaşılan ve hızlı sonuç veren bir yapıya sahiptir (Maron, 1961). Naive Bayes algoritması, sınıf koşul bağımsızlığı varsayımına dayandığından dolayı, gerekli hesaplama işlemleri etkin ve kolay bir biçimde gerçekleştirilebilmektedir. Algoritmanın akış şeması Şekil 7'de gösterilmiştir.



Şekil 7. Naive Bayes Algoritması

Gaussian Process Algoritması

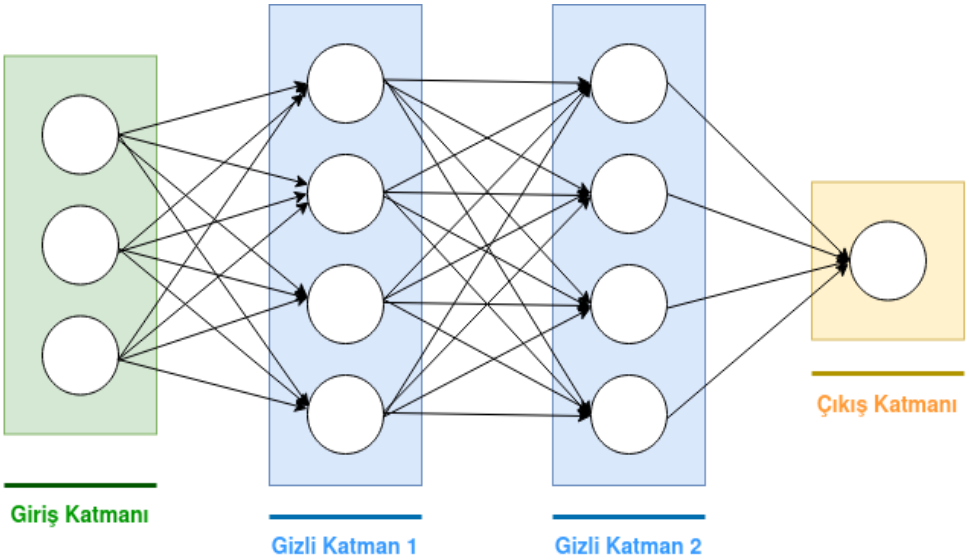
Adını Carl Friedrich Gauss' den alan Gauss Süreçleri, Bayes metodolojisini temel alan, parametrik olmayan bir sınıflandırma algoritmasıdır (Leen, Dietterich, ve Tresp, 2001). Gauss olasılık dağılımına dayanır ve sınıflandırma problemlerinde sıklıkla kullanılır. Bir Gauss süreci stokastik bir süreçtir. Algoritma, eğitim verilerinden görünmeyen bir noktanın değerini tahmin etmek için öğrenme ve noktalar arasındaki benzerliğin bir ölçüsünü kullanır. Tahmin sadece bu nokta için bir tahmin değil, aynı zamanda belirsizlik bilgisine de sahiptir tek boyutlu bir Gauss dağılımıdır (Şekil 8).



Şekil 8. Gauss Süreçleri Algoritması

Derin Öğrenme (DNN)

Derin öğrenme algoritması Şekil 9'da yapısı gösterilmiş bir makine öğrenmesidir. Bir önceki katmandan elde ettiği çıktıyı girdi verisi olarak kullanan derin öğrenme, özellik çıkarma ve dönüştürme için birçok doğrusal olmayan işlem birimi katmanını kullanır (Deng ve Lu, 2014). 1990-2000 yıllarında yapay sinir ağlarının hesaplama maliyetinden dolayı çalışmalar destek vektör modeli üzerinde olmuştur (Cortes ve Vapnik, 1995). Geçen 10 yıllık sürede özellikler GPU programlama ile işlem hızı 1000 kat artması ile yapay sinir ağları destek vektör algoritmalarına rakip olmaya başlamıştır (Schmidhuber, 2015).



Şekil 9. Derin Öğrenme Algoritması

GEREÇ VE YÖNTEM

Veri Setinin Hazırlanması

G-proteinine bağlı reseptörler (GPCR'ler), farmasötik geliştirmede ilaç hedeflerinin en önemli ailelerinden birini temsil ederler. GLIDA, öncelikle GPCR'ler ve ligandları arasındaki bilgilerin entegrasyonuna odaklanan, GPCR ile ilgili bir Kimyasal Genomik veri tabanıdır. Tablo 1'de görülen ligandlar, GLIDA veri tabanından elde edilmiş iyi bilinen ve çalışılan ligandlardır.

