## **Bank Customer’s Churn Classification**

Bu veri setini oluşturmak için banka, müşteri kimliği, kredi skoru, cinsiyet, yaş, hizmet süresi, bakiye, müşterinin aktif olup olmadığı, kredi kartı olup olmadığı vb. gibi bilgileri toplamıştır. Banka, bu müşterilerin 6 aylık bir dönem boyunca bankadan ayrılıp ayrılmadığını gözlemlemiştir .Bu veri setini kullanarak amacımız , yukarıda verilen işlem bilgilerine göre, herhangi bir bireysel müşterinin bankadan ayrılıp ayrılmayacağını (müşteri kaybı), tahmin etmektir.

**Exploratory Data Analysis (EDA)**

İlk olarak Exploratory Data Analysis (EDA) yapıldı. EDA, bir veri setini anlamak, içindeki kalıpları keşfetmek ve önemli özellikleri belirlemek amacıyla yapılan keşifsel bir veri analizi sürecidir. Bu aşama, veri setini temizleme ve önceden işleme aşamalarına geçmeden önce gerçekleştirilir ve genellikle veri madenciliği ve modelleme süreçlerine ön bilgi sağlamak için kullanılır. İşte EDA işleminde aşağıdaki adımlar gerçekleştirildi:

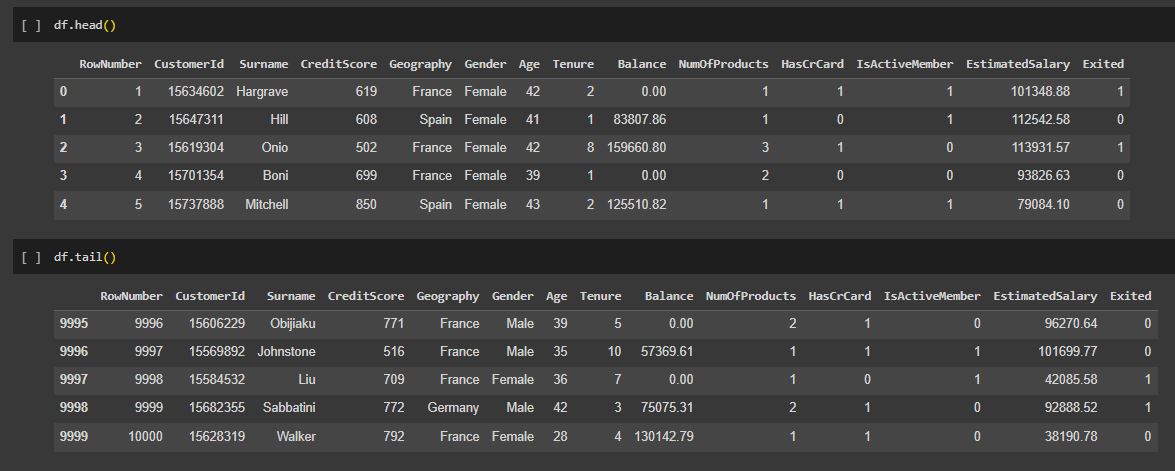
Veri Seti Gözden Geçirme:

- Veri setini yüklendi ve ilk birkaç satırını incelendi.

- Sütun başlıklarını ve veri tiplerini kontrol edildi.

