****

**KAHRAMANMARAŞ SÜTÇÜ İMAM ÜNİVERSİTESİ**

**BİLGİSAYAR MÜHENDİSLİĞİ**

YAPAY ZEKÂ RAPORU

17110131052 - Mehtap ÖKLÜ

18110131011 - Banu KÖSE

**Özet**

İnsanların yeni bilgileri öğrenmek ve karar almak için kullandığı düşünsel becerinin, matematik ve mantık kullanan bilgisayarlar tarafından simüle edilmesine yapay zeka denir. Yapay zeka, mevcut verilerle işlemler yapan, ardından hatalarından ders çıkartarak kendisini geliştiren sistemlerdir.

Temellerinin 1943’te atılmasıyla gelişmeye başlayan yapay zeka günümüzde hayatımızın her alanında bize kolaylık sağlamaktadır. İçerisinde makine öğrenmesi ve derin öğrenme disiplinlerini de barındırmaktadır. Daha hızlı ve daha doğru kararlar vermemize yardımcı olan yapay zeka, sürücüsüz araçlar, akıllı ev teknolojileri, sağlık verilerinin analizi gibi çeşitli alanlarda yer almaktadır. [1][2]

1. **Giriş**

Bu projede kırmızı şarabın, belli başlı özelliklerine bakılır. Bunlar; fixed acidity, volatile acidity, citric acid, residual sugar, chlorides, free sülfür dioxide, density, pH, sulphates, alcohol, quality. Bu özellikler sayesinde yapay sinir ağı eğitilerek, hangi şarabın kaliteli hangi şarabın kalitesiz olduğu belirlenmesi amaçlanmıştır.

Veri setimizde birden çok özellik bulunmaktadır. Yukarıda belirttiğim özelliklere bakılarak en son oluşturulan “quality” sütunun hepsini toplayıp ortalamasını aldıktan sonra o ortalamanın altında kalanları “0” kalitesiz, ortalamanın üzerinde kalanları “1” kaliteli olarak ayırdık. Böylece veri setimizi işlenecek duruma getirdik.

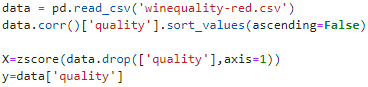
1. **Ağ Eğitme**

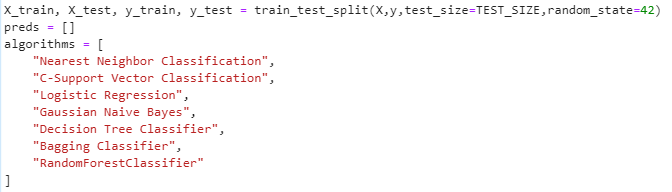
Yapay sinir ağlarında eğitme işlemi; veri setini tanımlama, ağ yapısını oluşturma ve eğitime sokarak belirli sonuçlar elde etme yöntemidir.

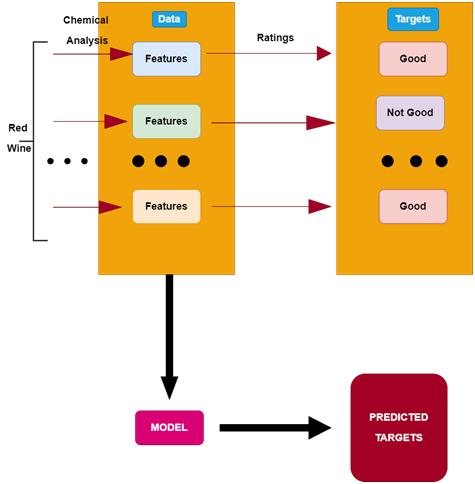
**2.1. Veri Seti**

Veri seti Kaggle platformu üzerinden bulduğumuz açık, .csv uzantılı bir veri settir. 12 öznitelikten oluşmakta olup toplam 1599 veri girişi vardır.

**2.2. Ağ Yapısını Oluşturma**



  
**Şekil 1 - Ağ Yapısı Oluşturma**

**  
Şekil 2 - Ağın Şematik Yapısı**

Yukarıda tasarladığımız kodda öncelikle setimizin yolunu verdik. Elimizde bulunan 12 öznitelik (Features) sayesinde vardığımız 2 sonuç çıkışımız olarak tanımlanmıştır.

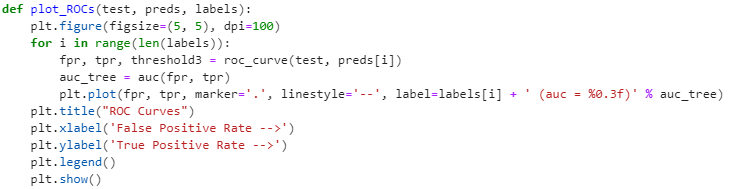
**2.3. Eğitim Sonuçları**

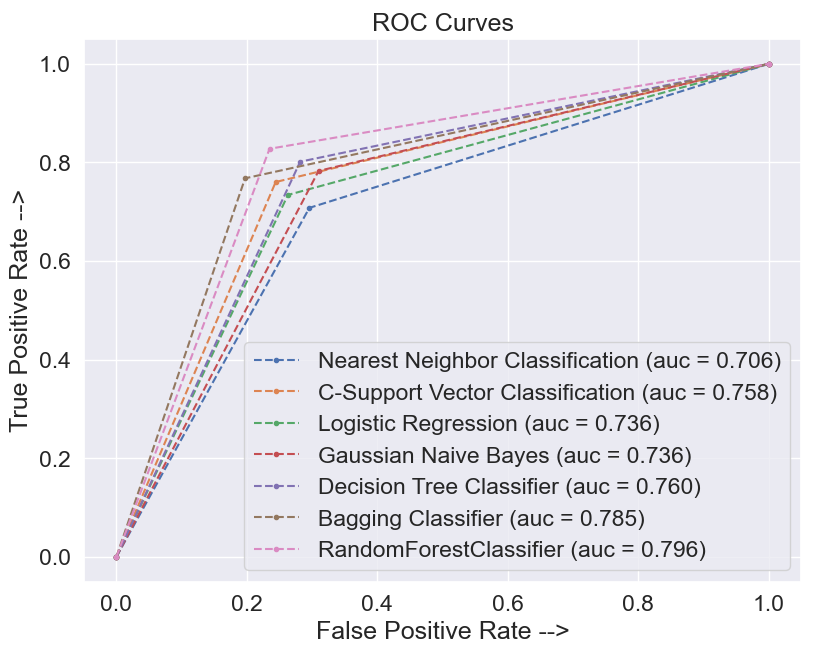
Eğitim sonuçları olarak ROC eğrisi, cross validation, confusion matrix ve MSE (Mean Squared Error) sonuçlarına ulaşacağız.

