Makine Öğrenmesi Modellerinin Analizi ve Karşılaştırılması

İbrahim AYDIN[1]

[1]Yazılım Mühendisliği
İzmir Kâtip Çelebi Üniversitesi
ibraydin@gmail.com

Özet

Diyabet kan şekerinin yükselmesine sebep olan kronik bir hastalıktır. Bu kronik hastalık ölümcül olabilmektedir. Tıp alanında diyabet için farklı tanı ve tedavi yöntemleri kullanmaktadır. Bilgisayar destekli teşhis yöntemleri başarılı, hızlı ve doktor kararını destekleyici alternatif bir yöntemdir. Diyabet ve daha birçok hastalık için bilgisayar destekli teşhis yaklaşımı kullanımı gün geçtikçe artmaktadır. Makine öğrenmesi sınıflandırma yöntemleri bilgisayar destekli teşhis için en sık kullanılan yöntemlerdir. Bu çalışmanın amacı, hastalarda diyabet olasılığını maksimum doğrulukla erken bir aşamada tespit etmek için model tasarlamaktır. Bu nedenle, öncelikle KNN algoritması üzerinde durulmuştur. KNN algoritması ve Kaggle veritabanında bulunan "diabetes.csv" veri seti üzerinde detaylı çalışılmıştır. Sonrasında ise Nearest_Neighbors,Linear_SVM,Polynomial_SVM,RBF_SVM,Gaussian_Process,Gradient_Boosting, Decision_Tree, Extra_Trees, Random_Forest, Neural_Net, AdaBoost, Naive_Bayes, QDA ve SGD olmak üzere on üç makine öğrenimi sınıflandırma algoritmaları ile Sklearn kütüphanesi ve veri seti kullanılarak "score" değerlerinin performansları karşılaştırılmıştır.

Anahtar kelimeler: Makine öğrenmesi; diyabet; sınıflandırma; Yapay Zeka;KNN

Abstract

Diabetes is a chronic disease that causes high blood sugar. This chronic disease can be fatal. In the field of medicine, different diagnosis and treatment methods are used for diabetes. Computer-aided diagnostic methods are a successful, fast and alternative method that supports the doctor's decision. The use of computer aided diagnostic approach for diabetes and many other diseases is increasing day by day. Machine learning classification methods are the most commonly used methods for computer aided diagnosis. The aim of this study is to design a model to detect the possibility of diabetes in patients at an early stage with maximum accuracy. For this reason, first of all, the KNN algorithm is emphasized. KNN algorithm and "diabetes.csv" data set have been studied in detail. Afterwards, the performance of thirteen machine learning data sets, namely Nearest_Neighbors, Linear_SVM, Polynomial_SVM, RBF_SVM, Gaussian_Process, Gradient_Boosting, Decision_Tree, Extra_Trees, Random_Forest, Neural_Net, AdaBoost, Naive_Bayes, QDA and SGD, are compared with the performance of thirteen machine learning datasets classification algorithms using the library's dataset classification and classification values.

Keywords: Machine learning; diabetes; classification; Artificial Intelligence;KNN

1. Giriş

Diyabet, vücudumuzda yeterli miktarda insülin hormonu üretilememesi ya da üretilen insülinin etkili şekilde kullanılamaması durumunda kan şekerinin yükselmesine neden olan ölümcül ve kronik bir hastalıktır. Diyabet hastalığı tedavi edilmediği taktirde ve tanımlanamazsa birçok komplikasyon meydana gelebilir. Bu nedenle modern tıpta, diyabeti mümkün olduğunca erken tespit etmek önemlidir. Genel teşhis yöntemi, hastanın teşhis merkezine gidip doktora danışmasıyla sonuçlanır. Ancak makine öğrenmesi ve yapay zeka yaklaşımı buna yeni bir çözüm getirmektedir. Hastaya ait veriler bilgisayar ortamında işlenerek modeller eğitilir ve daha sonra hiç görmediği bir hastanın bilgileri verildiğinde model hastalık sonucu hakkında karar verir.