metin, ekran görüntüsü, yazılım, multimedya yazılımı içeren bir resim

Açıklama otomatik olarak oluşturuldu

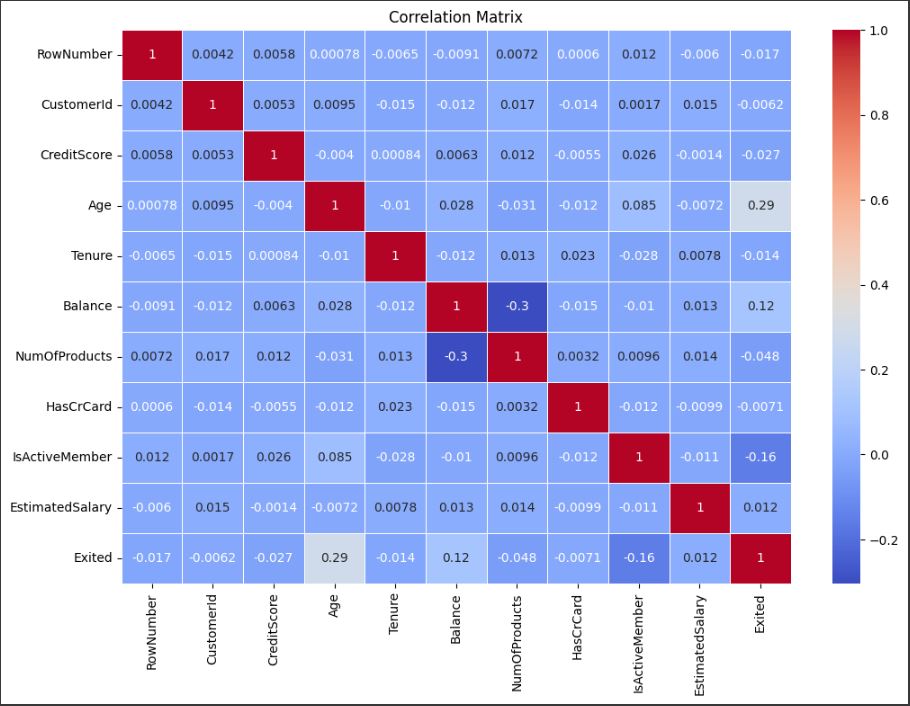


Veri Dağılımı:

- Sayısal değişkenlerin dağılımları görselleştirildi. Histogramlar, yoğunluk grafikleri ve kutu grafikleri gibi grafikler kullanarak veri dağılımlarına bakıldı.

Korelasyon Analizi:

- Değişkenler arasındaki ilişkiyi anlamak için korelasyon matrisini oluşturuldu. Korelasyon grafikleri veya ısı haritaları kullanarak görselleştirildi.



Eksik Veri Kontrolü:

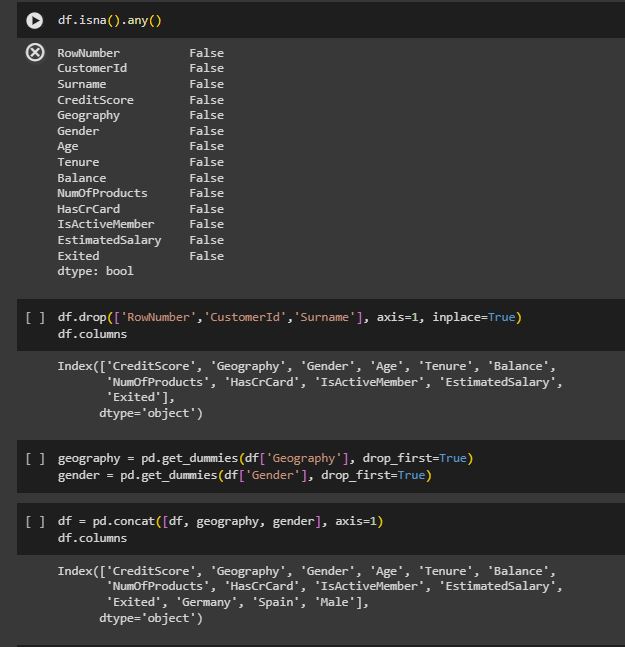
- Eksik veri olup olmadığını kontrol edildi. Eksik veri varsa, bunlarla başa çıkma stratejilerini belirlendi.

Kategorik Değişken Analizi:

- Kategorik değişkenlerin sınıflarının dağılımını ve etkilerini görselleştirildi. Belirli sütunlar veri seti için gereksiz veya kullanılmayacak bilgiler içeriyordu. Örneğin:

RowNumber: Bu sütun genellikle veri setindeki her bir satırın numarasını içerir. Veri setindeki satırları tanımlamak için kullanışlı olabilir, ancak model eğitimi için asıl özelliklerden biri değildir.

CustomerId ve Surname: Müşteri kimlik numaraları veya soyadları gibi sütunlar da genellikle modelin tahmin yapmak için gerekli olmayan bilgilerdir. Modelin müşteri churn (müşteri kaybı) tahmini yapması için bu tür kişisel bilgiler genellikle gereksizdir ve sadece gereksiz gürültü ekleyebilir. Bu nedenle, bu sütunların çıkarılması modelin performansını etkilemeyebilir, ancak modelin daha temiz ve odaklanmış bir veri üzerinde çalışmasına yardımcı olabilir.

Bu nedenlerden dolayı model eğitimi için bu sütunlar çıkartıldı.

