****

**GİRİŞ-HAZIRLIK**

**1.1 Arayüz**

Projede arayüz olarak Orange programı kullanılmıştır. Orange, görsel programlama arayüzü ve veri analizi için açık kaynaklı bir araçtır ve özellikle yapay zeka projelerinde kullanılmak üzere tasarlanmıştır. Orange kullanılmasının başlıca sebepleri şunlardır:

**Veri Görselleştirmesi:** Orange, veri setlerini görselleştirmek ve anlamak için çeşitli araçlar sunar. Bu, verilerin daha iyi anlaşılmasına ve model oluştururken daha bilinçli kararlar alınmasına yardımcı olmuştur.

**Hızlı Prototipleme:** Görsel programlama ve modüler yapısı sayesinde, Orange sayesinde hızlı prototipler oluşturabilir ve farklı modelleri kolayca denenebilir. Bu, projelerin hızla geliştirilmesine olanak tanır. Bu yüzden projede Orange kullanılmıştır.

Orange hakkında daha detaylı bilgi için :

* <https://orangedatamining.com/>

**1.2 Veri seti**

Kaggle internet sitesinden detaylı araştırma sonucunda,farklı cinsiyet ve milliyetletden oluşan, her yaştan insanlar içeren,aynı boyutlarda(200,200),görsel veri seti alınmıştır.

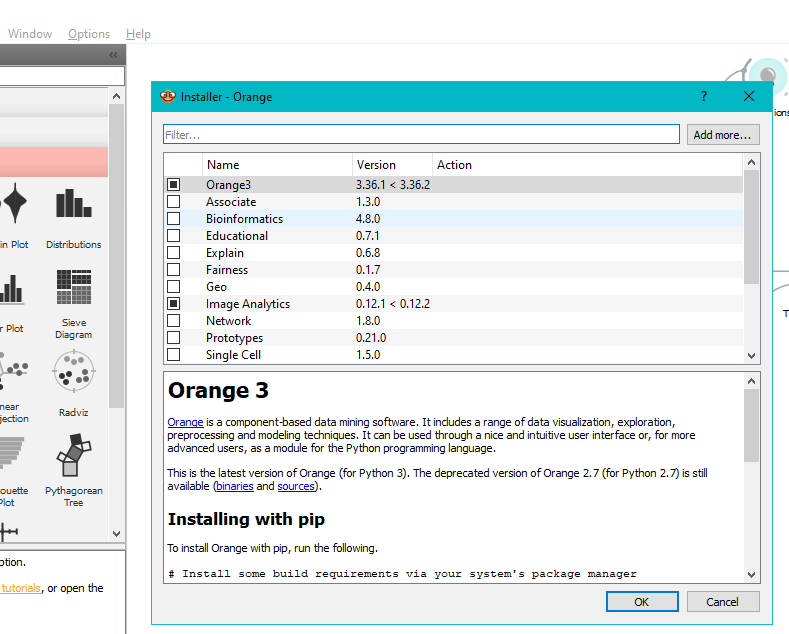
Veri seti seçilirken fotoğrafların boyutlarının aynı olmasına ve fotoğrafların konuya uygun konseptte olmasına dikkat edilmiştir. Veri setinin internet Linki aşağıya bırakılmıştır:

* <https://www.kaggle.com/datasets/shanmukh05/agedetection>

**PROJENİN SON HALİ**

**2.1 Görüntü İşleme(image processing)**

Orange programına zorunlu olarak,’Options -> Add-ons’ seçeneğinden görüntü işleme için eklenti olarak “Image Analytics” kütüphanesi eklenmiştir.