Bu tür algoritmalar statik program talimatlarını harfiyen yerine getirmek yerine örnek girişlerden tahminleri ve kararları gerçekleştirebilmek için bir model inşa ederek çalışırlar. Literatürde makine öğrenimi modelleri veterinerlik [13], inşaat [4], çevre [5], tıp [6], salgınlar [7] gibi hemen hemen tüm disiplinlerde başarılı bir şekilde uygulanmaktadır.

Kumari ve Chitra [8], diyabet hastalığını sınıflandırmak için SVM algoritmasını kullanılmıştır. Çalışmada %78 doğruluk, %80 hassasiyet ve %76.5 özgüllük oranı elde edilmiştir. Yu vd. [9], tarafından yapılan çalışmada farklı özniteliklere sahip iki veri seti oluşturularak SVM yöntemi ile sınıflandırma yapılmıştır. Aile öyküsü, yaş, ırk ve etnik köken, ağırlık, boy, bel çevresi, vücut kitle indeksi ve hipertansiyon özniteliklerini içeren birinci veri seti için %83.5, ikinci veri setinde bu özniteliklere ek olarak cinsiyet ve fiziksel aktivite eklenmiştir ve burada ise %73.2 doğruluk elde edilmiştir.

Sajida vd. [10], Diabetes Mellitus ve hastaları diyabet risk faktörlerine göre diyabetik veya diyabetik olmayan olarak sınıflandırmak için J48 karar ağacını kullanan Adaboost ve Bagging topluluk makine öğrenimi yöntemlerinin rolünü incelemişlerdir. Elde edilen sonuçlara göre, Adaboost makine öğrenimi topluluk yönteminin, bagging yönteminden daha iyi performans gösterdiği bildirilmiştir.

Bu çalışmanın amacı, farklı makine öğrenmesi sınıflandırma yöntemlerinin diyabet hastalığı teşhisindeki performanslarını karşılaştırmak ve en başarılı yöntemi belirlemektir. Bu amaçla çalışmada, on üç farklı makine öğrenmesi kullanılmıştır. Çalışmada Python programlama dili kullanılmıştır.

2. Materyal ve Metotlar

A. Veri Seti

Çalışmada kullanılan veri seti Kaggle veri tabanında bulunan Pima Indians Diabetes Database (PIDD) veri seti [12] üzerinde gerçekleştirildi. Veri seti 8 girdi, 1 çıktı değişkeni ve 768 örnek içermektedir. Veri setindeki öznitelikler ve bu özniteliklerin kısaltmaları Tablo I'de verilmiştir.

Tablo 1. Veri Setindeki Özellikler ve Kısaltmaları

	Öznitelik
1	Pregnancies: Hamile Sayısı
2	Glucose:Plazma Glikoz Konsantrasyonu
3	BloodPressure: Kan Basıncı(mm Hg)
4	SkinThickness: Cilt Kalınlığı(mm)
5	Insulin: Serum İnsulin Değeri(mu U/ml)
6	BMI: Boy-Kilo indeksi (kg/(m)^2)
7	DiabetesPedigreeFunction: Soyağacının fonksiyonu
8	Age:Yaş
9	Outcome: Çıktı (Sağlıklı:0 veya Şeker Hastası:1)

B. Makine Öğrenmesi Sınıflandırma Yöntemleri

Bu çalışmada, Python programlama dili, Anaconda ve Jupyter Notebook web çatıları ve Sklearn ile Keras modülleri kullanılmıştır.

Bazı algoritmalardan bahsetmek gerekirse;

Lojistik Regresyon (LogR, Logistic Regression), sınıflandırma problemlerinde bağımlı ve bağımsız değişkenler arasındaki ilişkiyi bir doğru ile ifade etmeye çalışır. Burada bağımlı değişken kategorik değerler iken bağımsız değişken ikili değerler ile ölçülüdür. Yani regresyon sonucunda olası iki sonuç vardır. Bu çalışmada, lojistik regresyon için sklearn.linear_model kütüphanesindeki LogisticRegression modeli kullanılmıştır.