**Feature Engineering**

pd.get\_dummies() fonksiyonu, kategorik sütunları ikili (binary) sütunlara dönüştürmek için kullanılır. Özellikle, makine öğrenimi modelleri genellikle sayısal verilerle çalışmayı tercih eder, bu yüzden kategorik verileri (örneğin, ülke isimleri veya cinsiyet gibi) modelin anlayabileceği sayısal formatlara dönüştürmek gerekir.

Örneğin, Geography sütunu ülke isimlerini içeriyorsa, bu sütunu ikili (binary) kodlama yoluyla dönüştürmek, her bir ülkeyi ayrı ayrı sütunlara çevirerek yapılır. Örneğin, eğer ülke sütunu 'Fransa', 'Almanya' ve 'İspanya' gibi ülke isimlerini içeriyorsa, pd.get\_dummies() fonksiyonu bu ülkeleri ayrı ayrı sütunlara çevirir. Bu durumda, her bir müşterinin ait olduğu ülke '0' veya '1' olarak ifade edilir.

Özetle, pd.get\_dummies() kullanılarak kategorik değişkenlerin ikili (binary) formatlara dönüştürüldü.Bu dönüşüm, model eğitimi sırasında kategorik bilgilerin sayısal olarak temsil edilmesini sağladı.

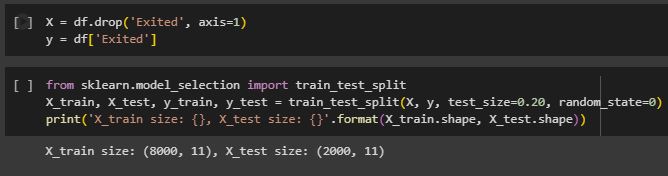
**Data Preprocessing**

'Exited' sütunu hedef değişken olacak şekilde ayarlandı, geriye kalan sütunlar bağımsız değişkenler olarak 'X' olarak atandı.

'Exited' sütununu hedef değişken olarak belirlendi ve 'y' değişkenine atandı. 'Exited' sütunu, modelin öngörmeye çalıştığı veya tahmin etmeye çalıştığı değerdir. Bu durumda, 'Exited' sütunu müşteri churn (müşteri kaybı) durumunu temsil ediyor.

Veri seti eğitim ve test olarak ikiye ayrıldı. Eğitim seti üzerinde model eğitilir ve test seti üzerinde modelin performansı değerlendirmesi yapılır.

Veri setinin %20'sinin test seti olarak ayrıldı. veri setinin bölmek için kullanılacak rastgele bir durumu sabitler, bu da her seferinde aynı rastgele bölme yapılmasını sağlar.



**Building Artificial Neural Network (ANN)**

İlk katman, 11 giriş özelliğini (input\_dim=11) alır. Bu katman 6 nörona sahiptir, He başlatma yöntemi ile ağırlıkları başlatılır ve ReLU aktivasyon fonksiyonu kullanıldı.

İkinci ve üçüncü gizli katmanlar sırasıyla 6 nörona ve 1 nörona sahiptir. Ağırlıklar He başlatma yöntemi ile başlatılır ve aktivasyon fonksiyonları 'relu', 'sigmoid', ve 'swish' olarak sırasıyla kullanılır.

Model, Adam optimizer ile derlenir. Binary crossentropy, ikili sınıflandırma problemleri için kullanılan bir kayıp fonksiyonudur. Accuracy (doğruluk), modelin başarısını değerlendiren bir metriktir. Bu kod, giriş katmanı, gizli katmanlar ve çıkış katmanı içeren bir yapay sinir ağı modeli oluşturur ve bu modeli derler. Önceki kod ile farkı, üçüncü gizli katmanın aktivasyon fonksiyonunun 'swish' olarak belirlenmiş olmasıdır. Swish, ReLU'ya benzer bir aktivasyon fonksiyonudur ve bazı durumlarda daha iyi performans gösterebilir.

Aşağıda, kodunuzu farklı parametrelerle deneyerek elde ettiğiniz sonuçları anlatan bir metin örneği bulunmaktadır:

Bu yapay sinir ağı modelini oluştururken çeşitli hiperparametreleri değiştirerek farklı sonuçlar elde ettik. İlk olarak, modelin karmaşıklığını belirleyen nöron sayılarını düzenledik. İlk katmanda 6 nöron, ikinci katmanda 6 nöron ve çıkış katmanında 1 nöron kullanarak modelin genel karmaşıklığını ayarladık. Bu değişiklikler, ağın veri setindeki örüntüleri öğrenme yeteneğini etkiledi.