Tablo 1: Kimyasal Veri Tabanlarına göre Ligandların Sınıflandırılması

Ligand	Agonist	Antagonist	C1	C2	C3
Apomorphine	1	16	00	11	01
Bromperidol	-	4	01	01	01
Clozapine	6	49	00	11	11
Haloperidol	1	31	11	11	01
Loxapine	-	30	01	01	01
Metergoline	-	24	00	01	01

Moperone	-	4	11	01	01
Olanzapine	5	37	10	11	11
Pimozide	-	20	11	01	01
Pramipexole	3	4	11	11	11
Raclopride	-	9	10	01	01
Risperidone	-	36	10	01	01
Sertindole	-	28	10	01	01
Trazodone	-	10	10	01	01
Xanomeline	12	3	11	11	11
Ziprasidone	4	9	10	11	11

Ligandlar agonist, antagonist veya her iki özelliği de göstermektedir. Bu özelliklere dayanılarak ligandlar C1, C2, C3 şeklinde gruplandırılmış ve gruplar binary olarak kodlanmıştır. Burada C1 gruplaması molekülün yapısına bağlı bulunduğu kimyasal gruplamadır. 00; Bütirofenonlar, 01; Benziazoksil piperidinler, 10; Polisiklik aromatikler ve 11; ise diğer bileşikler, olarak gruplamaya alınmıştır. C1 gruplaması kontrol amaçlıdır. Burada verilen agonist ve antagonist tam sayıları (C2, C3 sayıları) GLIDA veri tabanındaki raporlama sayılarıdır. C2 de zayıf olan agonist veya antagonist dikkate alınmadan yapılan gruplamadır. C3 de raporlama sayıları birbirine yakın ise her iki özelliği gösterecek şekilde gruplama yapılmıştır.

Tablo 2 'deki verilerden LJE ve CouIE Gromacs uygulaması ile hesaplanmış, BindE, NCls ve Ki değerleri AutoDock4 uygulaması ile yapılan docking hesaplamalarına ait değerlerdir. Homo-Lumo değerleri NWChem (Valiev vd., 2010) ile hesaplanmıştır. Ligandların geometrileri MP2/6-31G** baz seti ile optimize edilmiştir. Son olarak HwA, RotB, HBA, XLogP, Cpx, SfA ve MWt ise PubChem veri tabanından PyChem (PyChem, 2021) kütüphanesi aracılığı ile alınmıştır. Geometri optimizasyonu zaman alan bir çalışma olduğundan hesaplamaların bir kısmı TÜBİTAK ULAKBİM Yüksek Başarımlı ve Grid Hesaplama Merkezinde bulunan TRUBA (Türk Ulusal Bilim e-Altyapısı) sistemi üzerinde yapılmıştır.

Tablo 2: Kimyasal Veri Tabanından Ligandların Özellik Değerleri, Docking ve MD Değerleri

Ligand	**LJE**	**CoulE**	**BindE**	**NCls**	**Ki**	**Homo**	**Lumo**	**HwA**	**RotB**	**HBA**	**XlogP**	**Cpx**	**SfA**	**MWt**
Apomorphine	-131.2	-34.9	-8.9	55	315	-0.2	-0.17	20	0	3	2.3	374	43.7	267.3
Bromperidol	-170.3	-29.8	-9.8	37	67.9	-0.22	-0.17	26	6	4	3.3	451	40.5	420.3
Clozapine	-142.7	-34.6	-9.7	97	79	-0.17	-0.09	23	1	4	3.2	584	30.9	326.8
Haloperidol	-157.2	-34.6	-9.7	5	80.7	-0.22	-0.17	26	6	4	3.2	451	40.5	375.9
Loxapine	-128.3	-12.6	-8.4	10	665.8	-0.2	-0.15	23	1	3	3.1	450	28.1	327.8
Metergoline	-176.5	-48.1	-10.7	5	13.9	-0.18	-0.18	30	5	3	3.8	607	46.5	403.5
Moperone	-154	-26.6	-9.4	38	129.6	-0.22	-0.16	26	6	4	3	445	40.5	355.5
Olanzapine	-154	-43.7	-8.6	51	534.5	-0.16	-0.09	22	1	5	2.8	543	56.2	312.4