Destek Vektör Makinesi (SVM, Support Vector Machine), sınıflandırma problemlerinde kullanılır. Bir gözetimli öğrenme çeşididir. Çalışma mantığı düzlem üzerindeki noktaları ayırmak için bir doğru çizer, noktalar bizim veri setimizi temsil eder. Bu doğrunun böldüğü sınıfların arasındaki mesafenin maksimum olmasını amaçlar. Bu aradaki mesafeye margin değeri denir. Küçük ve karmaşık veri setleri için uygundur. Bu çalışmada, sklearn.svm kütüphanesindeki SVC modeli kullanılmıştır. Kernel fonksiyonu linear seçilmiştir.

Karar Ağaçları (DT, Decision Tree), veri kümeleri içerisinde belirlenen karar kurallarını uygulayarak bir veri düğümünü iki veya daha fazla düğüme bölmeyi hedefleyen yapılardır. Alt düğümler oluştukça oluşan düğümlerin homojenliği artar. Bu sayede veri sınıflandırılır. Bölme işlemi problem tipine göre değişen algoritmalar ile gerçekleştirilir. En sık kullanılan algoritmalar kategorik değişkenler için Gini ve Entropy; sürekli değişkenler için en küçük kareler yöntemidir. Karar ağacı yöntemi için, sklearn.tree kütüphanesindeki DecisionTreeClassifier metodu kullanılmıştır.

Gauss Naif Bayes (GNB, Gaussian Naive Bayes), olasılık tabanlı bir algoritma olup bir rassal değişken için koşullu olasılıklar ile marjinal olasılıklar arasındaki ilişkiyi inceler. Bunun için sklearn.naive_bayes kütüphanesindeki GaussianNB modeli kullanılmıştır.

K-En Yakın Komşu (KNN, K-Nearest Neighbors), bir veriyi daha önce eklenmiş verilerle olan yakınlık derecesine göre sınıflandırma işlemi yapar. Algoritma tasarlanırken bir k değeri belirlenir. Bu k değeri eklenecek verinin sınıflandırılırken kaç adet komşusunu referans almamız gerektiğini belirtir. Bu sebeple algoritma için önemlidir. Çalışmada, sklearn.neighbors kütüphanesindeki KNeighborsClassifier metodu kullanılmıştır.

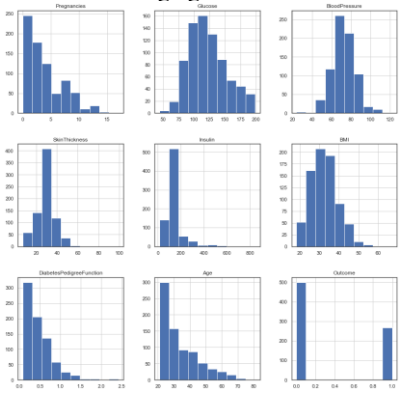
Rassal Orman (RF, Random Forest), karar ağacı algoritması gibi sınıflandırma yaparken bir orman oluşturur. Ama karar ağaçlarından farklı olarak rastgele bir orman oluşturur. Veri setindeki verileri böler ve belirlenen sayıda ağaç oluşturur. Sınıflandırma işlemi gerçekleştirilirken ilgili verinin tüm ağaçlardaki konumuna bakılır ve sonuç oy birliği ile belirlenir. Çalışmada, sklearn.ensemble kütüphanesindeki RandomForestClassifier modeli kullanılmıştır.

Yapay Sinir Ağları (ANN, Artificial neural network), insan beyninin özelliklerinden olan öğrenerek yeni bilgiler türetebilme yeteneğini otomatik olarak gerçekleştiren bilgisayar ağlarıdır. Yapay Sinir Ağları, insan beyni örnek alınarak öğrenme sürecini matematiksel ifadelerle modellenmesi ile ortaya çıkmıştır. Biyolojik sinir ağlarındaki öğrenme, hatırlama ve genelleme yapabilme özelliklerini taklit eder. Öğrenme işlemi örnekler üzerinden gerçekleştirilir. Ağ, eğitim esnasında veri seti üzerinden doğru sonuçlara ulaşmak için ağırlık değerlerini güncelleyerek genel ve doğru bir model oluşturmaya çalışır.

