**FBBMS512 | Derin Öğrenme ve Evrişimli Sinir Ağları Dersi**

**ARA SINAV**

**Raporu**

HALİL İBRAHİM TEBER

**22040718013**

**Ön işleme-Preprocessing**

Bu projede Hepatitis C veri seti kullanıldı.Bu veri seti Kaggle üzerinden

<https://www.kaggle.com/datasets/fedesoriano/hepatitis-c-dataset> adresinden csv formatında indirildi.

Veri seti incelendiğinde, veri setinin ilk sutunu kaldırıldı.Bu sutunda herhangi bir gerekli verinin olmadı görüldü. Gereksiz yere işleme sokulmadı. Kategori bölümünde ve cinsiyet bölümünde '0=Blood Donor', '0s=suspect Blood Donor', '1=Hepatitis', '2=Fibrosis', '3=Cirrhosis' gibistring ifadeler bulunmaktaydı. Bu ifadeleri Kan donor 0 , şüpheli kan donor -1 ,hepatit için 1, fibroz için 2 ve siroz için ise 3 olacak şekilde düzenlendi.

Cinsiyet verileri erkekler için m ve kadınlar için f olacak şekilde kategorize edilmişti.Bu verileri erkekler -1 kadınlar -2 olacak şekilde revize edildi.

Veriye göz atılınca veride boş verilerin olduğu saptandı kategorilerdeki null değerler aşağıdaki gibiydi;

|  |  |  |  |
| --- | --- | --- | --- |
| Category | 0 | BIL | 0 |
| Age | 0 | CHE | 0 |
| Sex | 0 | CHOL | 10 |
| ALB | 1 | CREA | 0 |
| ALP | 18 | GGT | 0 |
| ALT | 1 | PROT | 1 |
| AST | 0 |  |  |

Yukardaki tablodada görüldüğü gibi özniteliklerin bulunduğu sutunların bazılarında null değerler bulunmakta.Bu null verilerin toplam veri sayısına nazaran az olması sebebiyle ortalama değerler ile doldurulmasını uygun gördüm. Her kategorinin ortalama değerleri alınarak bu değerler null değerler konumuna gelecek şekilde dolduruldu.

|  |  |  |  |
| --- | --- | --- | --- |
| Category | 0 | BIL | 0 |
| Age | 0 | CHE | 0 |
| Sex | 0 | CHOL | 0 |
| ALB | 0 | CREA | 0 |
| ALP | 0 | GGT | 0 |
| ALT | 0 | PROT | 0 |
| AST | 0 |  |  |

Üstteki tabloda da görüldüğü gibi veri seti içerisinde hiç bir null değer kalmamış oldu.

Veri üzerinde herhangi bir normalizasyon işlemi yapıldığında modellerin doğruluk oranlarının düştüğü gözlemlenmiştir.Ham veriler ile daha yüksek doğrulukta model oluşmuştur. Bunun sebebi normalizasyon yapılırken normalizasyon verilerinin bütün özniteliklere uygun olmadığını tahmin ediyorum.Fakat tahmin değerlerinde herhangi bir değişim görülmedi.

**Öznitelik Mühendisliği(Feature Engineering)**

Korelasyon katsayısı 1'e yaklaşırken ilişki düzeyi yükselirken, -1'e yaklaştıkça ilişki düzeyi azalmaktadır. Korelasyon katsayısı “-1” ve “+1” arasında değerler alır.Eğer -1 ise tam negatif doğrusal bir ilişki, +1 ise tam pozitif doğrusal bir ilişki vardır.Bu değer 0 ise iki değişken arasında ilişki yoktur.Bu ilişki için çıkan sonuçlar aşağıdaki aralıklara göre şu şekilde yorumlanır:

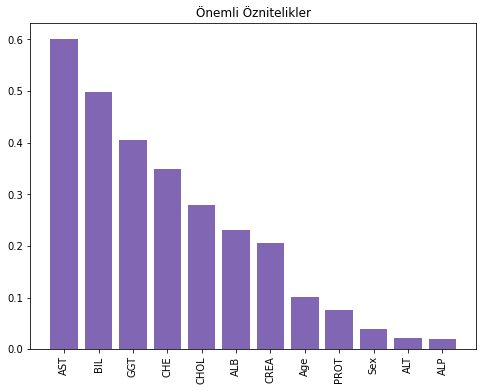
•0.00 ilişki yok

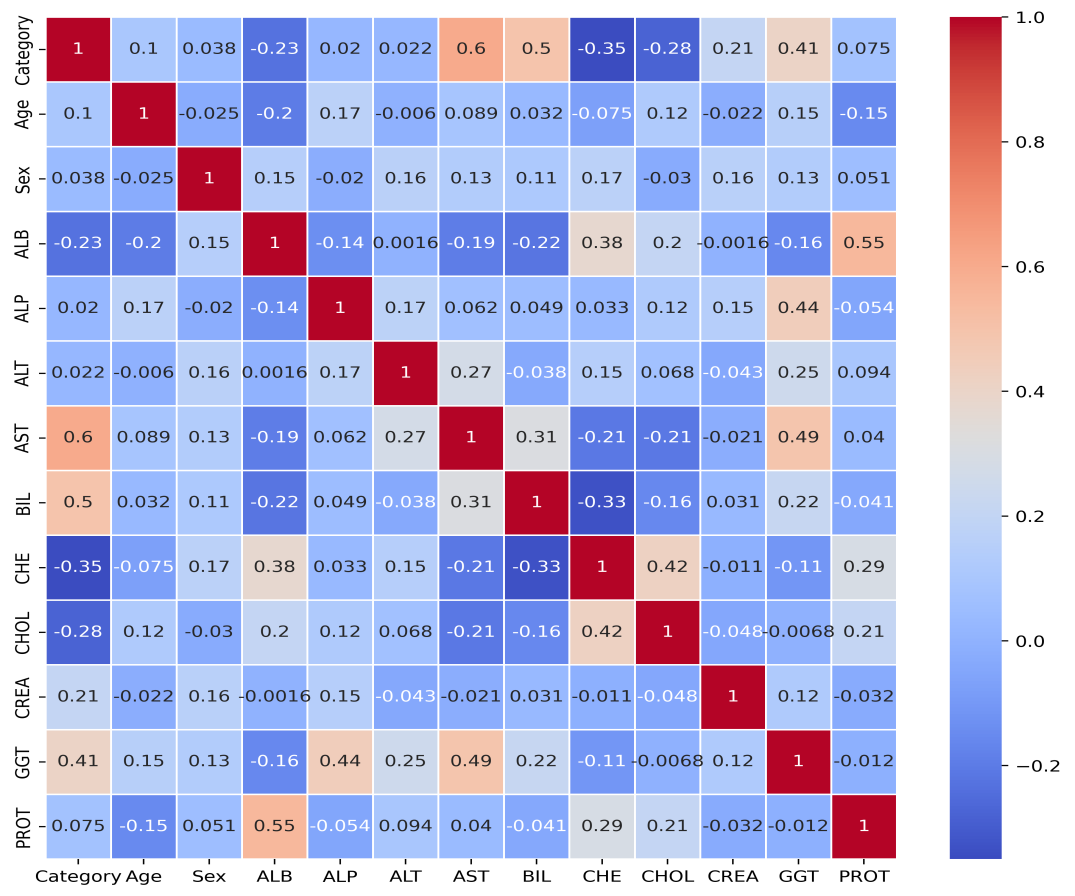
•0.01 – 0.29 düşük düzeyde ilişki

•0.30 – 0.70 orta düzeyde ilişki

•0.71 – 0.99 yüksek düzeyde ilişki

•1.00 mükemmel ilişki





Yapılan korelasyon analizinde yukardaki korelasyon dağılım haritası verilen resimde de görüleceği üzere AST,BIL ve GGT özniteliklerinin olduğu bulundu. Burada korelasyon değerlerini incelediğimizde aşırı öğrenmeye sebep olacak 0.90 üzeri veri olmaması, verileri 0.30~0.60 ile arasında olması sebebiyle öznitelik sınır değerini 0.4 ve üzeri değerler alındı. Bu sebeple 3 paremetre hepatit verisinde daha belirleyici özniteliklerdir.

Veri setini incelediğimde veri setindeki öznitelik sayısının 12 tane olması bu veriye herhangi bir PCA yapmayı gerek duymadım.Veri setinden sondaki 5 veri en az modele katkı sağlayan öznitelikler.Bu öznitelikleri verisetinden çıkarırsak daha yüksek başarı oranı sağlayacağını düşünmüyorum.Çünkü zaten en az belirleyici özniteliklerdir.Fakat daha iyi bir model olması en azından daha hızlı çalışan bir model kurmamıza yardımcı olabilir.Burada CREA ve ALB verisi için doğruluk oranı denenebilir.

**Deneysel Sonuçlar**

GridSearchCV ile yapılan optimum parametre bulma sonucunda aşağıdaki parametreler bu veri seti için optimum değerlerdir;

Logistic Regression: {'C': 10, 'max\_iter': 50, 'penalty': 'l2'}

Logistic Regression: 0.8602150537634409

KNN algoritması: {'n\_neighbors': 1, 'weights': 'uniform'}

KNN algoritması: 0.8494623655913979

Naive bayes: {'priors': None, 'var\_smoothing': 1e-09}

Naive bayes: 0.8602150537634409

Karar Ağacı: {'max\_depth': 4, 'random\_state': 0}

Karar Ağacı: 0.9032258064516129

Çok Katmanlı: {'activation': 'relu', 'hidden\_layer\_sizes': 100, 'learning\_rate': 'constant', 'learning\_rate\_init': 0.01, 'max\_iter': 100, 'solver': 'adam'}

Çok Katmanlı: 0.9139784946236559

Rasgele Ağaç: {'max\_depth': 20, 'n\_estimators': 150}

Rasgele Ağaç: 0.9032258064516129

Destek Vektör Makinesi: {'C': 10, 'gamma': 'scale', 'kernel': 'linear'}

Destek Vektör Makinesi: 0.8709677419354839

**Doğrulama verileri**

Aşağıda %85 Eğitim Seti ile eğitilmiş modellerin verileri bulunmaktadır.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Sınıf** | **Dogruluk**  **(Accuracy)** | **Duyarlilik**  **(Precision)** | **Hassasiyet**  **(Recal)** | **F-measure** | **AUC** |
| Logistic Regression | 0.860 | 0.825 | 0.860 | 0.836 |  |
| KNN algoritması | 0.849 | 0.819 | 0.849 | 0.830 |  |
| Naive bayes | 0.860 | 0.905 | 0.860 | 0.875 |  |
| Karar Ağacı | 0.903 | 0.889 | 0.903 | 0.886 |  |
| Çok Katmanlı | 0.913 | 0.919 | 0.913 | 0.908 |  |
| Rasgele Ağaç | 0.903 | 0.899 | 0.903 | 0.890 |  |
| Destek Vektör Makinesi | 0.870 | 0.832 | 0.870 | 0.847 |  |

Aşağıda %70 Eğitim Seti ile eğitilmiş modellerin verileri bulunmaktadır.

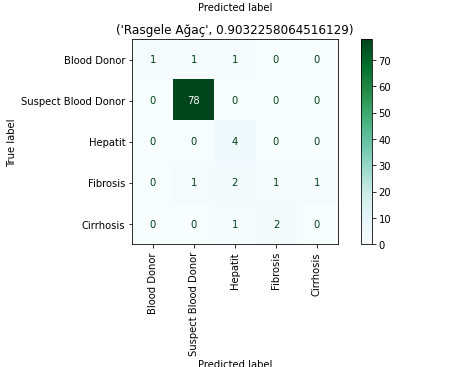
|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Sınıf** | **Dogruluk**  **(Accuracy)** | **Duyarlilik**  **(Precision)** | **Hassasiyet**  **(Recal)** | **F-measure** | **AUC** |
| Logistic Regression | 0.881 | 0.836 | 0.881 | 0.850 |  |
| KNN algoritması | 0.870 | 0.867 | 0.870 | 0.831 |  |
| Naive bayes | 0.859 | 0.856 | 0.859 | 0.857 |  |
| Karar Ağacı | 0.881 | 0.828 | 0.881 | 0.852 |  |
| Çok Katmanlı | 0.891 | 0.901 | 0.891 | 0.890 |  |
| Rasgele Ağaç | 0.881 | 0.848 | 0.881 | 0.843 |  |
| Destek Vektör Makinesi | 0.886 | 0.838 | 0.886 | 0.858 |  |

Veri setinde 5 farklı çıkış bulunduğundan AUC değeri hesaplanamamıştır.Fonksiyon sürekli boyut hatası vermiştir.