Pimozide	-196.4	-42.9	-11.3	27	4.8	-0.2	-0.19	34	7	4	6.3	632	35.6	461.6
Pramipexole	-81.8	-49.6	-6.8	57	10340	-0.18	-0.2	14	3	4	1.9	188	79.2	211.3
Raclopride	-141.3	-23.2	-8	15	1470	-0.22	-0.2	22	5	4	2.9	386	61.8	347.2
Risperidone	-169.8	-28.5	-11.4	11	4.1	-0.21	-0.16	30	4	6	2.7	731	61.9	410.5
Sertindole	-162.5	-35.6	-11.3	17	4.8	-0.2	-0.17	31	5	3	4.1	623	40.5	440.9
Trazodone	-163.4	-36.9	-9.6	51	85.8	-0.2	-0.15	26	5	4	2.8	611	42.4	371.9
Xanomeline	-136.6	-13.4	-7.5	40	3170	-0.21	-0.16	19	7	5	3.3	298	66.5	281.4
Ziprasidone	-180.1	-37.5	-10.5	8	19.7	-0.2	-0.16	28	4	5	4	573	76.7	412.9

Tablo 2'deki kısaltmalar, LJE: Lennard-Jones Potansyeli (kJ/mol), CoulE: Coulomb Potansiyeli (kJ/mol), BindE: Ligand Bağlanma Enerjisi (kJ/mol), Ncls: Kümedeki sayı, Ki: İnhibisyon sabiti (μM), Homo: En Yüksek İşgal Edilen Moleküler Yörünge Enerjisi, Lumo: Düşük Boş Yörünge Enerjisi, HwA: Ağır Atom Sayısı, RotB: Dönel Bağ Sayısı, HBA:Hidrojen Bağ Alıcı Sayısı, Cpx: Karmaşıklık, SfA:Topolojik Polar Yüzey Alanı, MWt: Moleküler Ağırlık şeklindedir.

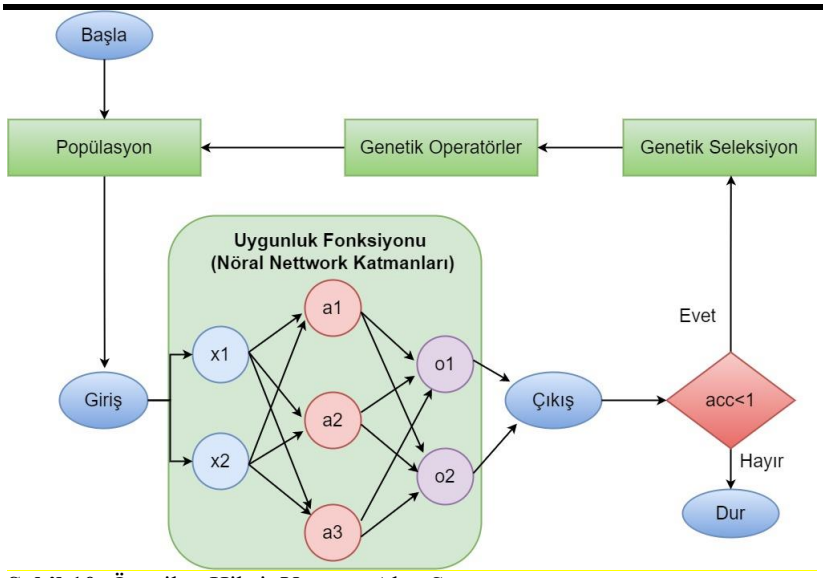
Yazılım ve Donanım

Algoritmamız Intel i8700 işlemci, 8Gb RAM ve RTX2060 GPU ile Python 3 (Python, 2021) betik dili ile geliştirilmiş olup, Tensorflow-Keras (Tensorflow, 2021) ve Sckit-learn (Scikit-learn, 2021) makine öğrenme frameworkları (çerçeveleri) kullanılmıştır. Kıyaslamanın anlamlı olması için algoritmalar ön tanımlı parametreler ile çalıştırılmıştır.

Hibrit Yapının Oluşturulması

Önceki çalışmalarımızda hibrit sistemin nasıl oluşturulduğu ve evrimsel hesaplamanın nasıl yapıldığı ile evrimsel hesaplama hyper parametrelerinin nasıl seçildiği detaylı olarak anlatılmıştır (Karakaplan ve Avcu, 2013; Karakaplan ve Avcu, 2021).