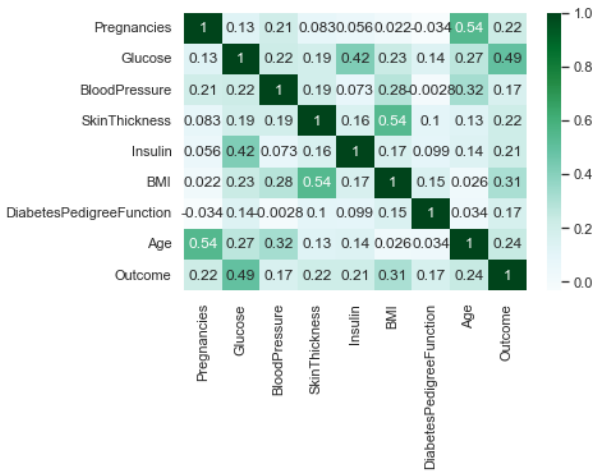
3. Bulgular

A. Veri Seti ve Özniteliklerin Analizi

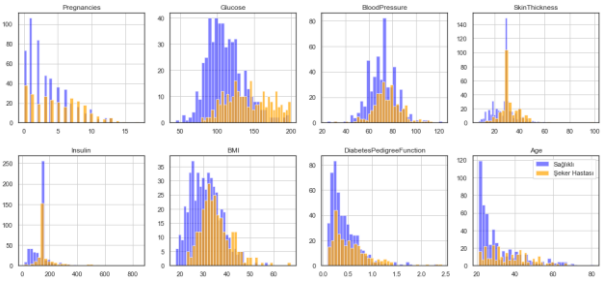
Diyabet veri setinde bireylerin sağlık durumu '0' veya '1' olarak gösterilmiştir. '0' bireyin diyabet hastası olmadığını, '1' ise bireyin diyabet hastası olduğunu göstermektedir. Veri setinde 500 birey diyabet hastası ve 268 birey ise diyabet hastası değildir. Veri setindeki özniteliklerin dağılımları Şekil 1'de korelasyon matrisi ise Şekil 2.'de sunulmuştur. Koyu renkler baskın ilişkiyi gösteriyor. Mesela Şeker hastalığı ile en ilişkili değer %49(0.49) ile Glikoz değeridir. İkinci en ilişkili öznitelik ise %42(0.42) insulin değeridir. Şekil 3.'de ise her bir özniteliğin çıktı ile ilgili ilişkisi görülmektedir. Burada Mavi olan değerler sağlıklı insanları (0), Turuncu değerler ise Şeker hastalarını (1) temsil ediyor. Glikoz değerinin diyabet hastalığı ile en ilişkili öznitelik olduğu görünmektedir.



Şekil 1: Özniteliklerin dağılımı



Şekil 2: Korelasyon matrisi.



Şekil 3: Her bir özniteliğin çıktı ile ilişkisi

Ön işlem süreci tamamlandıktan sonra modelleme işlemine geçilmiştir. Modellerin sınıflandırma performansları iki farklı veri seti ayırma tekniği üzerinde test edilmiştir. Bunlardan ilkinde veri setinin %80'i eğitim, %20'si test kümesi olarak ayrılmıştır.

Öncelikle veri setinde her öznitelik ağırlık değerleri 0-1 arasında değerlere dönüştürülerek normalizasyon yapılmıştır. Şekil 4.'de normalizasyon öncesi ve sonrası veri seti değeri görülmektedir.

```
Normalization öncesi ham veriler:

   Pregnancies  Glucose  BloodPressure  SkinThickness     Insulin   BMI  \
0            6    148.0           72.0       35.00000  155.548223  33.6
1            1     85.0           66.0       29.00000  155.548223  26.6
2            8    183.0           64.0       29.15342  155.548223  23.3
3            1     89.0           66.0       23.00000   94.000000  28.1
4            0    137.0           40.0       35.00000  168.000000  43.1

   DiabetesPedigreeFunction  Age
0                     0.627   50
1                     0.351   31
2                     0.672   32
3                     0.167   21
4                     2.288   33


Normalization sonrası yapay zekaya eğitim için vereceğimiz veriler:

   Pregnancies   Glucose  BloodPressure  SkinThickness   Insulin       BMI  \
0     0.352941  0.670968       0.489796       0.304348  0.170130  0.314928
1     0.058824  0.264516       0.428571       0.239130  0.170130  0.171779
2     0.470588  0.896774       0.408163       0.240798  0.170130  0.104294
3     0.058824  0.290323       0.428571       0.173913  0.096154  0.202454
4     0.000000  0.600000       0.163265       0.304348  0.185096  0.509202

   DiabetesPedigreeFunction       Age
0                  0.234415  0.483333
1                  0.116567  0.166667
2                  0.253629  0.183333
3                  0.038002  0.000000
4                  0.943638  0.200000
```