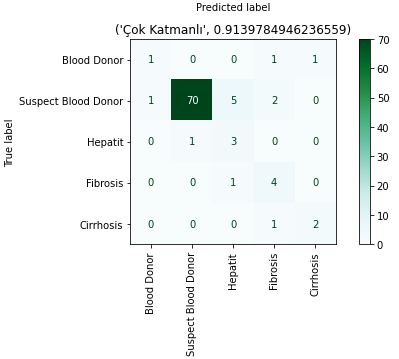
**Değerlendirme-Sonuç**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Sınıf** | **Dogruluk**  **(Accuracy)** | **Duyarlilik**  **(Precision)** | **Hassasiyet**  **(Recal)** | **F-measure** | **AUC** |
| Logistic Regression | 0.860 | 0.825 | 0.860 | 0.836 |  |
| KNN algoritması | 0.849 | 0.819 | 0.849 | 0.830 |  |
| Naive bayes | 0.860 | 0.905 | 0.860 | 0.875 |  |
| Karar Ağacı | 0.903 | 0.889 | 0.903 | 0.886 |  |
| Çok Katmanlı | 0.913 | 0.919 | 0.913 | 0.908 |  |
| Rasgele Ağaç | 0.903 | 0.899 | 0.903 | 0.890 |  |
| Destek Vektör Makinesi | 0.870 | 0.832 | 0.870 | 0.847 |  |

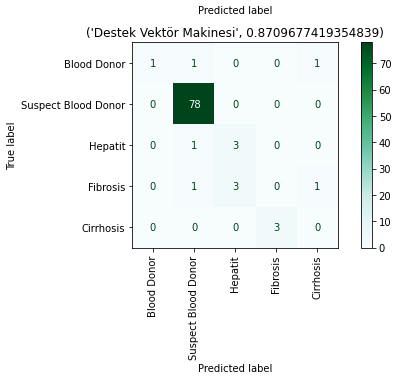
Yukarıdaki tabloda doğruluk ,duyarlılık ,hassasiyet gibi değerlere bakıldığında en iyi doğruluk değerinin çok katmanlı modelin verdiğini görebiliriz. Fakat tahmin verilerine bakarak en iyi modele karar verelim.Aşağıdaki tabloya bakarsak rasgele ağaç modeli şüpheli kan donorlarının 78ini doğru bilmiş.Gerçek hepatit hastası 4 tane iken model 8 tane hepatit hastası tahmin etmiş.4 tane FN hasta tahmin etmiş.Hiç siroz hastası tahmin edememiş.Siroz hastasını hepatit ve fibroz şeklinde tahmin etmiş(FN).



Karar ağaçlarının en büyük problemlerinden biri aşırı öğrenmedir. Rasgele ağaç modeli bu problemi çözmek için hem veri setinden hem de öznitelik setinden rasgele olarak 10'larca 100'lerce farklı alt veri setleri seçiyor ve bunları eğitiyor. Bu yöntemle 100'lerce farklı karar ağacı oluşturuluyor.Her bir karar ağacı bireysel olarak tahminde bulunuyor. En sonunda problemimiz regresyon ise karar ağaçlarının tahminlerinin ortalamasını verir eğer problemimiz sınıflandırma ise tahminler arasında en çok oy alan seçiliyor.

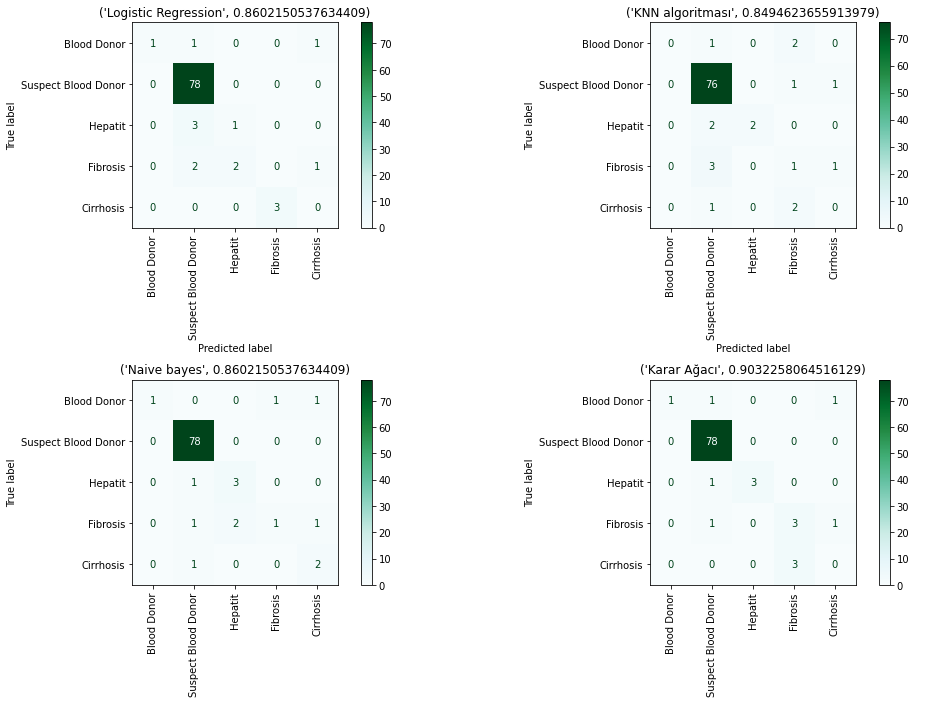


Çok katmanlı modeli incelediğimizde 78 hastanın 70 tane şüpheli donor tahmin etmiş yani 5 tane TN.Hepatitli hasta 4 tane iken model 3 tane tahmin edebilmiş yani 3 tanesi TP. 5 tane donore yanlış hepatit teşhisikoymuş(FN).Bu model 2 tane siroz hastası doğru tahmin etmiş (TP),bir tane yanlış tahmin etmiş siroz hastasını.Fibroz hastalarını teşhis etmede bu model daha yüksek oran vermiş 5 hastanın 4 ünü tahmin edebilmiş.