Bu çalışmada, Genetik algoritma işlemleri için PyEvolve çerçevesini kullandık. PyEvolve Python üzerinde çalışan açık kaynak kodlu bir yapıdır. PyEvolve genetik oparatörler için hazır parametreler sunmaktadır. Şekil 10'da algoritmanın akış şemasında görüleceği gibi, hibrit yapıda DNN'in giriş katmanı verileri genetik algoritma tarafından seçilmiştir.



Şekil 10. Önerilen Hibrit Yapının Akış Şeması

Algoritmaların Öğrenme İşlemleri ve Hyper Parametreler

Scikit-learn kütüphanesi aracılığı ile kullandığımız makine öğrenme fonksiyonları Tablo 3'de gösterilmiştir. Karar ağacı algoritmasında sınıflandırma fonksiyonu olan *DecisionTreeClassifier* fonksiyonunun *max_depth* parametresi, ağacın fazla dallanmaması için 5 olarak belirlenmiştir. Bu işlem ağacı budama olarak bilinir ve işlem süresi üzerinde etkisi vardır. Random Forest algoritmasında *max_depth* parametresi karar ağacında olduğu gibi 5 olarak değerlendirilmiştir. Ayrıca, tahminler için ormanda inşaa edilecek ağaç sayısını belirleyen *n_estimators* parametresinde öntanımlı gelen 10 değeri kullanılmıştır. Yani ormanda 10 ağaç ile işlem yapılacaktır. Bu algoritmada kullanılan son parametre ise *max_features* parametresidir ve 1 olarak alınmıştır. Bu parametre ağacın yaprak düğümüne gelinceye kadar ne kadar genişlemesi gerektiğini söyler. Bu parametre overfit sorununu çözmede başarılıdır. Bu sınıflama için kullanılan komut aşağıdaki gibidir. Diğer parametreler ön tanımlı olarak verilmiştir. SVM algoritmasında veri kümelerini ayırmada kullanılan destek vektör modeli *kernel* parametresi *lineer* olarak seçilmiştir. Ayrıca Düzenlileştirme (*Regularization*) parametresi olan *C* ise 1 tercih edilmiştir. K-En yakın komşu algoritmasında komşu sayısı belirleyen *k* parametresi 3 olarak seçilmiştir. SGD sınıflandırmasında *max_iter* değeri 1000 olarak alınmıştır. Naive Bayes sınıflandırmada, sınıflandırma fonksiyonu olan *GaussianNB()* parametresiz çalıştırılmıştır. MLP sınıflandırma algoritmasında SGD' de olduğu gibi *max_iter* değeri 1000 olarak alınmıştır. Gauss süreç sınıflandırmasında ise sınıf çekirdek sayısı belirlemede kullanılan *kernel* parametresi ön tanımlı hali ile (*1.0 * RBF(1.0)*) kullanılmıştır.

Tablo 3: Algoritmalar ve Kullanılan Foksiyonlar

Karar Ağacı	*>>> DecisionTreeClassifier(max_depth)*
Rastgele Orman	*>>> RandomForestClassifier(n_estimators, max_features, max_depth)*
Destek Vektör Makineleri	*>>> svm.SVC(kernel, C)*
K-En Yakın Komşu	*>>> KNeighborsClassifier(n_neighbors=3)*
Stokastik Dereceli İniş	*>>>SGDClassifier(max_iter)*
Naive Bayes	*>>> GaussianNB()*
MLP Nöral Network	>>> MLPClassifier(max_iter)
GaussianİşlemleriGaussianİşlemleri	*>>> GaussianProcessClassifier(kernel)*

Derin Öğrenme hesaplamalarında TensorFlow+Keras kütüphanesinden yararlanılmıştır. Hesaplamalara ilişkin bütün bilgiler ve ayrıntılı sonuçlar daha önceki çalışmamızda verilmiştir (Karakaplan ve Avcu., 2021). Yapay sinir ağımız; 1 Input, 1 Output ve 2 Hidden Layer den oluşmaktadır. Tablo 2 de verilen 16 parametreden 4'ü kullanılarak %93,8 doğruluk derecesi ile sınıflama yapılabilirliği gözlenmiştir. Bu parametreler kümedeki sayı (Ncls), inhibisyon sabiti (Ki), topolojik polar yüzey alanı (SfA) ve moleküler ağırlıktır. Derin öğrenme hesaplamasında aşırı uyum gösterme eğilimi (overfitting) oluşmaması için 5 input layere karşı 6 nodlu iki hidden layer ve 200 epochs değeri kullanılmıştır. Elde edilen hesaplama sonuçları Tablo 4'de verilmiştir.

Tablo 4: Makine Öğrenme Algoritmaların Sonuçları

Karar Ağacı	Rastgele Orman	Destek Vektör Makineleri	K-En Yakın Komşu	Stokastik Dereceli İniş	Naive Bayes	MLP Nöral Network	Gaussian İşlemleri	Hibrit Derin Öğrenme
%50	%25	%25	%75	%65	%50	%75	%50	%93,8

BULGULAR VE TARTIŞMA

Bu çalışmada veri sayısı az olan çalışmaların sınıflandırma işlemleri için kullanılan makine öğrenme algoritmaları karşılaştırılmıştır. Karşılaştırma için popüler agonist ve antagonist ligantlar kullanılmıştır. Karşılaştırma olarak da daha önceki çalışmalarımızda geliştirilen bir genetik algoritma-derin öğrenme hibrit algoritması kullanılmıştır. Tablo 3 olarak verilen hesaplama sonuçlarına bakılacak olursa, hibrit algoritma diğer sınıflama yöntemlerine göre daha iyi sonuç vermektedir. Beklendiği gibi MLP Neural Network sınıflaması yine içerisinde yapay sinir ağı taşıyan bir algoritma olduğundan Derin Öğrenme işlemlerine yakın sonuçlar vermiştir. K-NN'de ise komşu sayısı parametresini biz belirlediğimizden sonuç MLP algoritması ile aynı çıkmıştır. Yapılan bu çalışmada derin öğrenmenin az veri olması durumunda da diğer algoritmalara göre üstün olduğu görüldü. Geliştirdiğimiz hibrit algoritma bütün stokastik süreçleri de içerisinde taşımaktadır. Bu yöntem kimyasal veri tabanlarından elde edilen moleküllerin bilgileri ile ilaç tarama ve

sınıflandırma çalışmalarına veri sağlamak için kullanılabilir. Sonraki çalışmalarımızda ise hibrit modelin denetimsiz öğrenme algoritmaları ile kıyaslamayı planlamaktayız.

KAYNAKLAR

Aguiar, J. A., Gong, M. L., Tasdizen, T. (2020). Crystallographic prediction from diffraction and chemistry data for higher throughput classification using machine learning. Computational Materials Science, 173, 109409.

Altman, N. S. (1992). An Introduction to Kernel and Nearest-Neighbor Nonparametric Regression. The American Statistician, 46(3), 175–185.

Cortes, C., Vapnik, V. (1995). Support-vector networks. Machine Learning, 20(3), 273–297.

De Vito, S., Esposito, E., Salvato, M., Popoola, O., Formisano, F., Jones, R., Di Francia, G. (2018). Calibrating chemical multisensory devices for real world applications: An in-depth comparison of quantitative machine learning approaches, Sensors and Actuators B: Chemical, 255, 1191–1210.

Deng, L., Yu, D. (2014). Deep Learning: Methods and Applications. Foundations and Trends in Signal Processing, 7(3–4), 197–387.

Ding, W., Tong, Y., Zhang, Q., Yang, D. (2008). Image and video quality assessment using neural network and SVM. Tsinghua Science and Technology, 13(1), 112–116.

Drouhard, J.-P., Sabourin, R., Godbout, M. (1996). A neural network approach to off-line signature verification using directional PDF. Pattern Recognition, 29(3), 415–424.

Friedl, M. A., Brodley, C. E. (1997). Decision tree classification of land cover from remotely sensed data, Remote Sensing of Environment, 61(3), 399–409.

Furey, T. S., Cristianini, N., Duffy, N., Bednarski, D. W., Schummer, M., Haussler, D. (2000). Support vector machine classification and validation of cancer tissue samples using microarray expression data. Bioinformatics, 16(10), 906–914.

Goh, G. B., Hodas, N. O., Vishnu, A. (2017). Deep learning for computational chemistry. Journal of Computational Chemistry, 38(16), 1291–1307.