Şekil 4: Normalizasyon öncesi ve sonrası veriler

B. Sklearn ile Algoritmaların Karşılaştırılması

Sklearn ile algoritmaların karşılaştırılması Jüpiter Notebook kullanılarak gerçekleştirilmiştir. Şekil 5'de ilgili Python kodları görülmektedir.

```
from sklearn.ensemble import GradientBoostingClassifier
from sklearn.gaussian_process.kernels import RBF
from sklearn.tree import DecisionTreeClassifier
from sklearn.ensemble import ExtraTreesClassifier
from sklearn.ensemble import RandomForestClassifier, AdaBoostClassifier
from sklearn.naive_bayes import GaussianNB
from sklearn.discriminant_analysis import QuadraticDiscriminantAnalysis
from sklearn.linear_model import SGDClassifier
```

```
names = ["Nearest_Neighbors", "Linear_SVM", "Polynomial_SVM", "RBF_SVM", "Gaussian_Process",
         "Gradient_Boosting", "Decision_Tree", "Extra_Trees", "Random_Forest", "Neural_Net", "AdaBoost",
         "Naive_Bayes", "QDA", "SGD"]

classifiers = [
    KNeighborsClassifier(3),
    SVC(kernel="linear",C=0.01),
    SVC(kernel="poly", degree=3, C=0.025),
    SVC(kernel="rbf", C=1, gamma=2),
    GaussianProcessClassifier(1.0 * RBF(1.0)),
    GradientBoostingClassifier(n_estimators=100, learning_rate=1.0),
    DecisionTreeClassifier(max_depth=4),
    ExtraTreesClassifier(n_estimators=20, min_samples_split=3),
    RandomForestClassifier(max_depth=10, n_estimators=500),
    MLPClassifier(alpha=1, max_iter=1000),
    AdaBoostClassifier(n_estimators=100),
    GaussianNB(),
    QuadraticDiscriminantAnalysis(),
    SGDClassifier(loss="huber", penalty="l2")]
```

```
import pandas as pd
import seaborn as sns
```

```
scores = []
for name, clf in zip(names, classifiers):
    clf.fit(X_train, Y_train)
    score = clf.score(X_test, Y_test)
    scores.append(score)
```

```
df = pd.DataFrame()
df['name'] = names
df['score'] = scores
df
```

Şekil 5: Normalizasyon öncesi ve sonrası veriler

Aşağıda ise her bir algoritma modelinin ilgili parametrelerine verilen farklı değerlere karşılık alınan "score" değerleri verilmiştir. Her bir algoritmanın en yüksek score değerleri kombinasonu * işareti ile işaretlenmiştir. Şekil 6, Şekil 7 ve Şekil 8'de ise en yüksek score değerleri topluca gösterilmiştir.

K Neighbors Classifier

* K=3 iken Score= 0.820,
K=5 iken Score= 0.805,
K=7 iken Score= 0.815,
K=9 iken Score= 0.810,
K=11 iken Score= 0.820

Linear_SVM

C=0.025 iken Score= 0.815,
C=0.001 iken Score= 0.815,
C=0.00005 iken Score=0.500,
C=0.000001 iken Score=0.500,
* C=0.1 iken Score= 0.825

Polynomal_SVM

* degree=3 iken Score=0.785,
degree=2 iken Score= 0.505,
degree=4 iken Score=0.555,
degree=5 iken Score=0.740,
degree=6 iken Score=0.505,

RBF_SVM
* C=1 ve gamma=2 iken Score=0.810,
C=1 ve gamma=3 iken Score=0.780,
C=2 ve gamma=2 iken Score=0.805,
C=3 ve gamma=2 iken Score=0.810

Gaussan Process Classifier

* 1.0 * RBF(1.0) için Score=0.835,
1.0 RBF(0.05) için Score=0.780,
1.0 RBF(0.005) için Score=0.500,
1.0 RBF(0.01) için Score=0.505,
1.0 RBF(0.2) için Score=0.835

Gradent Boosting Classifier

* n_estimators=100 ve learning_rate=1.0 iken Score=0.785,
n_estimators=200 ve learning_rate=1.0 iken Score=0.785,
n_estimators=100 ve learning_rate=2.0 iken Score=0.205,
n_estimators=300 ve learning_rate=1.0 iken Score=0.765,
n_estimators=400 ve learning_rate=1.0 iken Score=0.760

Decison Tree Classifier

max_depth=5 iken Score=0.825,
* max_depth=4 iken Score=0.830,
max_depth=3 iken Score=0.815,
max_depth=6 iken Score=0.820,
max_depth=7 iken Score=0.795,
max_depth=8 iken Score=0.795

Extra Trees Classifier

n_estimators=10 ve min_samples_split=2 iken Score=0.825,
n_estimators=10 ve min_samples_split=3 iken Score=0.820,
n_estimators=10 ve min_samples_split=4 iken Score=0.815,
* n_estimators=20 ve min_samples_split=3 iken Score=0.830,
n_estimators=20 ve min_samples_split=5 iken Score=0.830,
n_estimators=20 ve min_samples_split=6 iken Score=0.825

Random Forest Classifier

max_depth=5 ve n_estimators=100 iken Score=0.820,
max_depth=4 ve n_estimators=100 iken Score=0.825,
max_depth=5 ve n_estimators=200 iken Score=0.820,
* max_depth=10 ve n_estimators=500 iken Score=0.830,
max_depth=20 ve n_estimators=500 iken Score=0.830,
max_depth=10 ve n_estimators=700 iken Score=0.830,
max_depth=10 ve n_estimators=800 iken Score=0.825

MLP Classifier

* alpha=1 ve max_iter=1000 için Score=0.845,
alpha=2 ve max_iter=1000 için Score=0.835,
alpha=1 ve max_iter=2000 için Score=0.845,
alpha=3 ve max_iter=1000 için Score=0.825,
alpha=4 ve max_iter=1000 için Score=0.820

Ada Boost Classifier

n_estimators=100 için Score=0.805,
n_estimators=200 için Score=0.775,
n_estimators=300 için Score=0.775,
n_estimators=500 için Score=0.750,
* n_estimators=50 için Score=0.810,

Gaussian Naive Bayes

* Score=0.835

Quadratic Discriminant Analysis

* Score=0.825

SGD Classfer

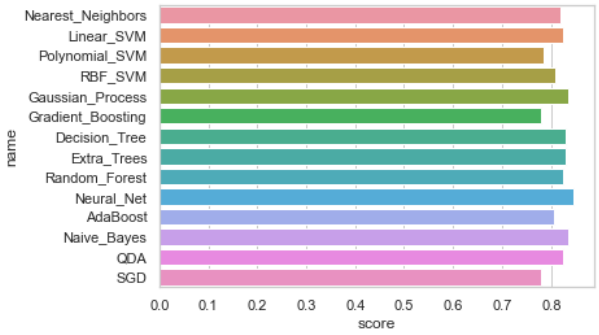
* loss="hinge" ve penalty="l2" iken Score=0.780,
loss="squared_hinge" ve penalty="l2" iken Score=0.565,
loss="perceptron" ve penalty="l2" iken Score=0.770,
loss="squared_error" ve penalty="l2" iken Score=0.675,
loss="huber" ve penalty="l2" iken Score=0.765,

	name	score
0	Nearest_Neighbors	0.820
1	Linear_SVM	0.825
2	Polynomial_SVM	0.785
3	RBF_SVM	0.810
4	Gaussian_Process	0.835
5	Gradient_Boosting	0.780
6	Decision_Tree	0.830
7	Extra_Trees	0.830
8	Random_Forest	0.825
9	Neural_Net	0.845
10	AdaBoost	0.805
11	Naive_Bayes	0.835
12	QDA	0.825
13	SGD	0.780