Destek vektör makinesini incelediğimizde 78 hastanın hepsini doğru tahmin etmiş(TP).5 tane hepatit hastasının yanlızca 2 tanesini bilebilmiş(TN). Siroz hastalarının tamamını bilmiş(TP). Fibroz hastalarının yanlızca birisini doğru bilmiş.Sağlıklı hiçbir donor tahmin edememiş(TF).

Görüleceği üzere çok katmanlı ve rasgele ağaç modelini karşılaştırdığımızda hepatit hastalarını doğru tahmin etmede en iyi modelin rasgele ağaç modeli olduğu görülmektedir.Doğruluk oranının çok katmanlıya göre düşük olmasının sebebi çok katmanlının fibroz hastalarını doğru tahmin etmesidir.Fakat modeli sadece hepatit hastası veya hasta değil şeklinde olsaydı doğruluk oranı rasgele ağaç modelinde daha yüksek olacaktı.

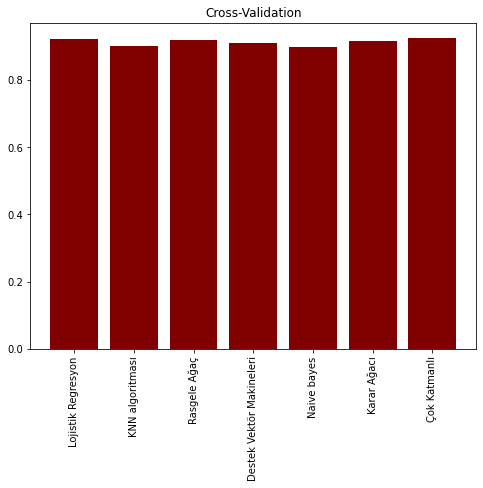


Lojistik regresyon modelinde sadece 1 tane hepatit hastası tahmini yapabilmiş.Siroz ve fibroz hastalarınının tamamını yanlış teşhis etmiş.

Genel olarak baktığımızda Hepatit teşhisi için rasgele ağaç modelini kullanmak mantıklı olacaktır. Çünkü 4 hastanın 4 ünüde doğru tahmin etmiş.Daha fazla hepatit olan hastalardan alınacak veriler ile daha doğru tahmin yapmak mümkün olacaktır.600 civarı bir veri seti çok kısıtlı bir veri setidir.Ayrıca veri seti içerinde null değerlerin olması modele etki eden faktörlerden.Daha yüksek sayıda veri seti ile (2000~3000) daha sağlam bir model geliştirilebilir.

**Cross-Validation**

Modelleri çapraz doğrulamaya tabi tutunca aşağıdaki veriler bulunmuştur.

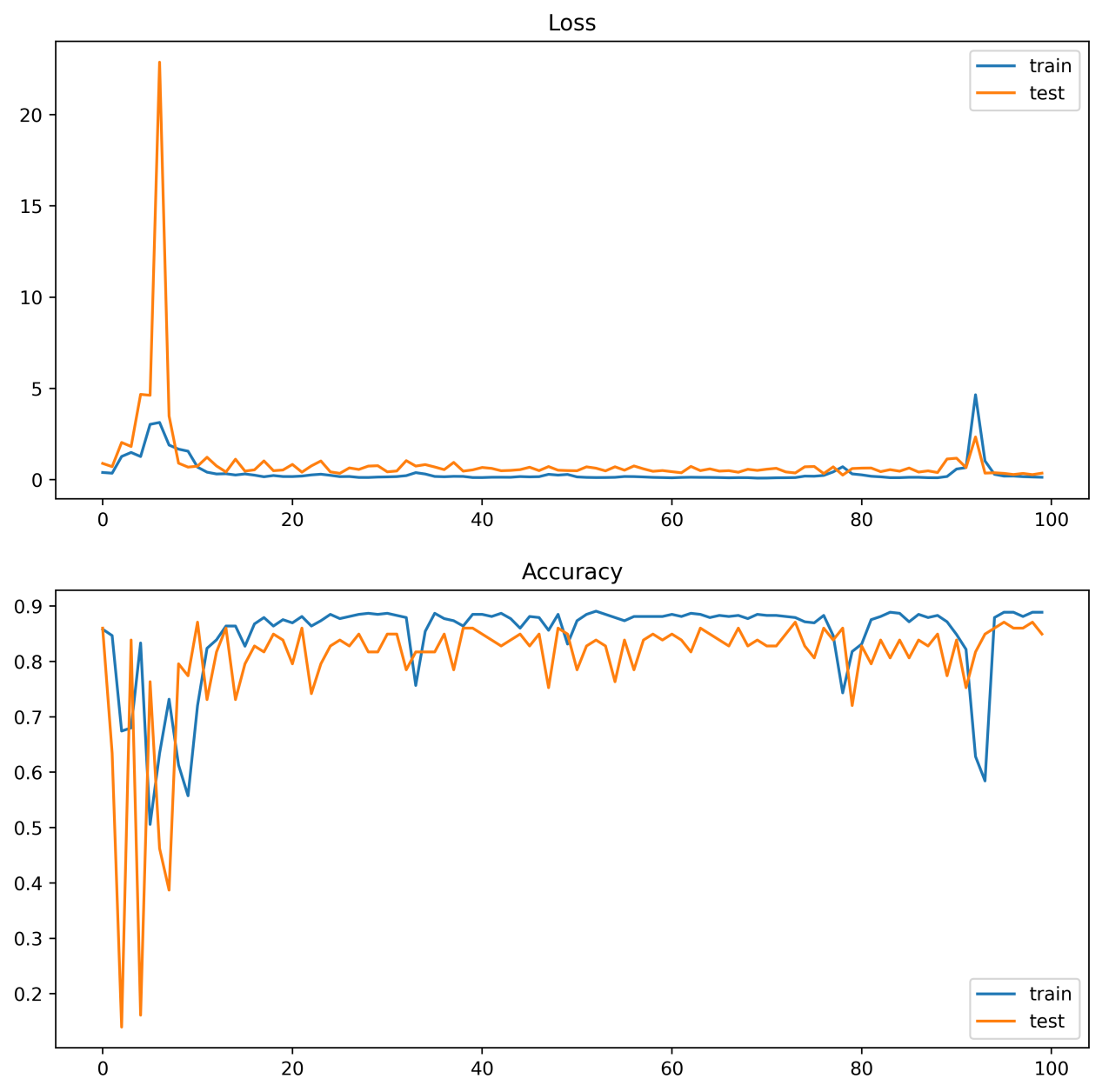


|  |  |  |
| --- | --- | --- |
| **Sınıf** | **Dogruluk**  **Cross-Val** | **Doğruluk**  **Normal** |
| Logistic Regression | 0.919 | 0.860 |
| KNN algoritması | 0.900 | 0.849 |
| Naive bayes | 0.896 | 0.860 |
| Karar Ağacı | 0.915 | 0.903 |
| Çok Katmanlı | 0.923 | 0.913 |
| Rasgele Ağaç | 0.917 | 0.903 |
| Destek Vektör Makinesi | 0.908 | 0.870 |

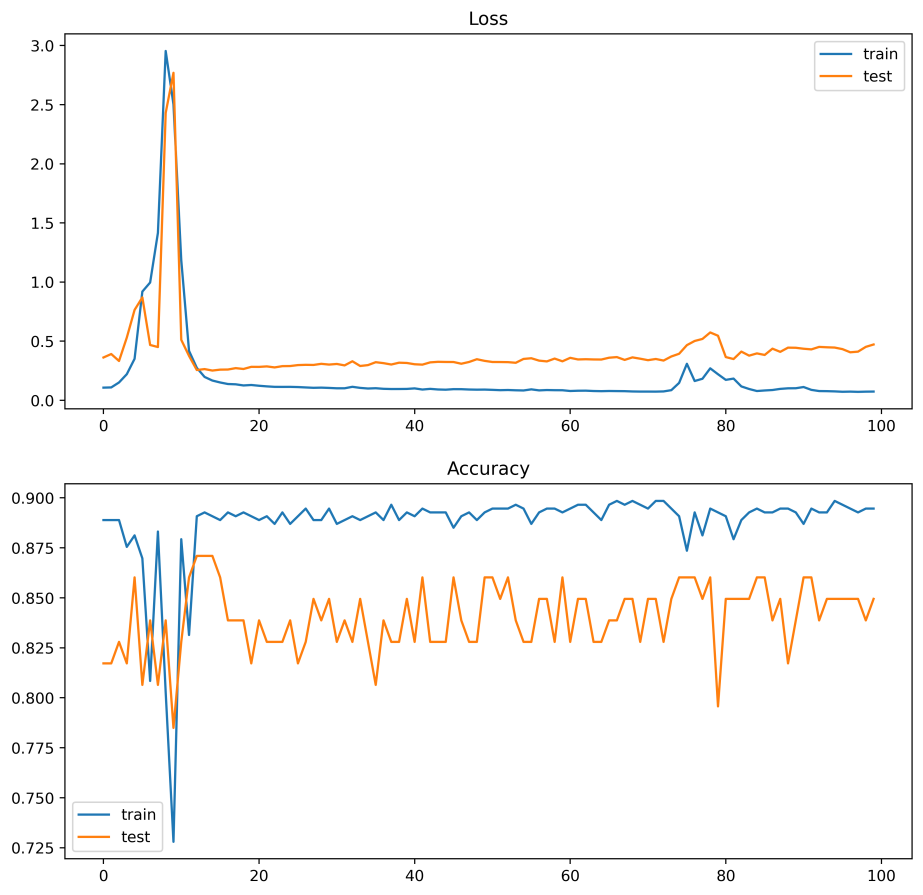
Çapraz doğrulamada Kfold değeri 5 olarak alındı.Bunun nedeni veri setinin küçük olmasından dolayı 5 katmanın yeterli olacağını düşündüm.Veri setini çapraz doğrulamaya sokunca bazı modellerin doğruluk oranları değişmiştir.Farklı bölünmeler için accuracy skorlarının değiştiğinden dolayı tek bir veri seti üzerinden modelin performansını değişiyor. Çapraz doğrulama veri seti rastgele olacak şekilde karıştırır daha sonra bu veri setini k tane gruba ayırır.Daha sonra her grup için o veri setini doğrulama seti olarak kullanılır.Diğer gruplar eğitim verisi olarak kullanılır.Daha sonra model kurularak pek çok sayıda doğruluk değeri oluşturulur.Bu değerlerin ortalamasını alarak ortalama çapraz değerlendirme değerini bulmuş oluruz.Yukarıdaki değerde bu bir kaç değerin ortalaması ile alınmış bir sonuçtur.Bu işlem bize daha doğru bir doğruluk oranı verir.

