**KARABÜK ÜNİVERSİTESİ**

**LİSANSÜSTÜ EĞİTİM ENSTİTÜSÜ**

**BİYOMEDİKAL MÜHENDİSLİĞİ**



**BİYOMEDİKAL MÜHENDİSLİĞİNDE YAPAY SİNİR AĞI UYGULAMALARI**

**FİNAL RAPORU**

**ADI: ESRA**

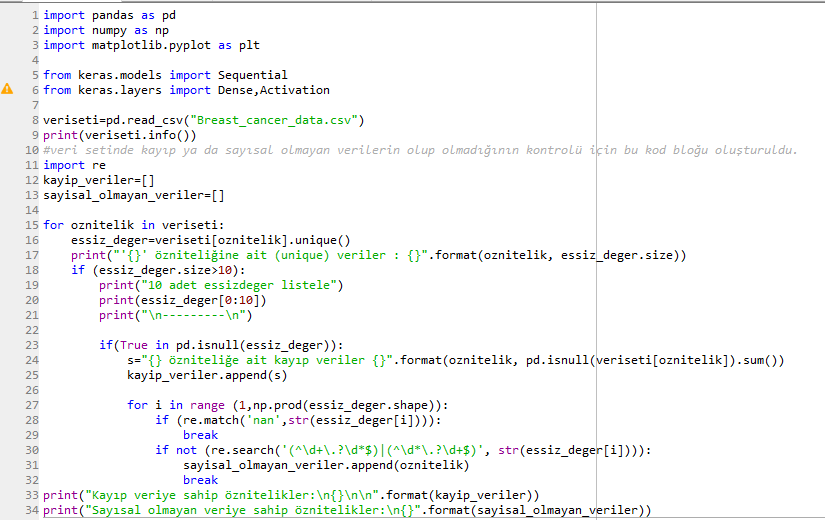
**SOYADI: ALGIM**

**NO: 1928142005**

Breast\_cancer\_data.csv adlı veri setini Keras ve MLP yapay sinir ağı kütüphanelerini kullanarak performans değerlendirmelerinin yapılması amaçlanmıştır.

**KERAS UYGULAMA 1**

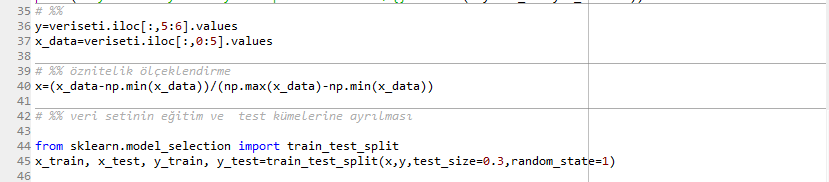
Öncelikle **veriseti** isimli DataFrame nesnesi pandas kütüphanesi yardımıyla oluşturulmuştur. Breast\_cancer\_data.csv isimli dosya 5 adet girdi özniteliği ve diagnosis isimli bir çıktı özniteliğine sahip 570 satırdan oluşmaktadır. Diagnosis özniteliği 0 ve 1 çıktılarını vermektedir. Bu data setinde meme kanseri teşhisi tahmini yapılacaktır.



Liste 1

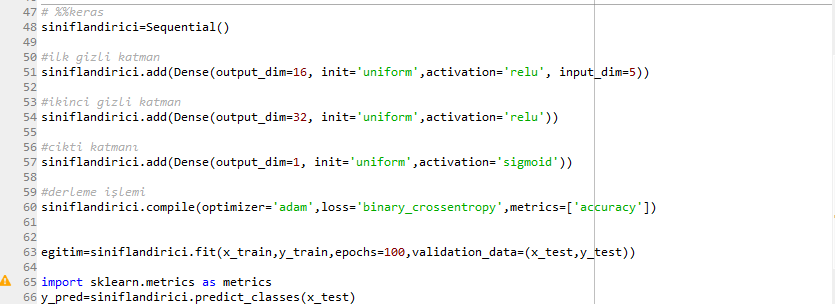
Liste 1’de öncelikle veriseti adlı dataframe içerisinden bağımsız değişken (X) ve bağımlı değişken (y) öznitelikleri, Breast\_cancer\_data.csv dosyasındaki isimleri kullanılarak oluşturulmuştur.