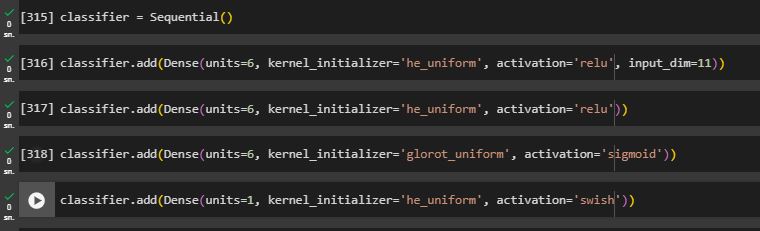
Ağırlık başlatma yöntemlerini de değiştirdik. Giriş ve ikinci gizli katmanlarda He\_uniform, çıkış katmanında ise Glorot\_uniform başlatma yöntemlerini kullandık. Bu başlatma yöntemleri, ağırlıkları rastgele başlatma şekillerini belirler ve bu dağınık başlatma genellikle modelin daha hızlı ve etkili bir şekilde öğrenmesine yardımcı olabilir.

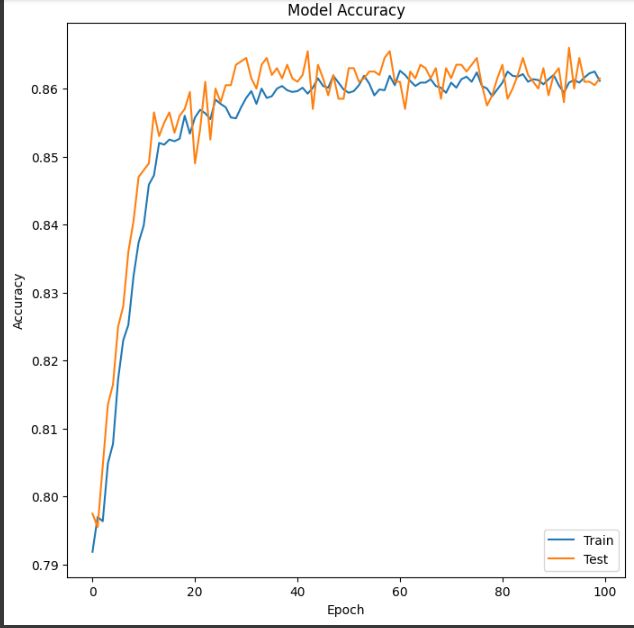
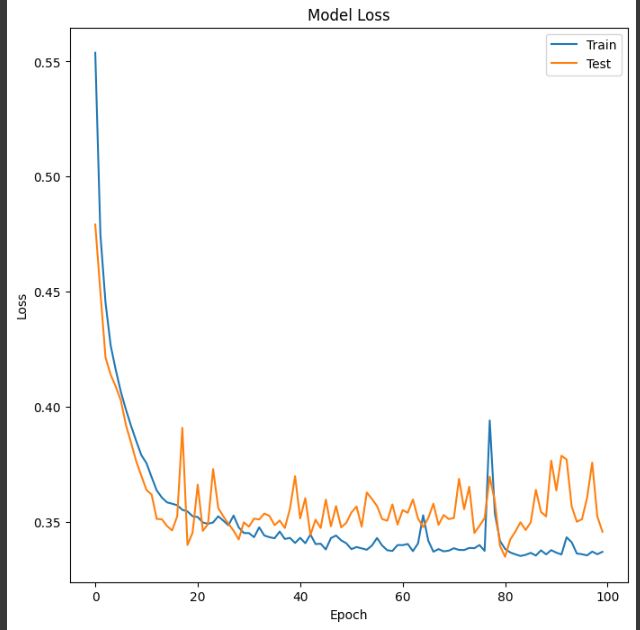
Optimizasyon algoritmasını Adam optimizer'ı ile belirledim ve öğrenme hızını (learning rate) 0.001 olarak ayarladık. Bu, modelin eğitim sürecini yönlendiren ve ağırlık güncellemelerini düzenleyen bir faktördür.

Eğitim sırasında, veriyi daha küçük gruplara (mini-batches) bölmek için batch\_size'ı 10 olarak belirledim. Bu, ağı eğitmek için kullanılan her veri setinin boyutunu belirler.

Eğitim sürecinin sonuçlarını değerlendirmek için modelin doğruluk (accuracy) ve kayıp (loss) metriklerini görselleştirdik. Grafikler, modelin eğitim setinde ve test setindeki performansını karşılaştırmak için kullanılmaktadır.