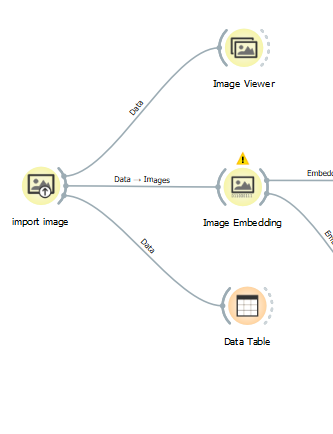


Şekiş 2.1.1

Image analyticsten elde ettiğimiz eklentilerle ve fonksiyonlarla görüntü işleme işlemleri yapılmıştır.İlk önce veri setini import etmek için “Image Import” düğümü eklenmiştir. Daha sonra import edilen fotoğrafları önce “Image Viewer” ile görüntüleyip daha sonra “Image Embedding” düğümü ile görüntüler sayısal bir vektör formatında kodlara dönüştürülmüştür.Son olarak “Data table” düğümü ile verilerin kategorilerini görmemizi ve istediğimiz şekilde sırayabilmemizi sağlamaktadır.

Image Embedding kısmında farklı denemeler sonucunda embedder olarak “openface” kullanılmıştır.OpenFace, yüzleri algılamak, tanımak ve analiz etmek için derin öğrenme ağları kullanır. Yüzlerin çeşitli açılardan görüntülerini analiz eder, yüzler arasındaki benzersiz özellikleri (örneğin, gözlerin konumu, burun ve ağız şekli) ayırt etmeye odaklanır. Bu algoritma, yüz tanıma, yüz özelliklerini çıkarma, yüzler arasındaki benzerlikleri belirleme gibi görevleri gerçekleştirebilir.

Aşağıda işlemlerin bağlantısal görseli mecvuttur:



Şekil 2.1.2

**2.2 Artifical Neural Networks**

Model olarak “Neural Network” kullanılmıştır.Orange’de Neural Network modeli sol menüde “Models” seçeneğinden seçilmiştir.Neural Network, biyolojik sinir sisteminden esinlenerek tasarlanmış matematiksel modellerdir. Bu modeller, veri içindeki desenleri tanımak, ilişkileri öğrenmek ve kararlar vermek için kullanılır. Yapay sinir ağları, genellikle katmanlardan oluşur ve bu katmanlar içindeki yapay nöronlar, bilgiyi işler.

Neural Network modelinde bazı parametreler bulunur.Bunlar: Gizli katmandaki nöron sayısı,aktivasyon foknsiyonu,optimizer algoritma,Regularization değeri(learning rate) ve iterasyon sayısıdır.

**Gizli katmandaki nöron sayıları** (20,20) olarak ayarlanmıştır. Daha fazla yapılsaydı çok zaman alabilirdi ve işler kompleksleşebilirdi. Daha az yapıldığında ise gerekli hassasiyeti ve doğruluk oranlarını yakalayamadığımız farkedilmiştir.

**Aktivasyon fonkisyonu** olarak “ReLu” fonksiyonu kullanılmıştır.ReLu(Rectified Linear Unit), Görüntü işleme alanında yapay sinir ağlarında birçok avantajı nedeniyle sıklıkla tercih edilen bir aktivasyon fonksiyonudur.

Diğer aktivasyon fonksiyonlarına göre hesaplama açısından daha hızlıdır. Sıfırın altındaki değerler doğrudan sıfıra dönüşür, bu da hesaplama süresini azaltır.

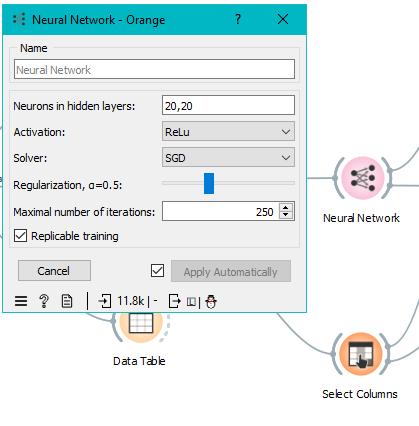
ReLu, 0'dan sınırsız bir çıktı aralığına sahiptir. Bu özelliği sayesinde, diğer aktivasyon fonksiyonlarına göre daha geniş bir çıktı aralığı sunarak ağın daha geniş bir öğrenme kapasitesine ulaşmasına yardımcı olabilir.

ReLu, Doğrusal olmayan bir aktivasyon fonksiyonudur. Bu, yapay sinir ağlarının daha karmaşık ve non-lineer ilişkileri öğrenmesine olanak tanır. Görüntüler genellikle non-lineer ilişkilere sahip olduğundan, bu tür fonksiyonlar görüntü işleme için daha etkilidir.