Veri setinde kayıp ya da sayısal olmayan verilerin olup olmadığının kontrolü için Liste 1’deki kod bloğu oluşturulmuştur. Kod bloğu veri setini daha iyi tanımak için yararlı olacaktır. Bu kod bloğunda, her özniteliğin içerdiği eşsiz(unique) değerler görülebilmektedir. Eşsiz değerin çok olması durumu dikkate alınarak eğer 10’dan fazla eşsiz değer var ise ekrana yazdırılmaktadır. Kodda kullanıla **re** (regular expression) kütüphanesi sayısal olmayan verilerin araştırılmasında kullanılmıştır. Ekran çıktısı incelendiğinde kayıp veri olmadığı görülmüştür. Veri ön işleme yapılmıştır.



Liste 2

Liste 2’ de veri setinin eğitim ve test kümelerine ayrılması ve öznitelik ölçeklendirme için uygulanan kodlar görülmektedir. Veri setinin %70’i eğitim %30’u test için ayrılmıştır. Veri ön işleme adımlarından olan öznitelik ölçeklendirme işlemi veri setindeki tüm değerlerin 0 ile 1 arasında değer alan bir dönüşüme uğratmıştır.



Liste 3

Liste 1’de Keras kütüphanesi import edilmiştir. 5 girdi ve 1 çıktıdan oluşan yapay sinir ağı modeli, iki gizli katman kullanılarak oluşturulmuştur. İlk gizli katmanda 16, ikinci 32 adet yapay nöron kullanılmıştır. Liste 3’te görülen kodlar yapay sinir ağı modelini oluşturmaktadır.

**add()** fonksiyonu yapay sinir ağı katmaları oluşturulmasında kullanılır. Gizli katman sayısı bu fonksiyon ile arttırılabilir.

**output\_dim** parametresi katmandaki yapay nöron sayısını ifade eder.

**input\_dim** parametresi girdi özniteliklerinin sayısını belirtmeye yarar.

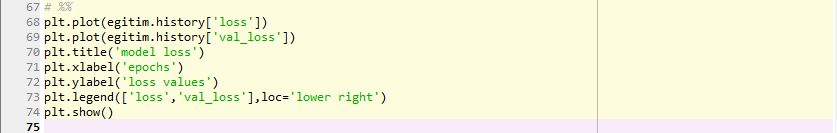
**İnit** parametresi ise başlangıç ağırlık değerlerinin belirlenmesi için alternatifler sunar.

**compile()** metodu öğrenme sürecini yapılandırmak için kullanılır.

**optimizer** parametresi kullanılan en iyileme algoritmasının belirlemesini sağlar.

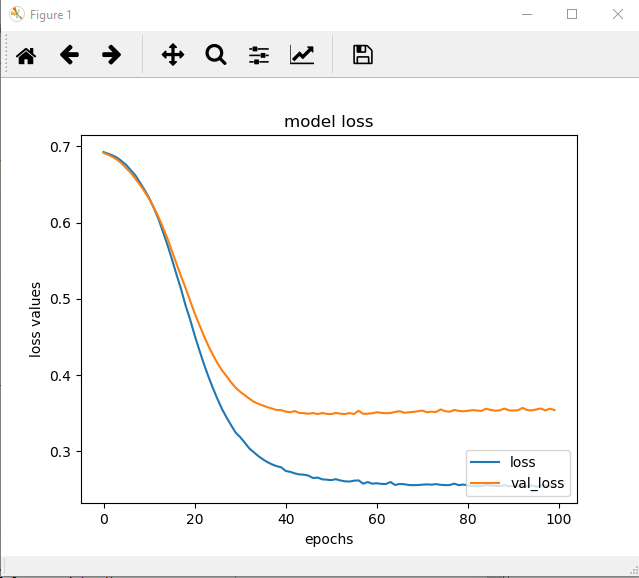
**metrics** parametresi model performansının değerlendirilmesinde kullanılan değerler alır.