**2.3.1. ROC Eğrisi**

ROC eğrisi tanı testleri gibi sonuçları var/yok, yüksek/ normal şeklinde iki sonuçlu ise testlerin başarısının belirlenmesi yöntemidir. İkili sınıflandırma sistemlerinde ayrım eşik değerinin farklılık gösterdiği durumlarda, hassasiyetin kesinliğe olan oranıyla ortaya çıkmaktadır. [3]

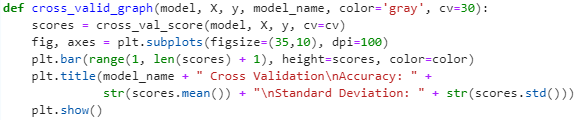
ROC eğrisi kodu ve gösterimi aşağıdaki gibidir.

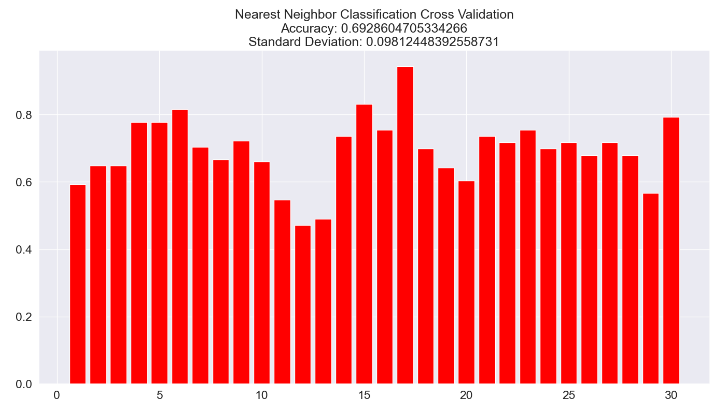
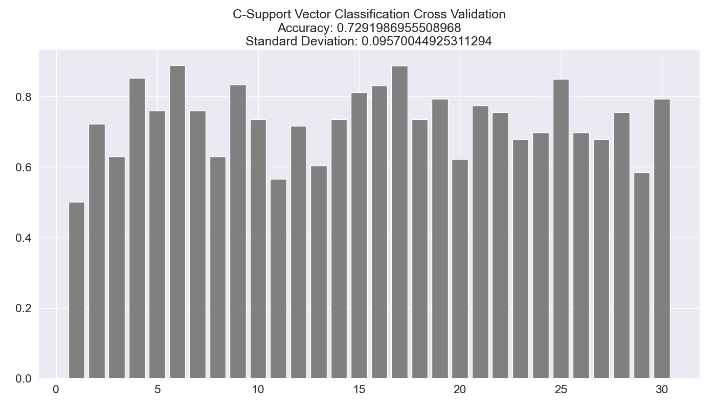
  
**Şekil 3 - ROC Eğrisi Gösterim Kodu**

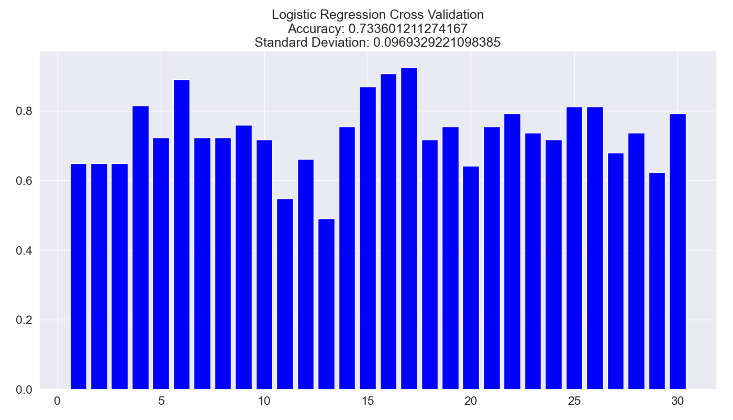
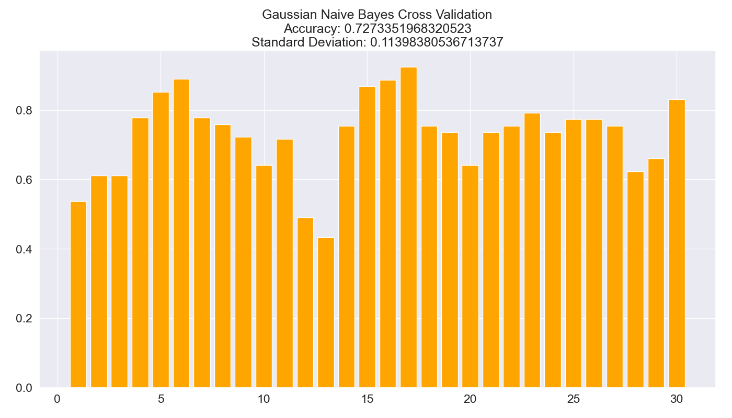
**  
Şekil 4 - ROC Eğrisi Gösterimi**