Şekil 6: En yüksek score değerleri

	name	score
5	Gradient_Boosting	0.780000
13	SGD	0.780000
2	Polynomial_SVM	0.785000
10	AdaBoost	0.805000
3	RBF_SVM	0.810000
0	Nearest_Neighbors	0.820000
1	Linear_SVM	0.825000
8	Random_Forest	0.825000
12	QDA	0.825000
6	Decision_Tree	0.830000
7	Extra_Trees	0.830000
4	Gaussian_Process	0.835000
11	Naive_Bayes	0.835000
9	Neural_Net	0.845000

Şekil 7: En yüksek score değerleri (sıralı)



Şekil 8: En yüksek score değerleri grafiği

4. Sonuç

Makine öğrenmesi yöntemleri, sağlık alanındaki erken tanı ve planlama konusunda kendisine yer edinmiştir. Özellikle maliyeti yüksek kronik rahatsızlıklarda makine öğrenmesi metotları oldukça kullanışlı hale gelmektedir. Diyabet gibi ölümcül bir hastalığın erken aşamada tespit edilmesi önemli bir tıbbi problemdir. Bu çalışmada, farklı makine öğrenmesi sınıflandırma yöntemlerinin diyabet hastalığını tahmin edilmedeki performansları karşılaştırılmıştır. Bunun için Pima Indians Diabetes veri seti kullanılmıştır. Deneysel sonuçlara göre veri seti rastgele eğitim (%80) ve test olarak (%20) ayrıldığında Neural_Net (MLP Classifier) algoritması alpha=1 ve max_iter=1000 için Score=0.845 (%84.5 Doğruluk Oranı) değerleri ile geçerli model için en yüksek score puanını veren algoritma olmuştur.

5. KAYNAKLAR

[1] Cihan, P., Gökçe, E. and Kalıpsız, O., "A review of machine learning applications in veterinary field", Kafkas Univ Vet Fak Derg, 23(4):673680,2017.

[2] Cihan, P., Kalıpsız, O. and Gökçe, E., "Yenidoğan kuzularda bilgisayar destekli tanı", Pamukkale Üniversitesi Mühendislik Bilimleri Dergisi, 26(2):385-391, 2020.

[3] Cihan, P., et al. "Prediction of Immunoglobulin G in Lambs with Artificial Intelligence Methods", Kafkas Üniversitesi Veteriner Fakültesi Dergisi, 27(1):21-27, 2021.

[4] Cihan., M. T., "Prediction of Concrete Compressive Strength and Slump by Machine Learning Methods", Advances in Civil Engineering, 2019, 2019.

[5] Cihan, P., Ozel, H., and Ozcan, H. K., “Modeling of atmospheric particulate matters via artificial intelligence methods”, Environmental Monitoring and Assessment, 193(5):1-15, 2021.

[6] Kononenko, I., "Machine learning for medical diagnosis: history, state of the art and perspective", Artificial Intelligence in medicine, 23:89-109, 2001.

[7] Cihan, P., "Fuzzy Rule-Based System for Predicting Daily Case in COVID-19 Outbreak", 2020 4th International Symposium on Multidisciplinary Studies and Innovative Technologies (ISMSIT). IEEE, 2020.

[8] Kumari, V. and Chitra, R., "Classification of diabetes disease using support vector machine", International Journal of Engineering Research and Applications, 3:1797-1801, 2013.

[9] Yu, W. et al. "Application of support vector machine modeling for prediction of common diseases: the case of diabetes and pre-diabetes", BMC medical informatics and decision making, 10:1-7, 2010.

[10] Sajida, P., et al. "Performance analysis of data mining classification techniques to predict diabetes", Procedia Computer Science, 82:115-121, 2016.

[11] Özkan Y., Sarer Yürekli B., Suner A., "Diyabet tanısının tahminlenmesinde denetimli makine öğrenme algoritmalarının performans karşılaştırması", Gümüşhane Üniversitesi Fen Bilimleri Enstitüsü Dergisi, 202

[12] Pima Indians Diabetes Database (PIDD), https://www.kaggle.com/saurabh00007/diabetescsv, accessed 23.10.2022.