**loss** parametresi hata fonksiyonlarının tanımlanmasında kullanılır.

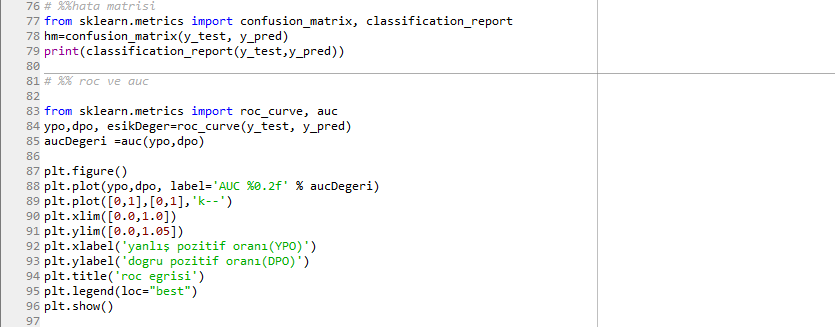


Liste 4

Liste 4’te parametrelerin grafiği çizdirilmiştir.



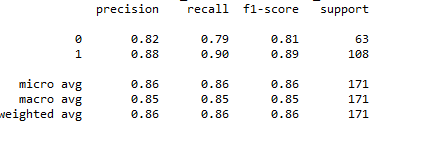
Şekil 1



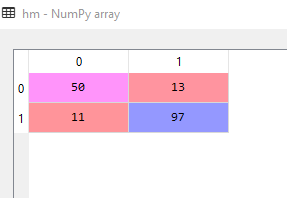
Liste 5

Tablo 1

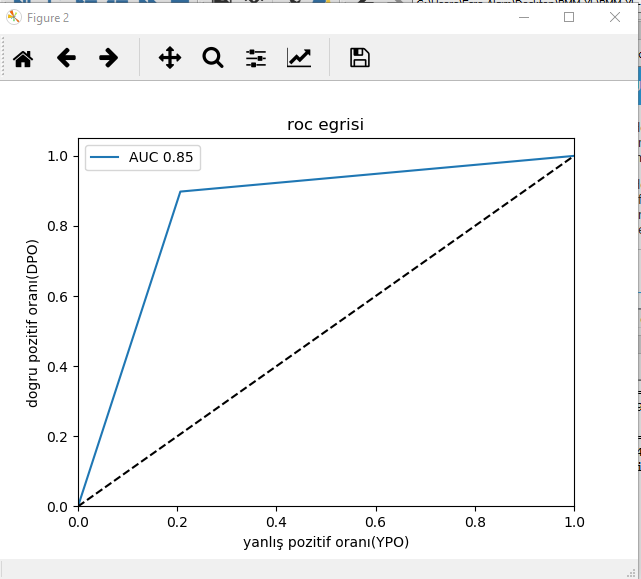
confusion\_matrix() fonksiyonu ile oluşturulan hata matrisi hm adlı listeye atanmıştır. Elde edilen hata matrisine ait performans değerlendirme ölçütleri kesinlik, duyarlılık ve f1 skorlama değerlerini içeren sonuçlar listelenmiştir.



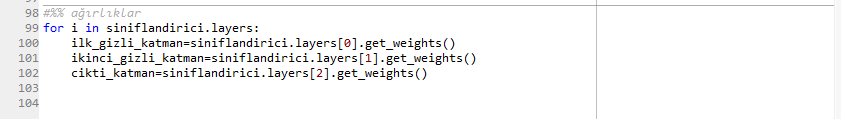
Hata matrisi doğru pozitif, doğru negatif, yanlış pozitif ve yanlış negatif olmak üzere dört kısımdan meydana gelir.