Keras Denemesi

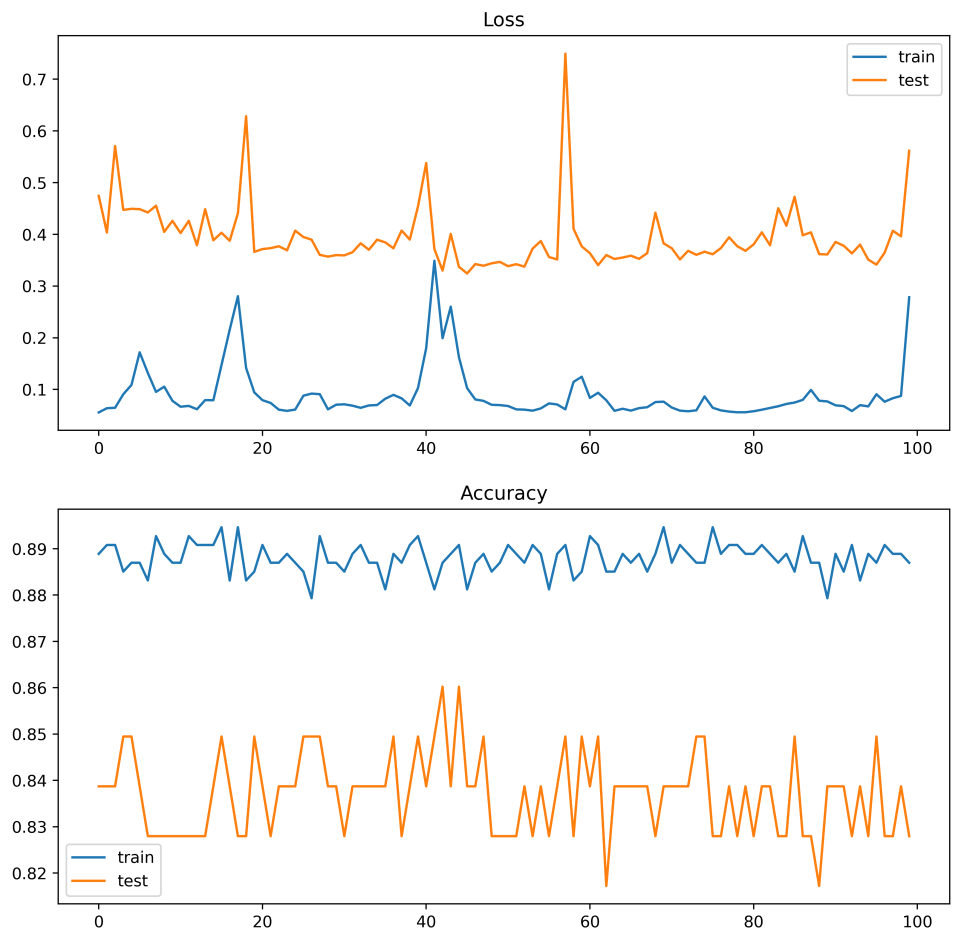
Kerasda iki adet 60 denselik katman ile adam optimizer da farklı denemeler yapılmıştır.



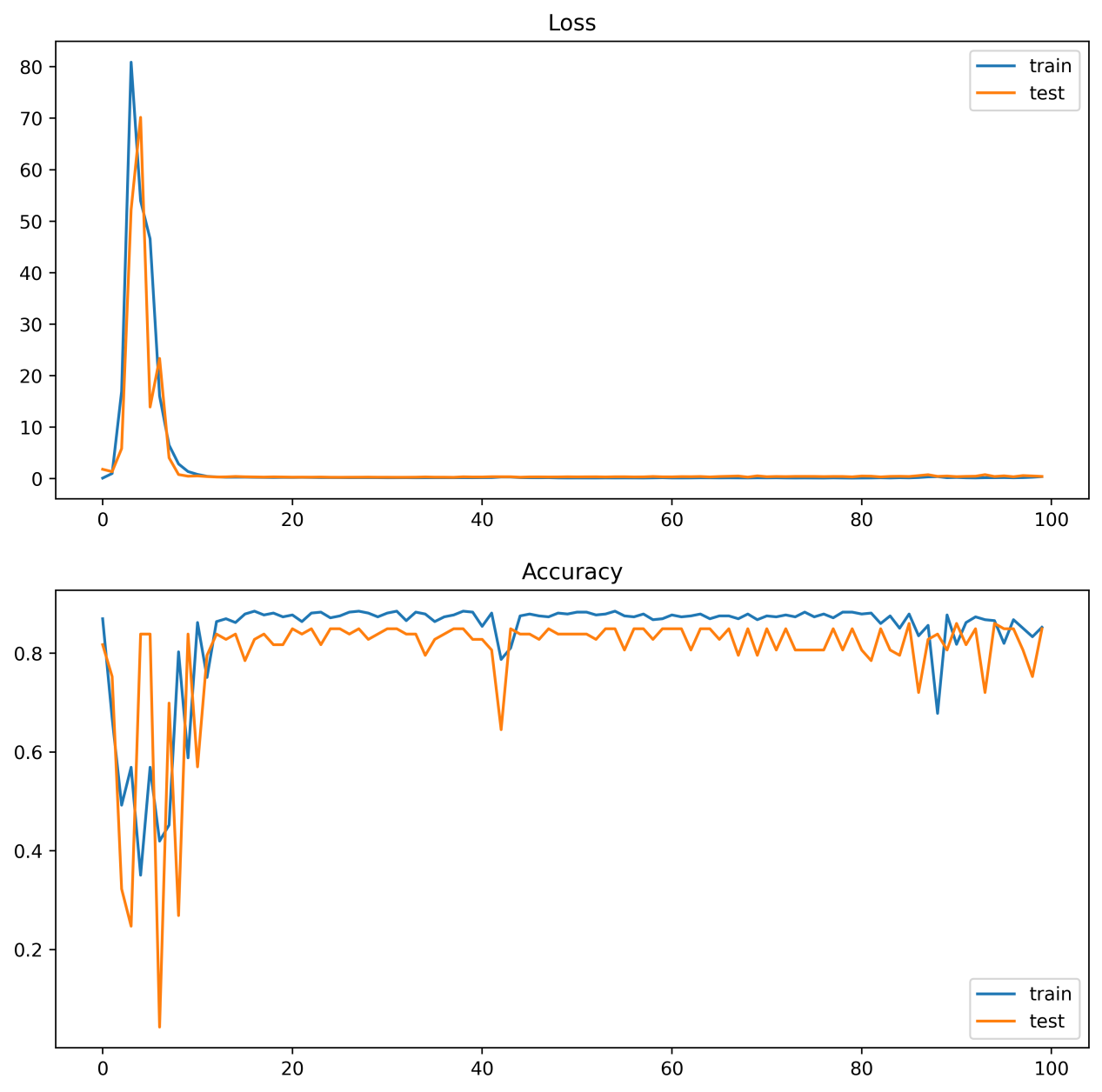
batch\_size=50, epochs=500

****

**batch\_size=100, epochs=1000**

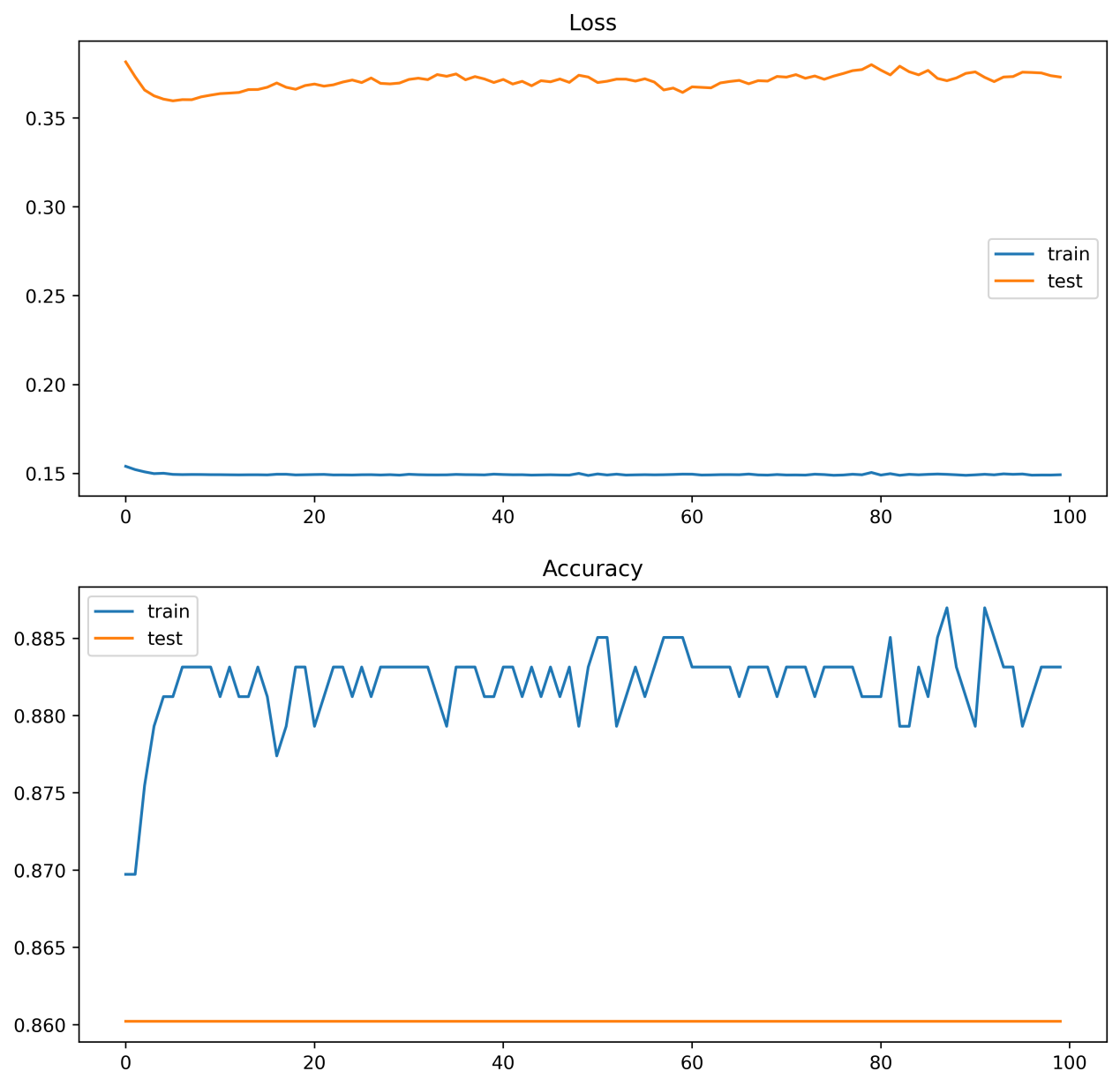
****

**batch\_size=50, epochs=1000**

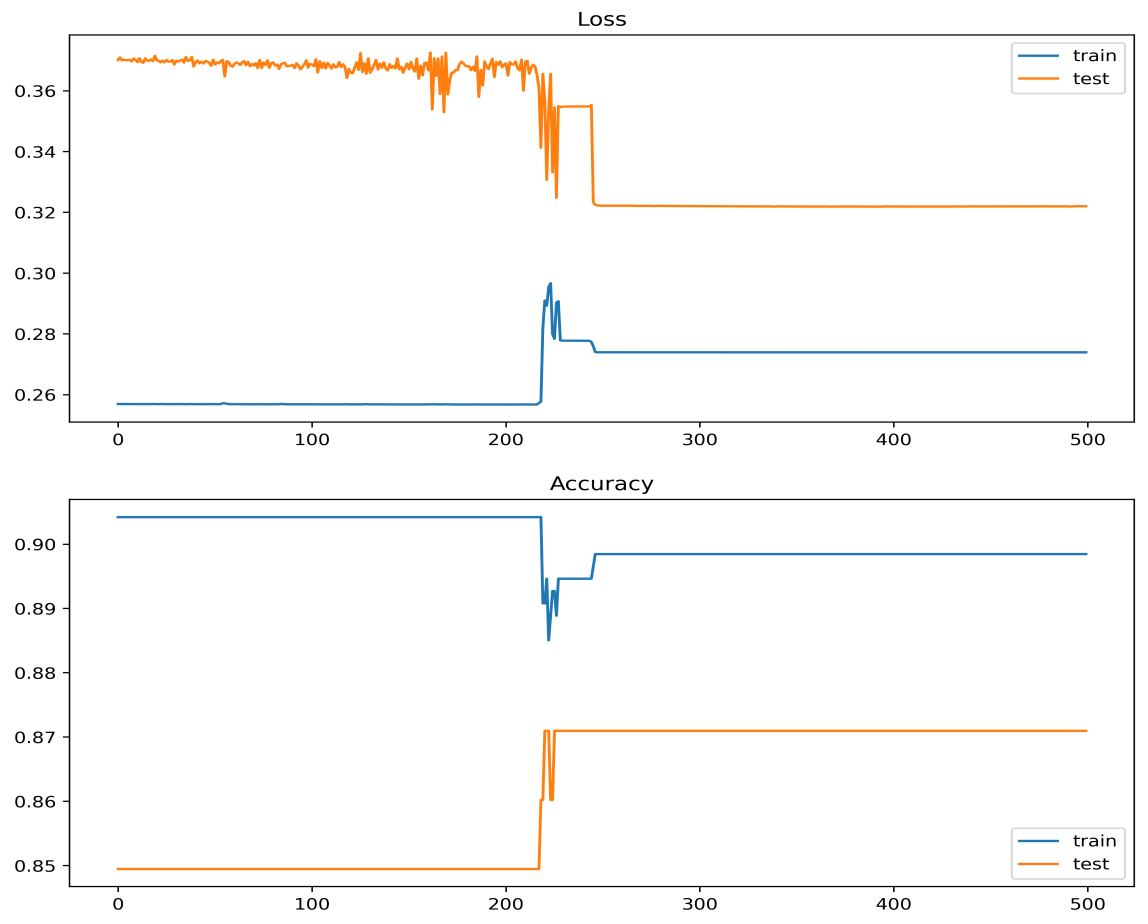
****

**batch\_size=250, epochs=1000**

Yukardaki farklı değerlerde yapılan testlerde batch boyutu ve epoch sayısının çok iyi seçilmesi gerektiği gözlemlenmiştir. Veri seti batch değeri olarak belirlenen değere göre parçalara ayrılır ve her iterasyonda modelin eğitimi bu parça üzerinden yapılmaktadır.Batch boyutunun küçük seçilmesi regülasyon etkisi yapmaktadır. Modele veri büyük gruplar halinde verildiğinde ezberleme daha fazla olmaktadır.Veri setindeki sınıflar bir birine çok fazla benziyorsa ya da çok fazla gürültülü veri varsa, doğruluk grafiği çok