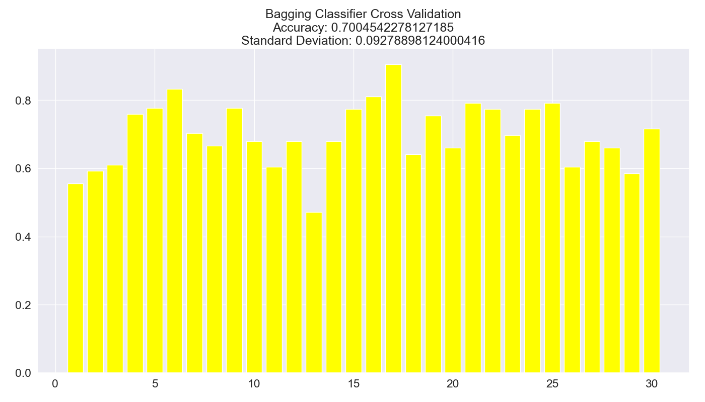
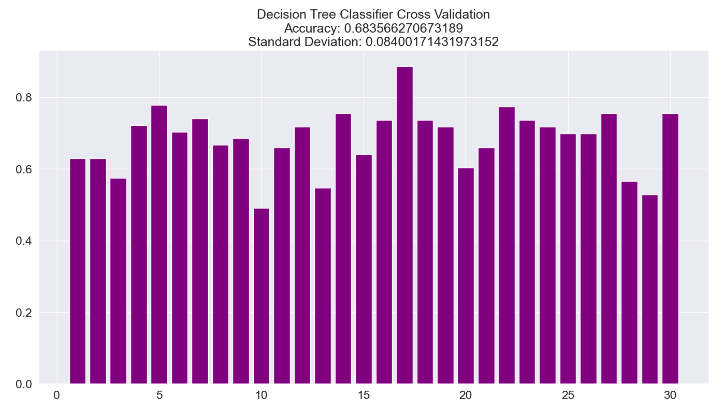
**2.3.2. Cross Validation**

Cross Validation, oluşturulan modeli birden fazla kez, farklı train test verileriyle fit ederek daha objektif ve doğru bir sonuç elde etmemizi sağlar. Cross Validation‘a ait kod bloğu ve çıktılar aşağıdaki gibidir.

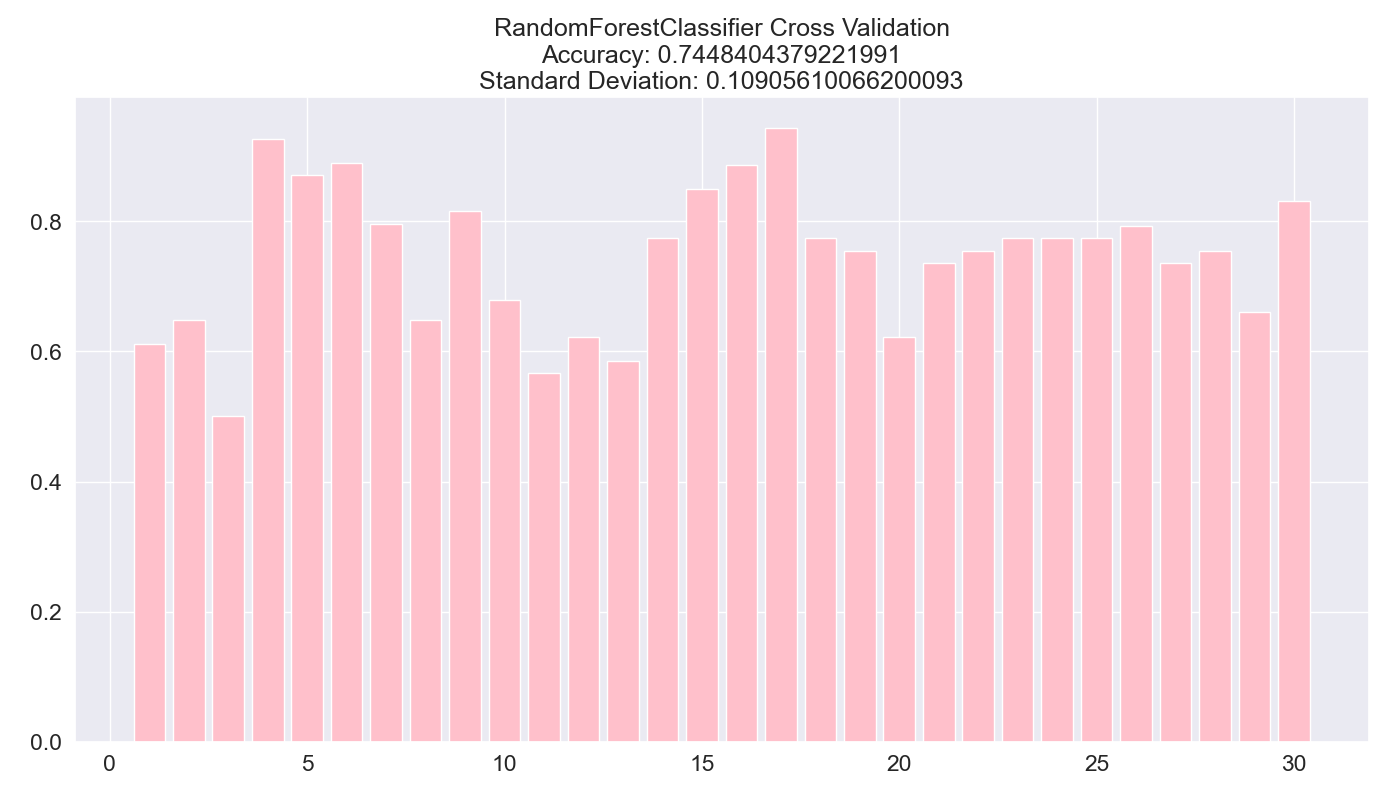
  
**Şekil 5 - Cross Validation Gösterim Kodu**





****

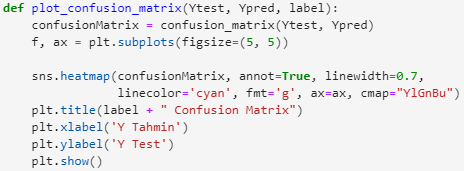
**Şekil 6 - Cross Validation Sonuçları**

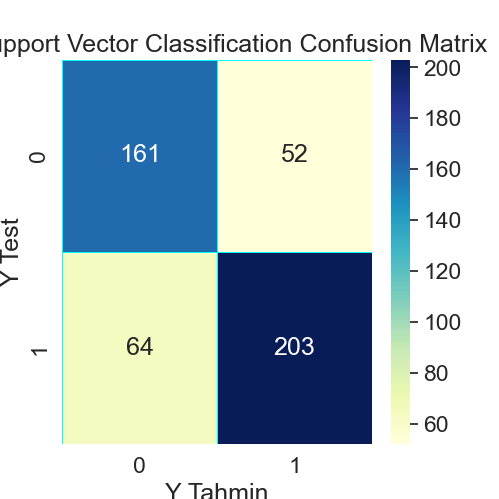
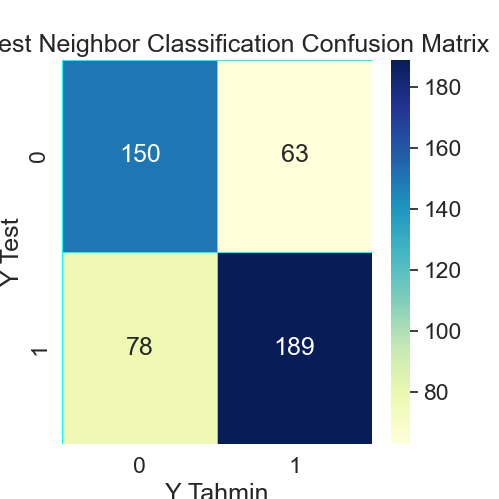


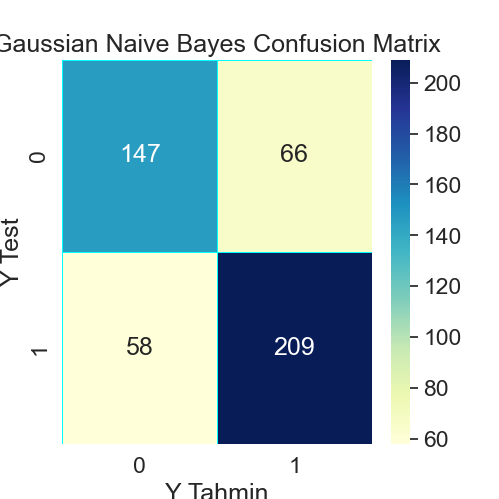
**2.3.3. Confusion Matrix**

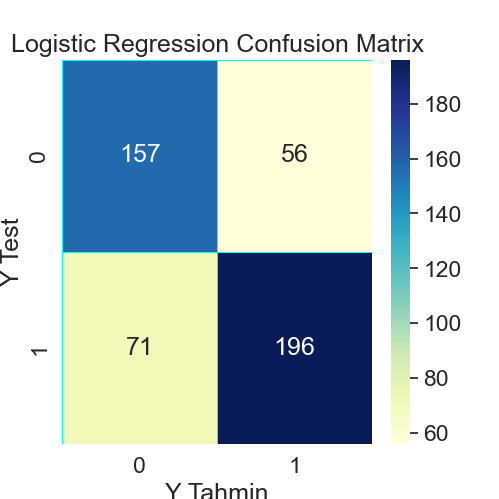
Sınıflandırma modellerinin performanslarını ölçmek için kullanılan bir makine öğrenmesi yöntemidir. Tahminlerin ve gerçek değerlerin karşılaştırıldığı hata matrisi yaygın olarak kullanılmaktadır.[4]

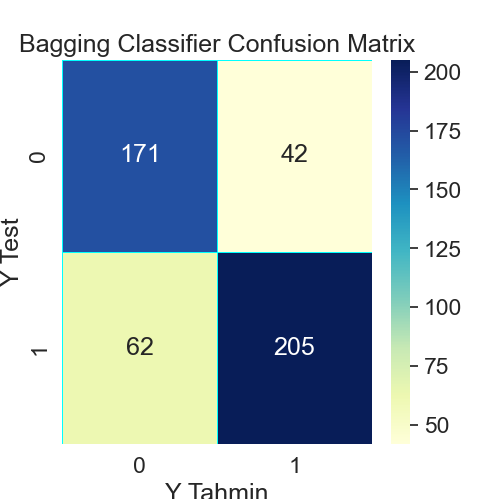
Confusion matrix kodu her sınıf için ayrı ayrı oluşturulmuş olup gösterimi aşağıdaki gibidir.