Doğru pozitif ve doğru negatif örneklerin sayısının çok olması sınıflandırma performansı için olumlu yönde etki yapacaktır. Dengeli dağılıma sahip olmayan veri setleri için ROC eğrisi kullanılarak performans değerlendirmesi yapılabilir. Doğru pozitif oranı ve yanlış pozitif oranı ROC eğrisini oluşturan iki temel unsurdur. Doğru pozitif oranının yüksek, yanlış pozitif oranının düşük olması performans için olumlu olarak nitelendirilir. ROC eğrisinin altında kalan alanın 1’e yakın olması sınıflandırma performansı açısından hedeflenen bir sonuçtur. AUC değeri 0 ile 1 arasında değişir arzulanan değeri ise 1’e olabildiğince yakın olmasıdır.



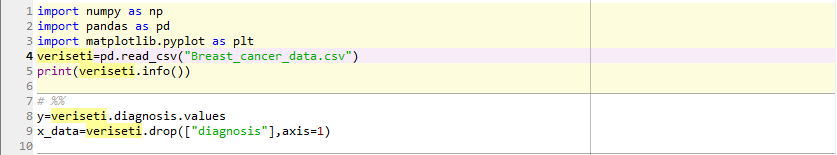
Şekil 2



Liste 6

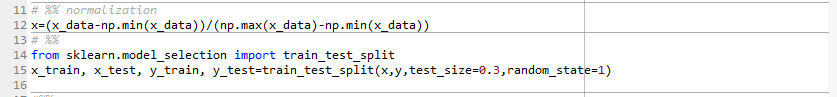
Bu kod bloğu ile üç adet ağırlık matrisi oluşmaktadır. ilk\_gizli\_katman listesinde 5\*6 boyutunda bir liste görülmektedir. Burada 5 değeri girdileri 6 değeri ise 1. Gizli katmandaki yapay nöron sayısını vermektedir. 6\*1 boyutlu liste ise bias’a ağırlık değerlerini vermektedir. Ağırlık değerleri modeli oluşturan denklemin katsayılarını oluşturmaktadır.

**MLP UYGULAMA 1**



Liste 7

Öncelikle **veriseti** isimli DataFrame nesnesi pandas kütüphanesi yardımıyla oluşturulmuştur.



Liste 8

Liste 8’ de veri setinin eğitim ve test kümelerine ayrılması ve öznitelik ölçeklendirme için uygulanan kodlar görülmektedir. Veri setinin %70’i eğitim %30’u test için ayrılmıştır. Veri ön işleme adımlarından olan öznitelik ölçeklendirme işlemi olan normalizasyon yapılmıştır.

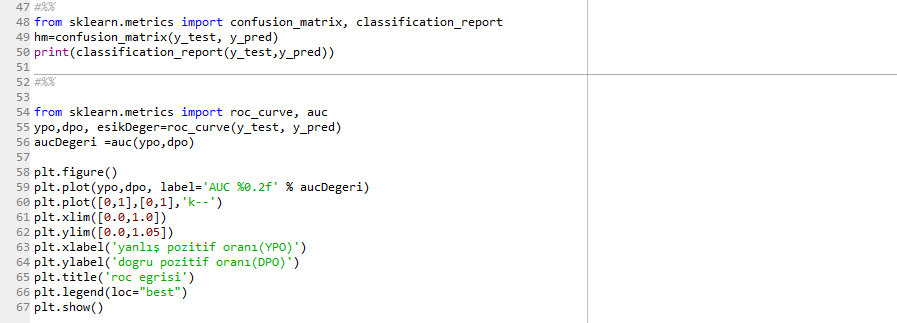


Liste 9

Liste 9’daki kod bloğunda yapay sinir ağı modeli çok katmanlı perceptron ile oluşturulmuş. Grid search ile geliştirilmiştir. MLPClassifier, her adımda kayıp fonksiyonunun model parametreleri ile ilgili kısmi türevlerinin parametreleri güncelleştirmek için hesaplandığından yinelemeli olarak çalışır.