Farklı hiperparametre değerleri ve başlatma yöntemleri kullanarak modeli eğittiğimde, doğruluk ve kayıp değerlerinde değişiklikler gözlemledim. Bu, modelin belirli hiperparametre ayarlarına nasıl tepki verdiğini anlamama yardımcı oldu ve optimal performansı elde etmek için deneme yanılma yöntemini kullanmama olanak tanıdı.

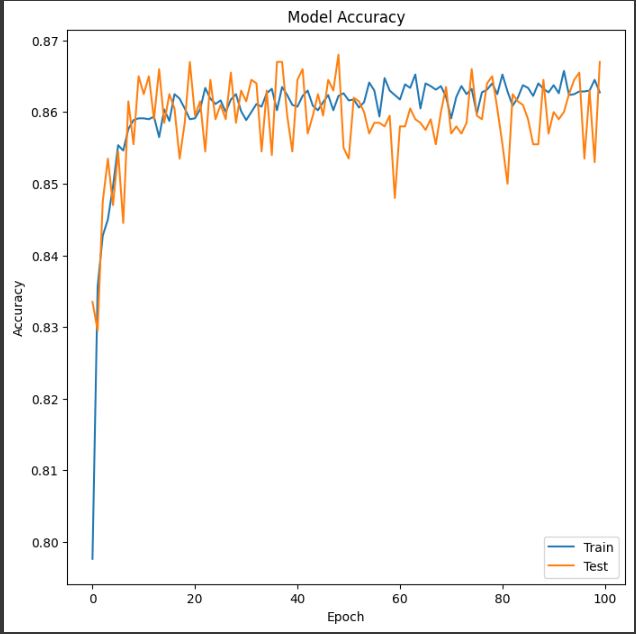
Yapılan bazı parametre değişiklikleri şu şekilde sonuçlar ortaya çıkarmıştır.

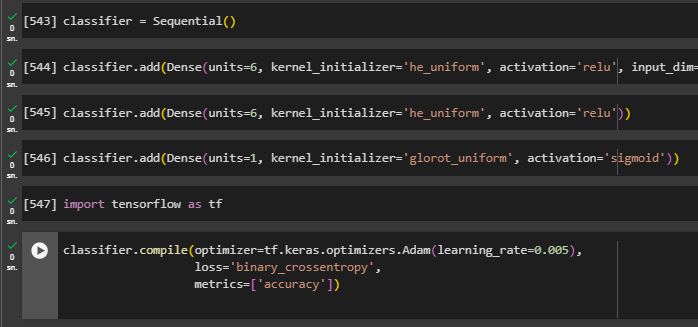


ekran görüntüsü, metin, öykü gelişim çizgisi; kumpas; grafiğini çıkarma, yazılım içeren bir resim

Açıklama otomatik olarak oluşturulduekran görüntüsü, metin, öykü gelişim çizgisi; kumpas; grafiğini çıkarma, diyagram içeren bir resim

Açıklama otomatik olarak oluşturuldu

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

Açıklama otomatik olarak oluşturuldu

metin, ekran görüntüsü, yazılım, multimedya yazılımı içeren bir resim

Açıklama otomatik olarak oluşturulduTüm yapılan epoch sayısı, learning rate oranı,nöron sayısı değişiklikleri sonucunda en iyi sonuç bu şekilde alınmıştır.Ve Confusion matrixi oluşturulmuştur.

ekran görüntüsü, ekran, görüntüleme, metin, yazılım içeren bir resim

Açıklama otomatik olarak oluşturulduekran görüntüsü, metin, dikdörtgen içeren bir resim

Açıklama otomatik olarak oluşturuldu

True Positive (TP): 1509 Gerçek sınıfı 1 olan 1509 örneği doğru bir şekilde 1 olarak tahmin ettik.

True Negative (TN): 222 Gerçek sınıfı 0 olan 222 örneği doğru bir şekilde 0 olarak tahmin ettik.

False Positive (FP): 86 Gerçek sınıfı 0 olan 86 örneği 1 olarak yanlış bir şekilde tahmin ettik.

False Negative (FN): 183 Gerçek sınıfı 1 olan 183 örneği 0 olarak yanlış bir şekilde tahmin ettik.

metin, ekran görüntüsü, yazı tipi, diyagram içeren bir resim

Açıklama otomatik olarak oluşturuldu

Emir Yılmaz 190101025

Esra Özdemir 190101040