**Regularization değeri(learning rate)** 0.5 olarak ayarlanmıştır.Bu değer en uygun zamanda en uygun ‘accuracy’ değerini sağladığı için seçilmiştir.

**İterasyon sayısı** denemeler sonucunda 250 olarak belirlenmiştir.daha alt ve üst değerlerde accuracy değerinin optimalliği bozulmaktadır.

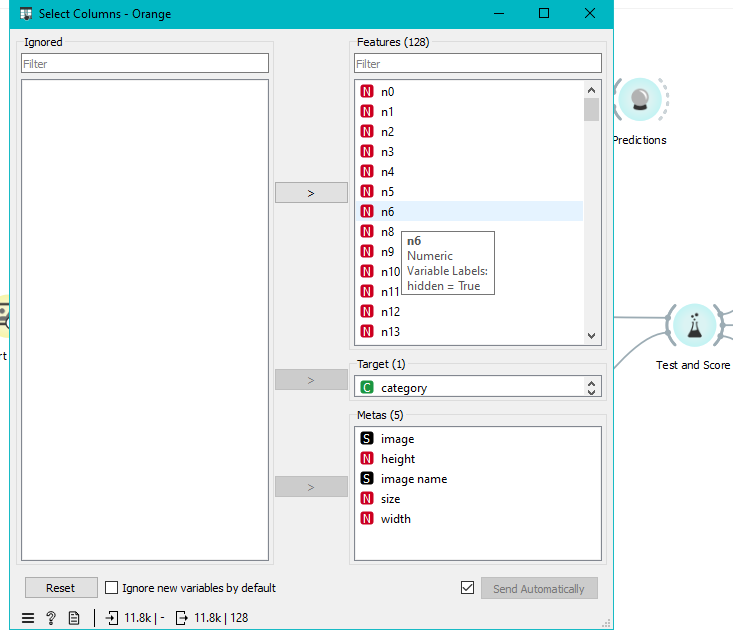
**Optimizer** olarak SGD(stochastic gradient descent) algoritması kullanılmıştır.



Şekil 2.2.1

**2.3 Select Columns**

‘Select Columns’ düğümünde Image Embedding ile oluşan verilerin özelliklerini(features) görmemizi ve sınıflandırmamızı sağlar.Düğüm tarafından 128 adet feature tespit edilmiştir.

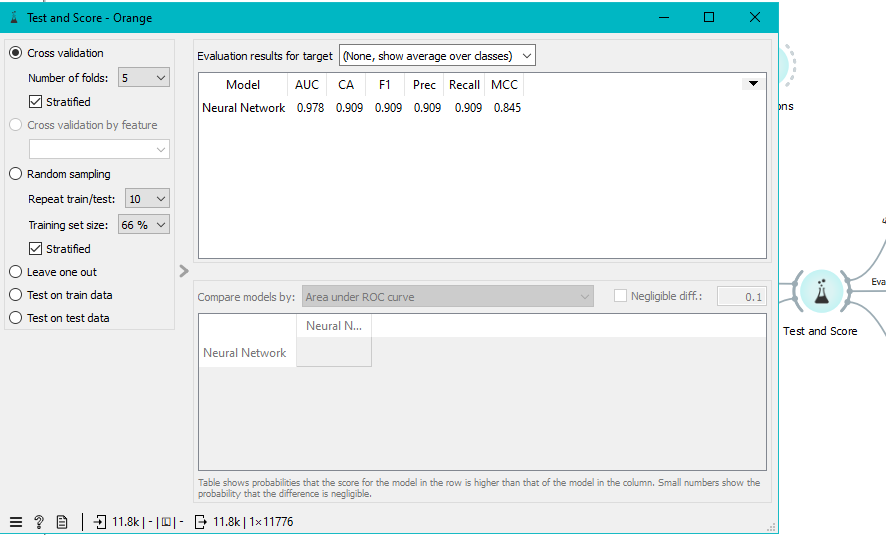


Şekil 2.3.1

**2.4 Test and Score**

‘Test and Score’ düğümü ile beraber otomatik olarak veri setimizin %66’lık kısmını öğrenme(train) için, kalan %34’lük kısmı da test için kullanılmıştır. Bu şekilde modelimiz aynı düğümde hem öğrenme hem test aşamalarını yapmış ve bize uygun olan sonuçları vermiş bulunmaktadır. Test and score düğümü bize classification accuracy,f1,precision,recall ve MCC değerlini göstermektedir.