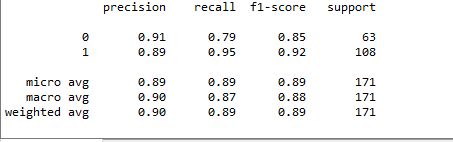
**hidden\_layer\_sizes parametresi** gizli tabakadaki nöronların sayısını temsil eder.

**alpha** normalleştirme terimi parametresidir.

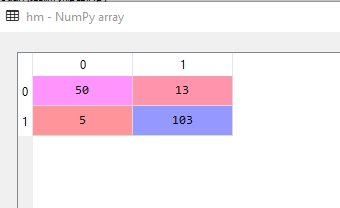


Liste 10

Keras uygulama 1’de uygulanan hata matrisi oluşturma ve roc eğrisi oluşturma kodları aynı şekilde yazılmıştır.

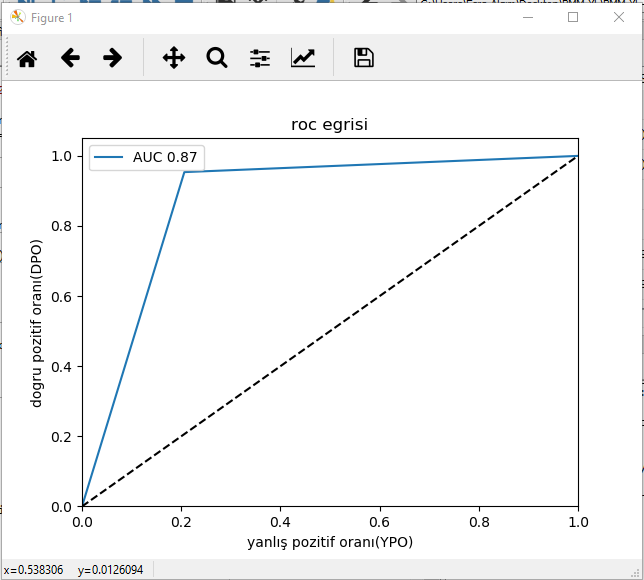


Elde edilen hata matrisine ait performans değerlendirme ölçütleri kesinlik, duyarlılık ve f1 skorlama değerlerini içeren sonuçlar listelenmiştir.



Tablo 2

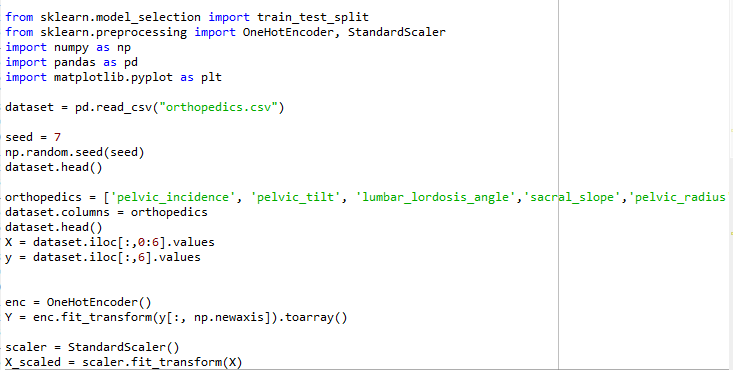
Hata matrisi doğru pozitif, doğru negatif, yanlış pozitif ve yanlış negatif olmak üzere dört kısımdan meydana gelir.