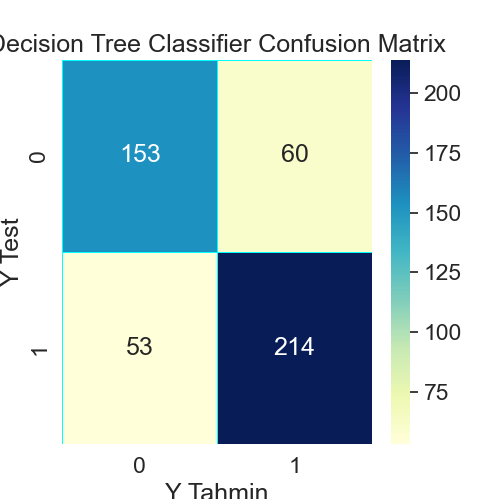
  
**Şekil 7 - Confusion Matrix Kod Bloğu**



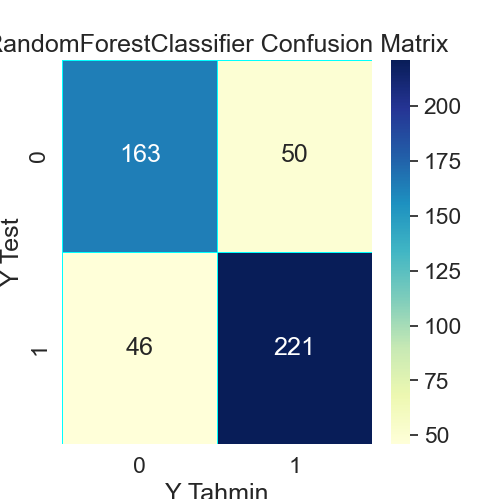








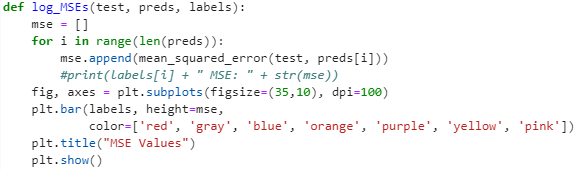
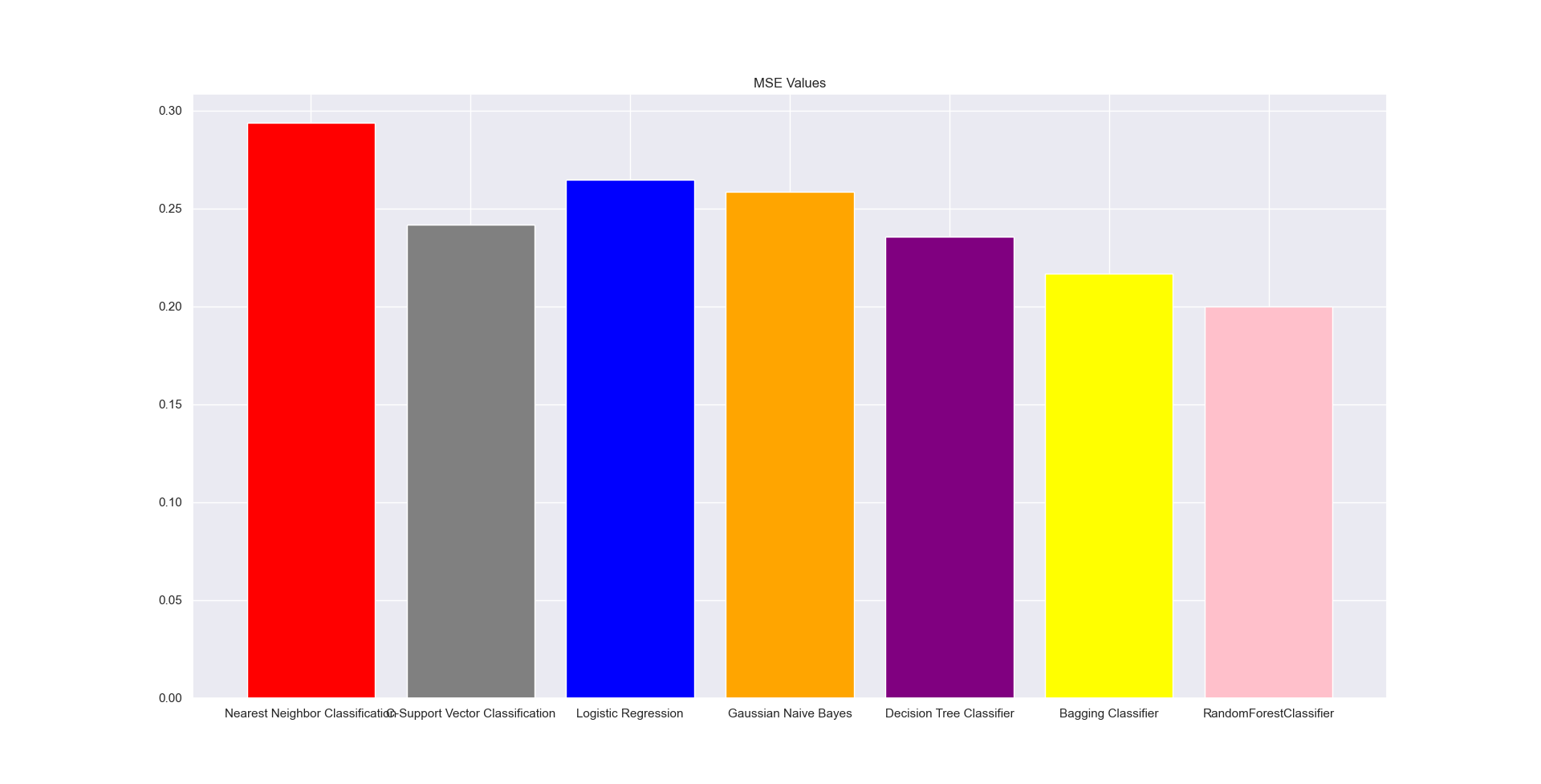
**Şekil 8 - Confusion Matrix Sonuçları**



* + 1. **MSE (Mean Squared Error)**

MSE, oluşturulan modelin, tahminleyicinin performansını ölçer. Pozitif ve 0'a yakın bir değer olmalıdır. [12]

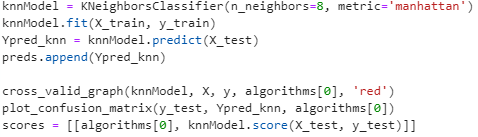
Aşağıda işlem yaptığımız algoritmaların MSE kodu ve değerleri verilmiştir.

  
**Şekil 9 - MSE Kod Bloğu  
Şekil 10 - MSE Çıktıları**

1. **Nearest Neighbor Classification**

En yakın komşu olarak bilinen KNN algoritması gözetimli öğrenme metodlarından birisidir. Basit sınıflandırma işleminde kullanılıp, çıktısı bir sınıf üyeliğidir. Nesne, en yakın komşuları arasında en yaygın olan sınıfa verilir.[5]

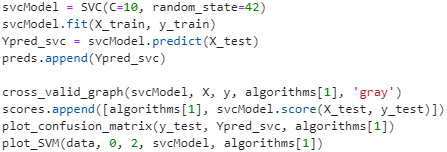
KNN algoritmasının kod bloğu aşağıdaki gibidir.

  
**Şekil 11 - KNN Algoritması Gösterimi**

1. **Support Vector Classification**

Destek vektör algoritması olarak bilinen bir algoritmadır. İki sınıfı bir doğru veya düzlem ile birbirinden ayırma işlemi yapmaya çalışırken sınırındaki elemanları baz alır. Alanındaki bir çok yönteme göre daha yüksek başarı verir.[6]

SVM algoritmasının kod bloğu ve çıktısı aşağıdaki gibidir.

  
**Şekil 12 - SVM Algoritması Gösterimi**

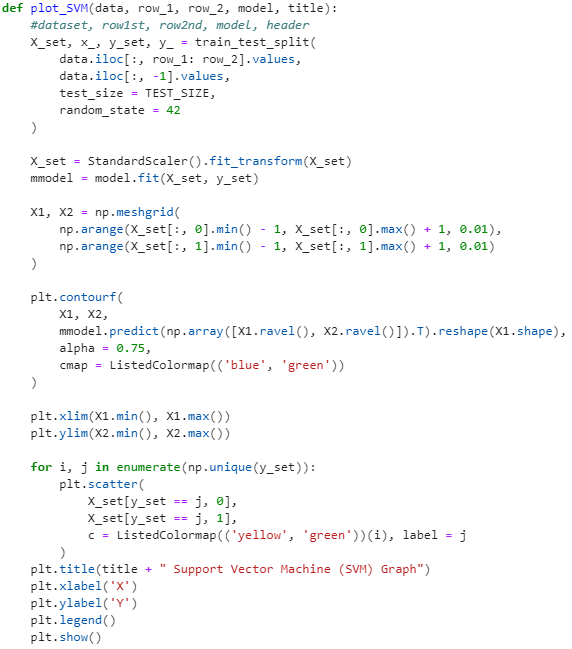
Kırmızı şarap kalitesini içeren veri setini farklı algoritmalarla eğittik. Bu algoritmalardan birisi de SVC algoritmasıydı. Veri setinin iki sütununu örnek olarak alıp ‘Destek Vektör Makinesi Grafiği’nin plot görselini çizdireceğiz.

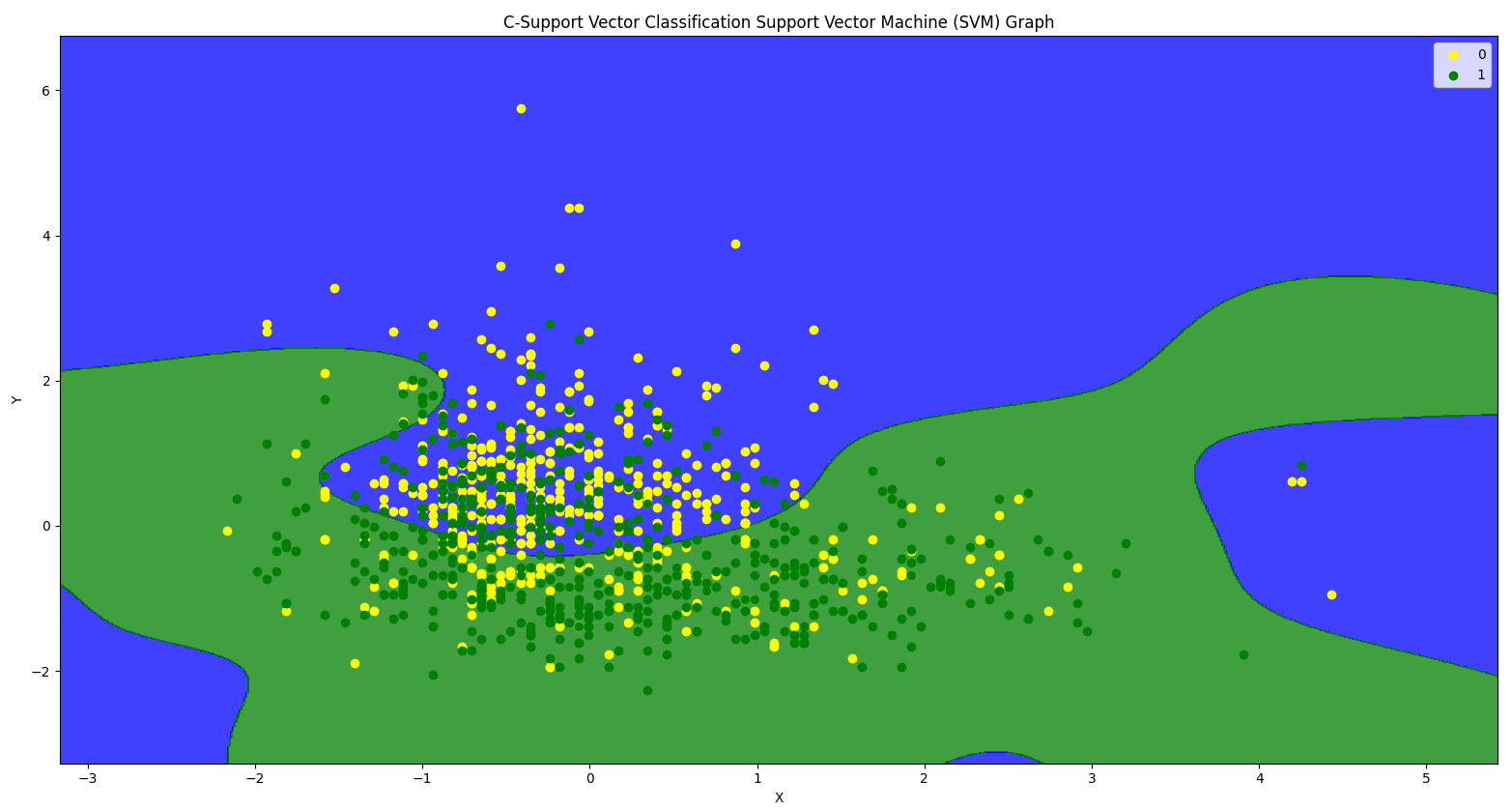
Öncelikle verimizin eğitimini tekrar yapmamız gerekiyor. Çünkü SVM grafiği, 2 adet X sütununu destekliyor. Bu yüzden istediğimiz iki sütunu ‘row’ değişkenleriyle belirtip, train\_test\_split fonksiyonuna gönderiyoruz.

Bu sayede yeni X, y değişkenlerimizin test, train verilerini almış oluyoruz. Daha sonra 2 sütunlu X verimizi yeniden boyutlandırıp (StandartScaler) eğitim sonucunu alıyoruz. Böylece SVM grafiğini çizdirmemiz için gerekli olan veriler elimize geçmiş oluyor.

İki adet olarak aldığımız X değerini ‘meshgrid’ kullanarak ikiye ayırdık. Daha sonra grafiğimizi oluşturmak için plt figürünün içine yolladık. Figürün fiziksel özelliklerini de tanımladıktan sonra ‘scatter’ kullanarak, noktalı veriye göre çizgi çekilmesini sağladık. Daha sonra figürümüzü ekrana çıkarttık.