Grömping, U. (2009). Variable Importance Assessment in Regression: Linear Regression versus Random Forest. The American Statistician, 63(4), 308–319.

Gumus, O., Yasar, E., Gumus, Z. P., Ertas, H. (2020). Comparison of different classification algorithms to identify geographic origins of olive oils. Journal of Food Science and Technology, 57(4), 1535–1543.

Judson, R., Elloumi, F., Setzer, R. W., Li, Z., Shah, I. (2008). A comparison of machine learning algorithms for chemical toxicity classification using a simulated multi-scale data model. BMC Bioinformatics, 9(1), 241.

Karakaplan, M., Avcu, F. M. (2013). A parallel and non-parallel genetic algorithm for deconvolution of NMR spectra peaks. Chemometrics and Intelligent Laboratory Systems, 125, 147-152.

Karakaplan, M., Avcu, F. M. (2021). Classification of some chemical drugs by genetic algorithm and deep neural network hybrid method. Concurrency and Computation: Practice and Experience, 33(13), e6242.

Kumar, J., Singh, A. K. (2018). Workload prediction in cloud using artificial neural network and adaptive differential evolution. Future Generation Computer Systems, 81, 41–52.

Leen, T. K., Dietterich, T. G., Tresp, V. (2001). Advances in Neural Information Processing Systems 13: Proceedings of the 2000 Conference. MIT Press.

Maron, M. E. (1961). Automatic Indexing: An Experimental Inquiry, Journal of the ACM, 8(3), 404–417.

Mayr, A., Klambauer, G., Unterthiner, T., Steijaert, M., K. Wegner, J., Ceulemans, H., …Hochreiter, S. (2018). Large-scale comparison of machine learning methods for drug target prediction on ChEMBL. Chemical Science, 9(24), 5441–5451.

Minerali, E., Foil, D. H., Zorn, K. M., Lane, T. R., Ekins, S. (2020). Comparing Machine Learning Algorithms for Predicting Drug-Induced Liver Injury (DILI). Molecular Pharmaceutics, 17(7), 2628–2637.

Pal, M. (2005). Random forest classifier for remote sensing classification. International Journal of Remote Sensing, 26(1), 217–222.

PyChem homepage | PyChem. (n.d.). 7 Kasım 2021 tarihinde, http://pychem.sourceforge.net/ adresinden erişildi.

Python.org.. Python.Org. 7 Kasım 2021 tarihinde,https://www.python.org/ adresinden erişildi.

Russo, D. P., Zorn, K. M., Clark, A. M., Zhu, H., Ekins, S. (2018). Comparing Multiple Machine Learning Algorithms and Metrics for Estrogen Receptor Binding Prediction. Molecular Pharmaceutics, 15(10), 4361–4370.

Schmidhuber, J. (2015). Deep learning in neural networks: An overview. Neural Networks, 61, 85–117.

Scikit-learn: Machine learning in Python—Scikit-learn 1.0.1 documentation. 7 Kasım 2021 tarihinde https://scikit-learn.org/stable/ adresinden erişildi.

Sekeroglu, B. (2004). Classification of sonar images using back propagation neural network, IGARSS 2004. 2004 IEEE International Geoscience and Remote Sensing Symposium, 5, 3092–3095 vol.5.

Taddy, M. (2019). Business Data Science: Combining Machine Learning and Economics to Optimize, Automate, and Accelerate Business Decisions, McGraw Hill Professional.

TensorFlow.. TensorFlow. 7 Kasım 2021 tarihinde, https://www.tensorflow.org/ adresinden erişildi.

Tso, G. K. F., Yau, K. K. W. (2007). Predicting electricity energy consumption: A comparison of regression analysis, decision tree and neural networks. Energy, 32(9), 1761–1768.

Valiev, M., Bylaska, E. J., Govind, N., Kowalski, K., Straatsma, T. P., Van Dam, H. J. J., … de Jong, W. A. (2010). NWChem: A comprehensive and scalable open-source solution for large scale molecular simulations. Computer Physics Communications, 181(9), 1477–1489.

Xie, Y., Zhang, C., Hu, X., Zhang, C., Kelley, S. P., Atwood, J. L., Lin, J. (2020). Machine Learning Assisted Synthesis of Metal–Organic Nanocapsules. Journal of the American Chemical Society, 142(3), 1475–1481.