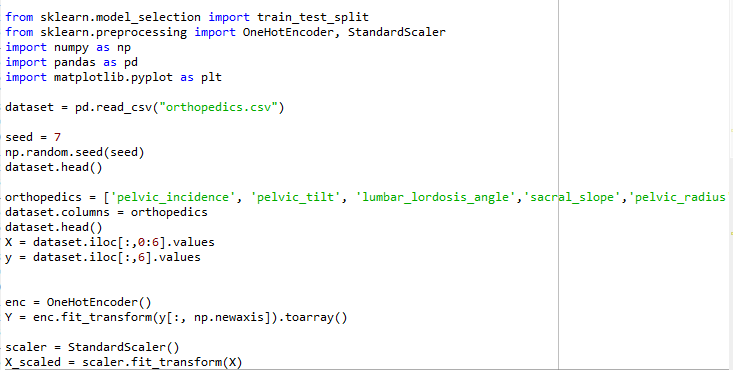


Şekil 3

Keras ile elde edilen auc 0.85 değeri ile Mlp de elde edilen auc 0.87 değerleri birbirine oldukça yakındır. Mlp’de uygulanan Grid Search Cross Validation’ın modeli geliştirdiği düşünülmektedir.

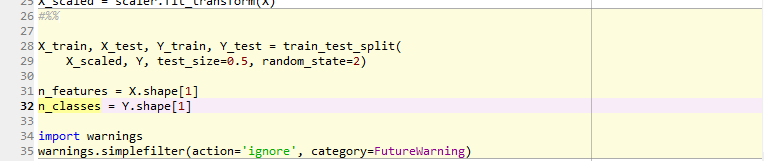
**KERAS UYGULAMA 2**

Öncelikle **veriseti** isimli DataFrame nesnesi pandas kütüphanesi yardımıyla oluşturulmuştur. orthopedics.csv isimli dosya 6 adet girdi özniteliği ve diagnosis isimli bir çıktı özniteliğine sahip 310 satırdan oluşmaktadır. Diagnosis özniteliği Hernia , Spondylolisthesis ve Normal isimli 3 çıktı vermektedir. Bu data setinde ortopedik hastalık olan Hernia ve Spondylolisthesis teşhisi tahmini yapılacaktır.



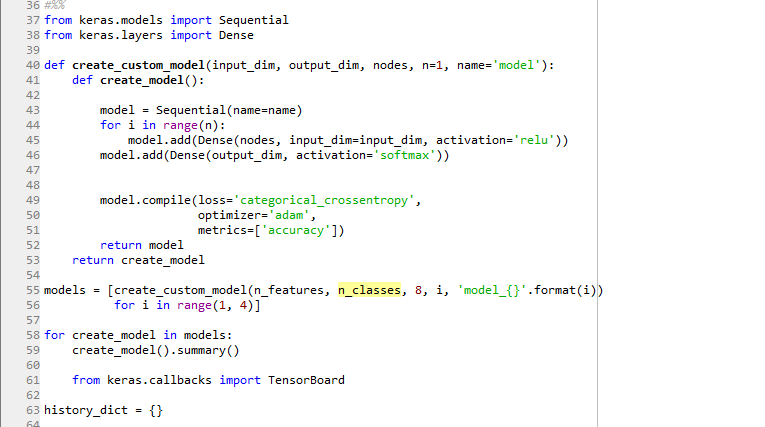
Liste 11

Veri setinde diagnosis özniteliği sayısal olmayan 3 çıktı içermektedir. Veri hazırlama aşaması olan OneHotEncoder ile sınıflar arasındaki ilişkiye değil onların varlık yokluklarına odaklanır, yapay sinir ağı sistemi için tercih edilebilecek bir veri ön işleme sistemidir.