fazla inişli çıkışlı bir grafik sergileyebilir. Veri işlendiği için bazı iterasyondaki veriler için seçilen parametreler tam uygun iken bazısında uygun olmayabilir. Bununla birlikte iterasyon arttıkça bu zikzaklar azalacaktır. Bunu çözmenin yolu düşük öğrenme oranı vermektir.

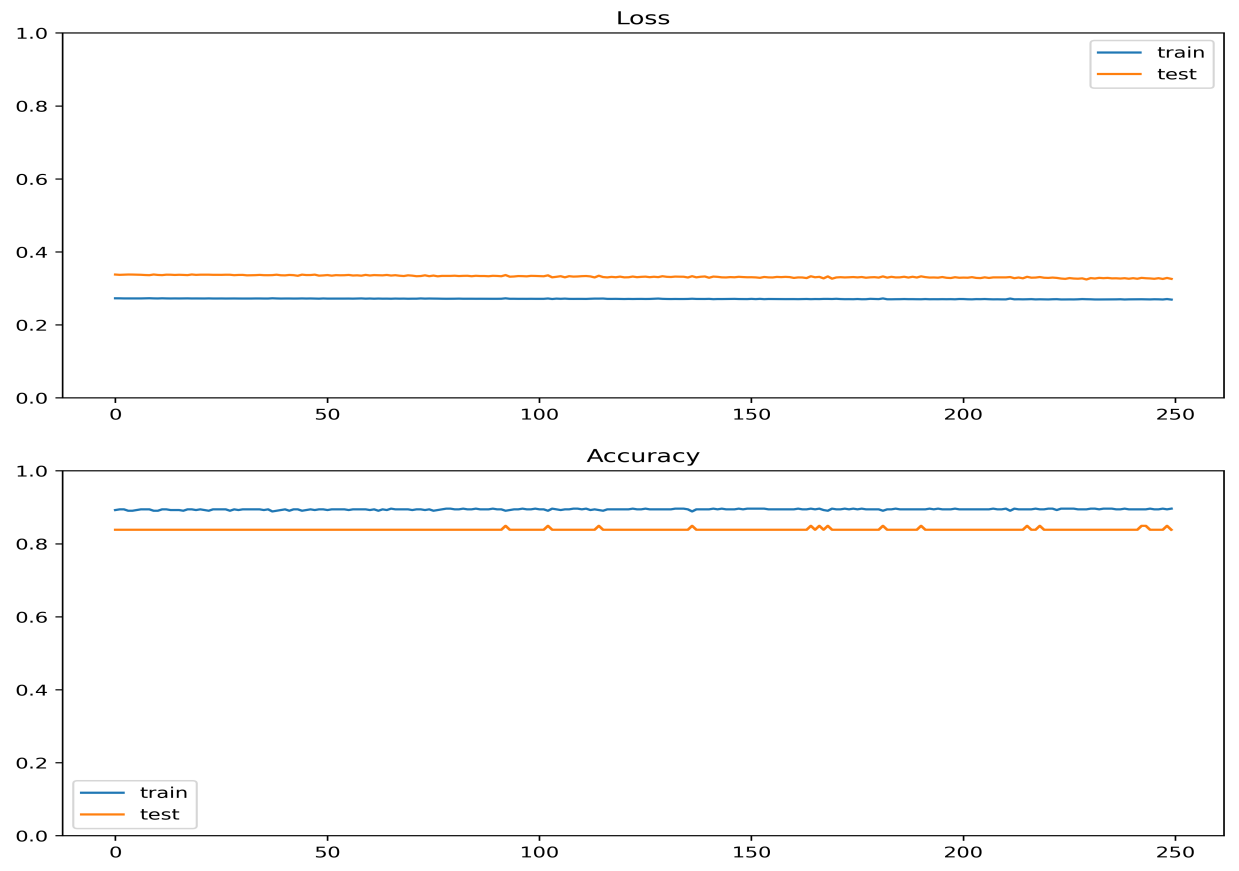


Epoch 1000 batch 2 olarak öğrenme yapıldığında test verisindeki dalgalanmanın tamamen gittiğini görüyoruz.Önceki yüksek batch verilerinde aşırı öğrenmeye sebep olduğu grafikten görülmüştü.Burada daha doğru bir öğrenme modeli oluşturduğumuz görülmektedir.



Batch:16 , epoch:500

Yukardaki grafikte görüldüğü gibi 250 iterasyondan sonra hata ve accuracy değişmedinden epoch verisini 250 yapmak yeterli olacaktır.



Yukarda görüldüğü gibi 250 epoch yeterli olmuş durumdadır.