Kod bloğu aşağıdaki gibidir.

**  
Şekil 13 - SVM Grafiği Kod Bloğu**

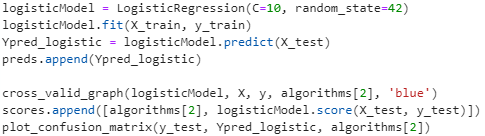
****

**Şekil 14 - SVM Grafiği**

1. **Logistic Regression**

Lojistik Regresyon, istatistiksel bir yöntemdir. Bir veya birden fazla bağımsız değişken bulunduran veri kümesini analiz etmek için kullanılır. [7]

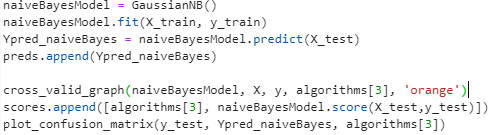
Lojistik regresyonun kod bloğu aşağıdaki gibidir.

  
**Şekil 15 - Lojistik Regresyon Kod Bloğu**

1. **Naive Bayes**

Tembel bir öğrenme algoritması olup Bayes teoremine dayanır. Elimizde bulunan örneklerin hangi kategoriye ait olduğunu elde ettiği oranla tahmin eder. Aynı zamanda dengesiz veri kümelerinde de çalışabilir.[8]

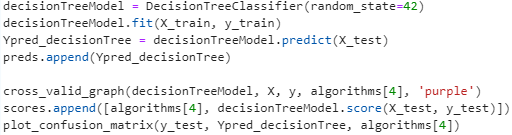
Navie Bayes algoritması kod bloğu aşağıdaki gibidir.

  
**Şekil 16 - Naive Bayes Kod Bloğu**

1. **Decision Tree**

Yaygın olarak kullanılan gözetimli öğrenme algoritmalarından biridir. Karar düğümleri ve yaprak düğümlerinden oluşur. Ağaç dalı hiyerarşisi gibi dallanır.[9]

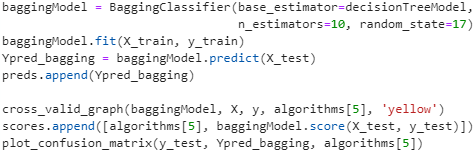
Decision Tree algoritması kod bloğu aşağıdaki gibidir.

  
**Şekil 17 - Decision Tree Kod Bloğu**

1. **Bagging**

Var olan bir eğitim setinden yeni eğitim setleri türeterek temel öğrenciyi yeniden eğitmeyi amaçlayan bir yöntemdir.[10]

Bagging algoritması kod bloğu aşağıdaki gibidir.

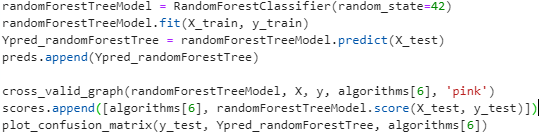
****

**Şekil 18 - Bagging Kod Bloğu**

1. **Random Forest Tree**

Denetimli bir algoritma olan Random Forest Tree, diğer algoritmalara göre daha yüksek doğruluk oranı verir. Hem sınıflandırma hem de regresyon analizi için uygulanabilirdir. Bu algoritma rastgele bir orman yaratıyor. Ağaç sayısı arttıkça doğruluk oranıda artar.[11]

Random Forest algoritması kod bloğu aşağıdaki gibidir.

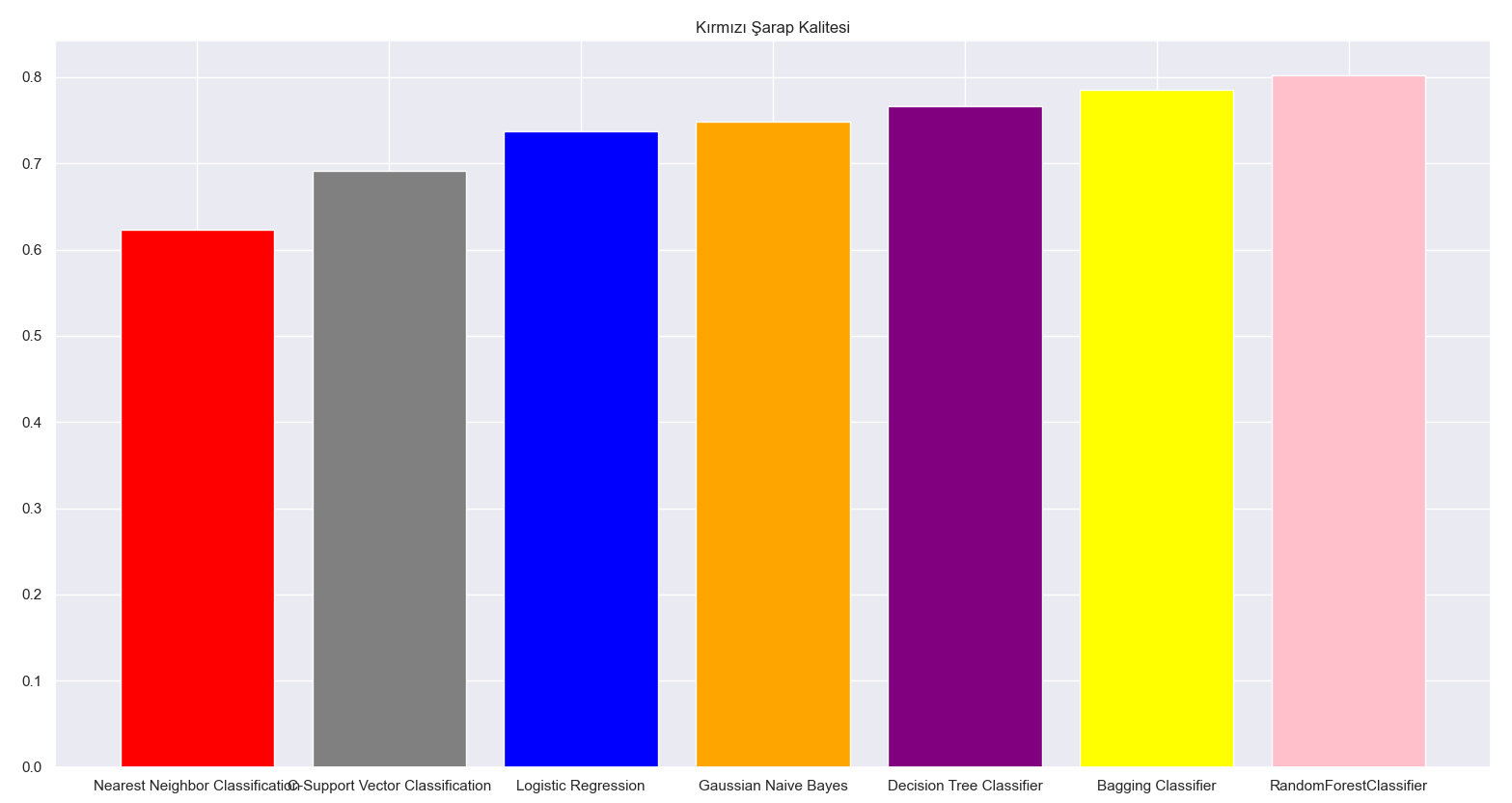
  
**Şekil 19 - Random Forest Tree Kod Bloğu**

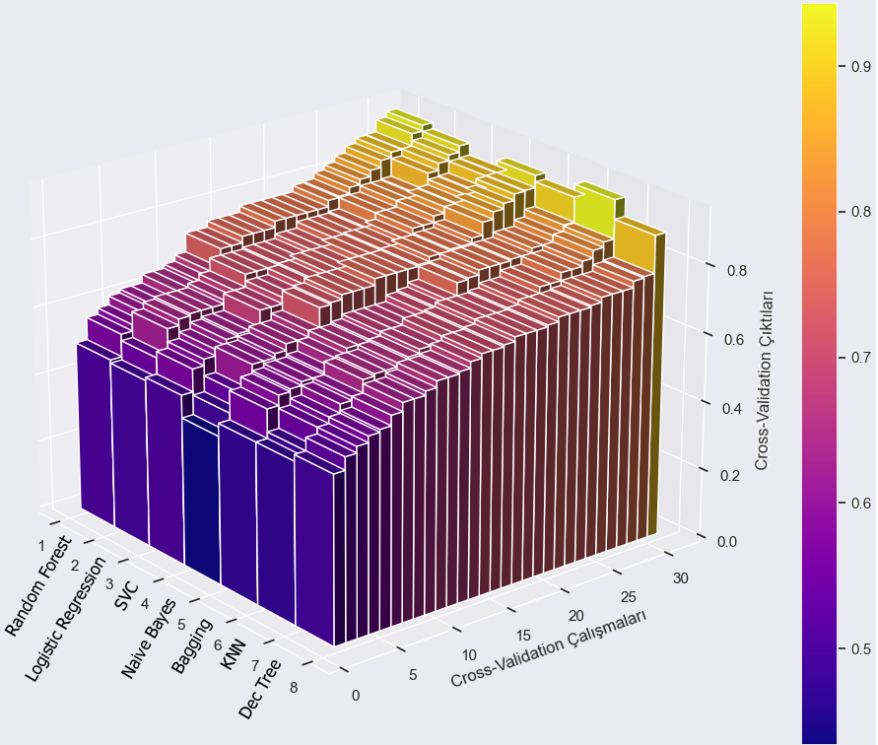
1. **SONUÇ**

Red Wine Quality adında bir veri seti seçtik. Kurduğumuz yapay sinir ağlarında modellemesini, eğitimini gerçekleştirdik. Daha sonra eğitim sonuçlarını, ROC eğrisini, confusion matrix’ini ve MSE değerini grafiksel sonuç olarak elde ettik.

Çalışmamızın diğer kısmında birden çok algoritma kullandık. Kullandığımız algoritmaları modelleyip, eğititik. Bu eğitim sonucunda tüm algoritmaların sonuçlarını histogram grafiğine aktardık. Histogram grafiği aşağıda verilmiştir.

* **Kırmızı:** Nearest Neighbor Classification
* **Gri:**C-Support Vector Classification
* **Mavi:** Logistic Regression
* **Turuncu:** Gaussian Naive Bayes
* **Mor:** Decision Tree Classifier
* **Sarı:** Bagging Classifier
* **Pembe:** RandomForestClassifier

  
**Şekil 20 - Final Sonuç Grafiği (Accuracy)**



**Şekil 21 – Tüm Algoritmaların Cross Validation Sonuçlarının 3D Grafiği**

**KAYNAKLAR**

[1]<https://azure.microsoft.com/tr-tr/overview/what-is-artificial-intelligence/#how>

[2] <https://tr.wikipedia.org/wiki/Yapay_zek%C3%A2#Tarih%C3%A7e>

[3] <https://www.leyastat.com/roc-analizi-ve-roc-egrisi/>

[4]<https://www.veribilimiokulu.com/hata-matrisini-confusion-matrix-yorumlama/>

[5]<https://veribilimcisi.com/2017/07/20/k-en-yakin-komsu-k-nearest-neighborsknn/>

[6]<https://www.veribilimiokulu.com/support-vector-machine-svm-ile-siniflandirma-python-ornek-uygulamasi/>

[7] <https://veribilimcisi.com/2017/07/18/lojistik-regresyon/>

[8]<https://medium.com/@ekrem.hatipoglu/machine-learning-classification-naive-bayes-part-11-4a10cd3452b4>

[9]<https://erdincuzun.com/makine_ogrenmesi/decision-tree-karar-agaci-id3-algoritmasi-classification-siniflama>

[10]<https://kadirguzel.medium.com/kolektif-%C3%B6%C4%9Frenme-ve-bagging-algoritmas%C4%B1-e8ea3d932b72>

[11]<https://medium.com/data-science-tr/makine-%C3%B6%C4%9Frenmesi-dersleri-5-bagging-ve-random-forest-2f803cf21e07>

[12]<https://veribilimcisi.com/2017/07/14/mse-rmse-mae-mape-metrikleri-nedir/>