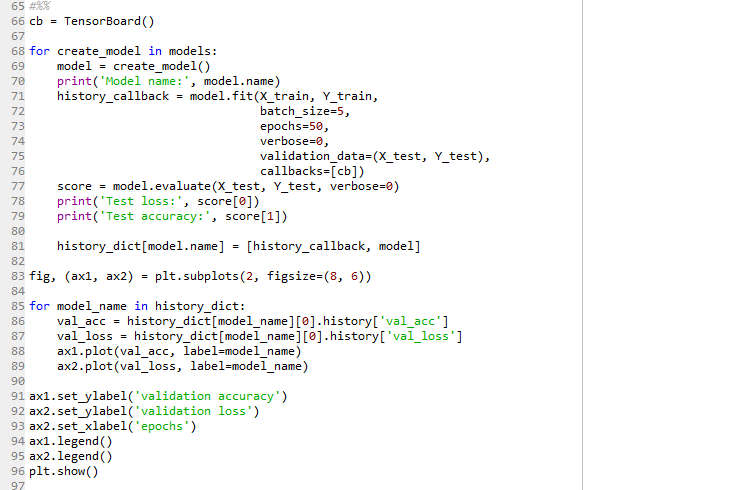
Liste 12

Liste 12’ de veri setinin eğitim ve test kümelerine ayrılması ve öznitelik ölçeklendirme için uygulanan kodlar görülmektedir. N\_features ve n\_ classes shape ile farklı boyutlara dönüştürülmüştür.



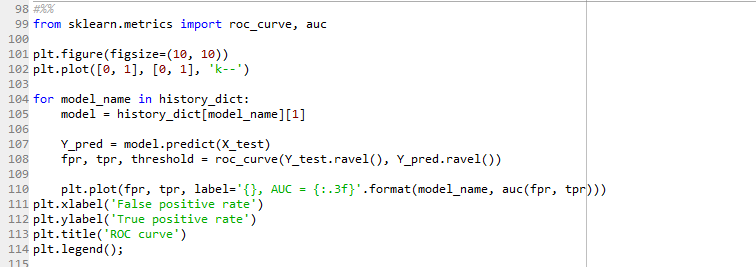
Liste 13

Liste 13 keras ile yapay sinir ağı uygulamasına başlandığı kod bloğudur. Create\_custom\_model ve create\_model adlı fonksiyonlar oluşturulmuştur. Keras for döngüsü içerisine alınmış n kadar döndürülerek yapay sinir ağı modeli oluşturulmuştur.



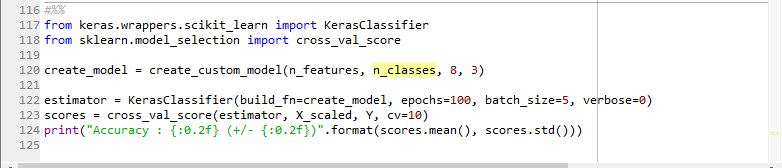
Liste 14

Liste 14’te model eğitimi yapılmıştır. TensorBoard geri çağırılma olarak kullanılmıştır. TensorBoard TensorFlow ile sağlanan bir görselleştirme aracıdır.



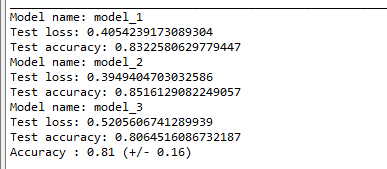
Liste 15

Liste 15 ile ROC eğrisi çizilmiştir.

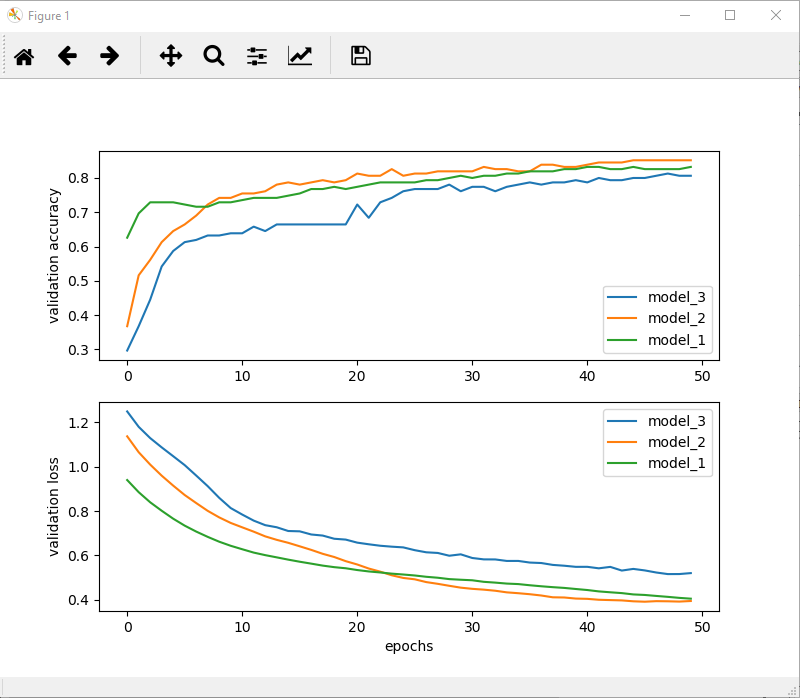


Liste 16

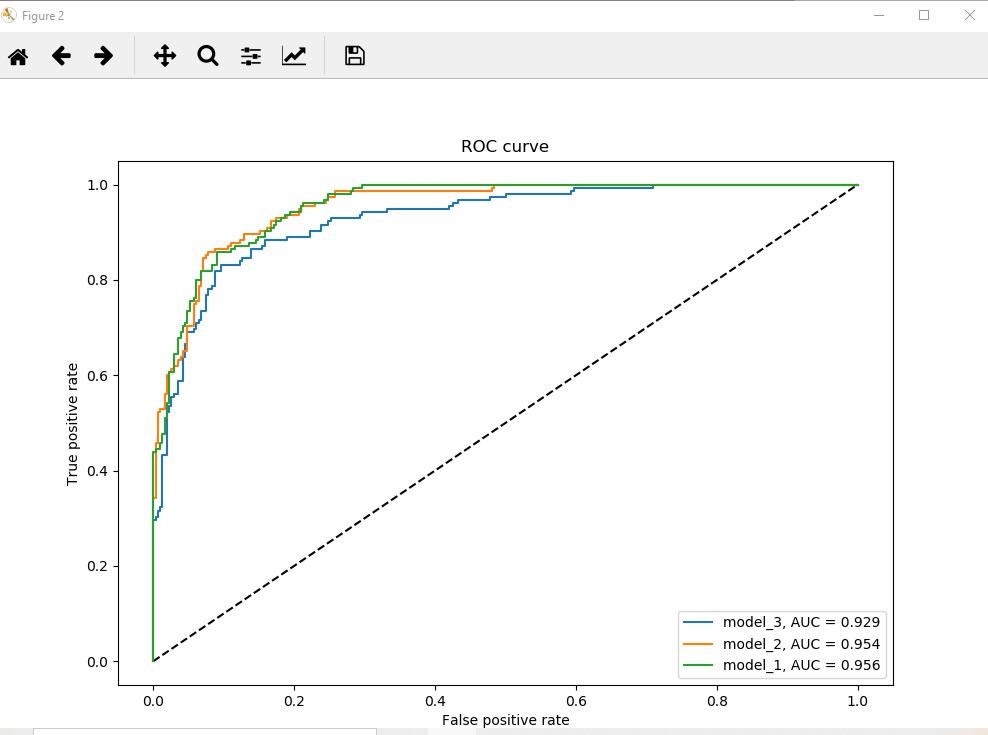
10 kat çapraz doğrulama ile performans ölçümü yapılmıştır.



Oluşturulan modelin doğruluk değerleri yukarıdaki gibidir.



Şekil 4



Şekil 5

ROC grafiği yukarıdaki gibi olan modelin oluşturulan model1 :0.956, model2: 0.954, model 3: 0.929 olarak bulunmuştur.