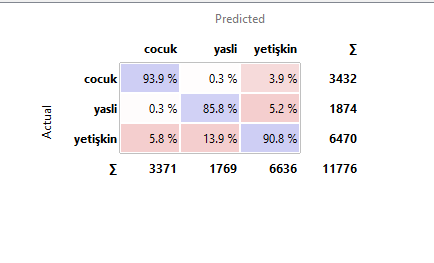
Aşağıda modelin son hali ve değerler mevcuttur:



Şekil 2.4.1

**2.5 Confusion Matrix**

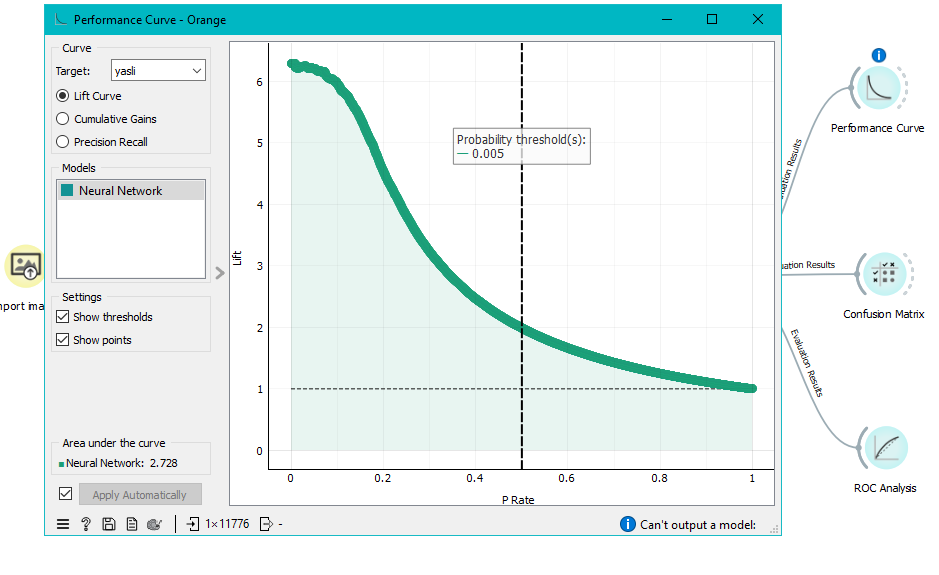
‘Confusion Matrix’ düğümü bize projemizde verilerin kategorisel olarak doğru tahmin edilen doğrularını, doğru tahmin edilen yanlışlarını, yanlış tahmin edilen doğrularını ve yanlış tahmin edilen yanlışlarını ve oranları ve sayılarını göstermektedir.Aşağıda oranlar ve sayılar mevcuttur.



Şekil 2.5.1

**2.6 Performans Curve**

‘Performans Curve’ düğümü modelin test sonuçlarının Lift Curve, Cumulative Gains ve Precision Recall grafiklerini gösterir.Bu grafikler ile modelin sonuçları analiz edilir.Grafiklere bakılarak model şekillendirilmiştir.Aşağıda örnek grafik ve sonuçları mevcuttur:



Şekil 2.6.1

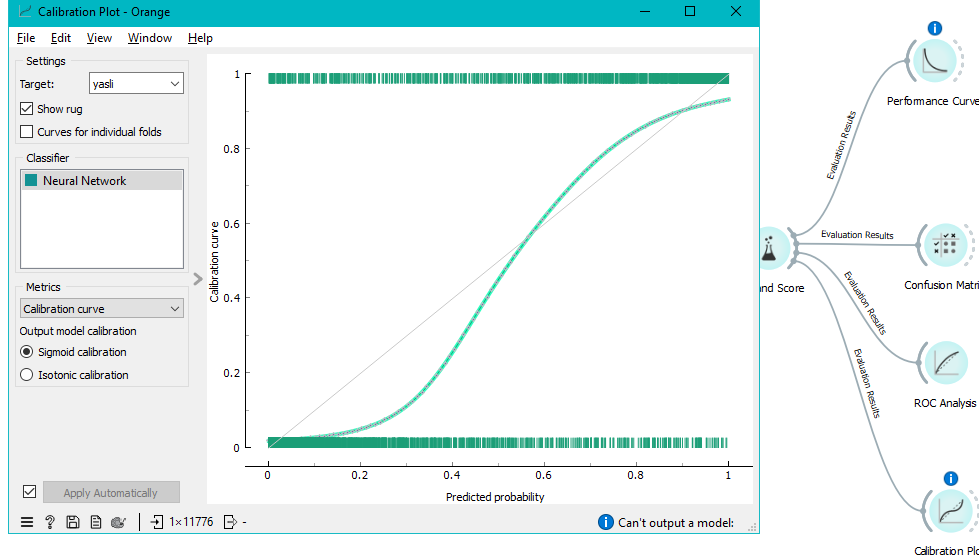
**2.7 Calibration Plot**

Bir yapay zeka modelinin çıktıları, öngörülen olasılıklar veya tahmin edilen değerler olabilir. Ancak, bu çıktılar gerçek dünyadaki durumlarla uyumlu olmayabilir veya güvenilirliklerini kaybedebilir. Bu nedenle, bu çıktıların doğruluğunu ve güvenilirliğini sağlamak için bir kalibrasyon adımı önemlidir. Bu adımı bize Calibration Plot düğümü sağlamaktadır.Aşağıda çocuk kategorisi için grafik mevcuttur:

metin, ekran görüntüsü, diyagram, öykü gelişim çizgisi; kumpas; grafiğini çıkarma içeren bir resim

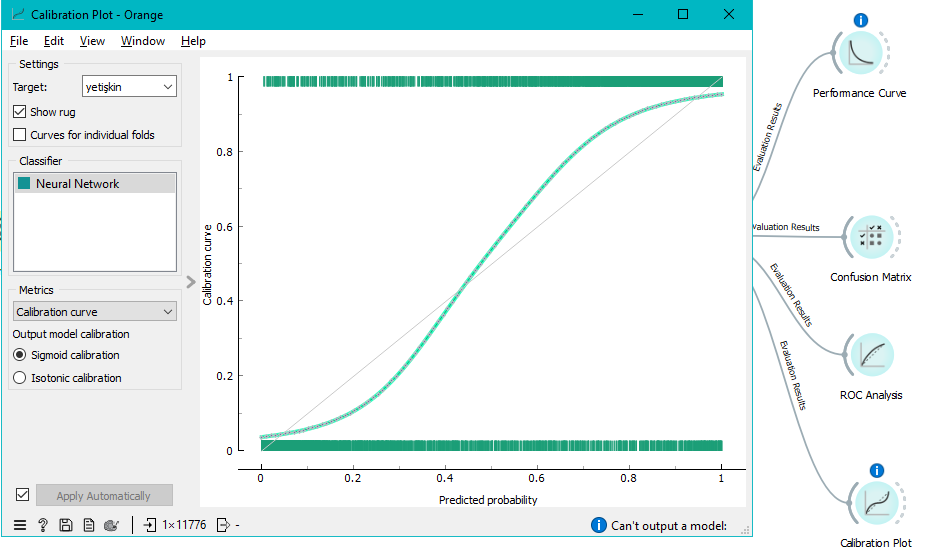
Açıklama otomatik olarak oluşturuldu

Yaşlı kategorisi için:



Şekil 2.7.2

Yetişkin kategorisi için:

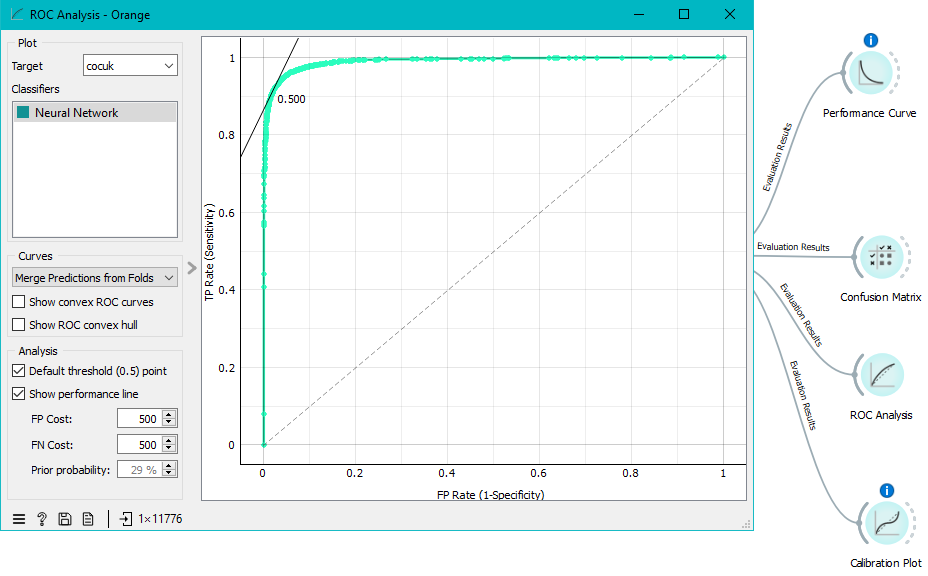


Şekil 2.7.3

**2.8 ROC Analysis**

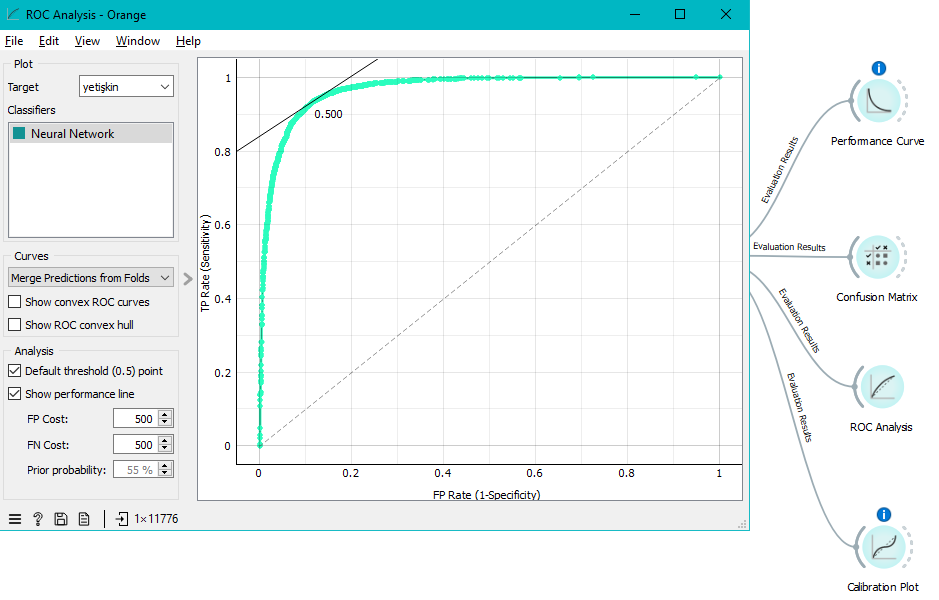
ROC eğrisi, duyarlılık (sensitivite) ve özgüllük (specificity) arasındaki ilişkiyi gösterir. Bu eğri, modelin farklı eşik değerlerinde (threshold) nasıl performans gösterdiğini görsel olarak temsil eder.Aşağıda kategorisel olarak grafiksel mevcuttur:

Çocuk için:



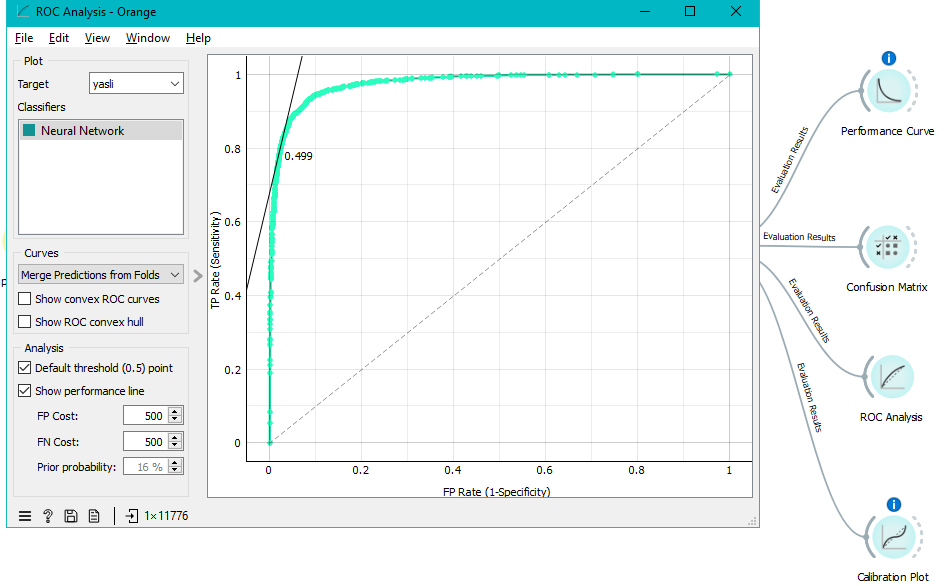
Şekil 2.8.1

Yetişkin için:



Şekil 2.8.2

Yaşlı için:



Şekil 2.8.3

**PROJE YAPIM AŞAMASI**

**1.Veri ve Görüntü İşleme**

Kaggle’dan alınan veri seti, yeni klasörler açarak kategorilendirilmiştir. Image Embedding düğümünde embedder model olarak farklı modeller denendi.İlk olarak InceptionV3 modeli denenmiştir çünkü görüntü işleme de sık kullanılan modellerin başında gelmektedir.Amaç en yüksek accuracy değerini almak olduğu için farklı modeller denenmiştir ve aralarından en optimal olanı, yüz tanımlama için kullanılanı olan openface’i tercih edilmiştir.Ayrıca en yüksek accuracy değerini de böyle bulunmuştur.

**2. Neural Network**

İlk olarak gizli katmandaki nöron sayısı rastgele 10,10,2 şeklinde denenmiştir. Bu şekilde verimlilikte problem yaşadığımızdan dolayı araştırmalarımız sonucunda zaman ve doğruluk oranından bize daha doğru sonuç verdiğine inandığımız 20,20 nöron sayısına geçiş yaptık.İki katmanlı yerine 3 katmanlı mimariyi kullandığımızda işlem süresi olarak ve performans açısından kısır sonuçlar vermekteydi.

Aktivasyon fonksiyonu olarak ReLu fonksiyonu tercih edilmiştir çünkü diğer fonksiyonları denediğimizde performans ve doğruluk ‘tanh’ ve ‘logistic’ fonkisyonlarına kıyasla daha kısa zamanda daha iyi performans sağladığı ve daha doğru accuracy değeri verniştir.

Optimizer(solver) olarak SGD(stochastic gradient descent) algoritması kullanılmıştır çünkü küçük veri gruplarında daha keskin sonuçlar verdiği görülmüştür.Zaman konusunda dezavantajlıdır fakat bizim projemiz bu yönteme uygundur. Seçenekler arasında olanAdam algoritması denenmiştir fakat SGD algoritması daha iyi performans ve sonuç sağlamıştır.

Learning Rate 0.5 olarak alınmıştır çünkü Accuracy değerine dikkat edilerek zamana bağlı olarak en optimal değer 0.5 learning rate değerinde çıkmıştır.

İterasyon sayısı hocamızdan aldığımız bilgiler eşiğinde 250 olarak belirlenmiştir.

Proje modelimizin son hali aşağıdaki gibidir:

