YILDIZ TEKNİK ÜNİVERSİTESİ
KİMYA-METALÜRJİ FAKÜLTESİ
MATEMATİK MÜHENDİSLİĞİ BÖLÜMÜ

BİLGİSAYAR TABANLI ÖĞRENEN SİSTEMLER

HIGGS VE RCV1 DATA SETLERİ İÇİN BİR İNCELEME

Ders Yürütücüsü: Doç. Dr. Nilgün GÜLER BAYAZIT

21052049, Hümeyra BEDİZ

İstanbul, 2025

İÇİNDEKİLER

1. GİRİŞ

Raporun bu kısmında ödevde kullanılan veri setleri tanıtılacaktır. Verilerin analizi için oluşturulan kodlar Google Colab'de Python 3 ile yazılmıştır.

1.1. HIGGS (UCI) Veri Seti

Higgs veri kümesi, Higgs Bozonu üreten olaylar ile üretmeyen olaylardan oluşan 11 milyon simüle edilmiş parçacık çarpışmasını içeren büyük ölçekli bir veri kümesidir. Her olay, bozunma sonucunda ortaya çıkan parçacıkların yörüngelerini ve özelliklerini tanımlayan 28 nitelik ile temsil edilir. Bu veriler, İsviçre'nin Cenevre kenti yakınlarındaki CERN'de bulunan Büyük Hadron Çarpıştırıcısı (LHC) üzerindeki ATLAS dedektöründe gerçekleşen parçacık çarpışmalarının gerçekçi simülasyonlarından elde edilmiştir.

Higgs veri seti, düşük boyutlu fakat öznitelikler arası ilişkileri doğrusal olmayan (non-linear) bir yapıdır. Bu ödevde, StandardScaler ile normalize edilen ve 20000 olaydan oluşan bir alt kümeyle çalışılmıştır.

1.2. RCV1 (Reuters) Veri Seti

Reuters Corpus Volume 1 (RCV1), Reuters Ltd. tarafından 1996–1997 yılları arasında yayımlanan yaklaşık 800.000 haber metninden oluşan geniş bir veri koleksiyonudur. Haber içerikleri ekonomi, siyaset, spor, teknoloji ve finans gibi çok çeşitli alanları kapsamaktadır.

Bu ödev için 10000 haberden oluşan bir alt küme incelenmiştir. Veri seti yüksek boyutlu olup seyreltik (sparse) bir yapıya sahiptir. TF-IDF vektörleştirme kullanılarak sayısal forma dönüştürülen veri setiyle çalışma yapıldı.

2. MLP ANALİZİ

Bu bölümde, verilen ödevdeki Görev 1 bağlamında elde edilen teknik çıkarımlar sunulmuştur.

2.1. Aktivasyon Fonksiyonu ve Optimizer Karşılaştırmaları

İnceleme yapmak üzere yazılan kodlar çalıştırıldığında aşağıdaki grafik ortaya çıkmaktadır.

Tablo 2.1 Higgs veri seti için aktivasyon fonksiyonu – optimizer ikilisi sonuçları

RELU	
SGD	0.685674525
MOMENTUM	0.685083142
RMSPROP	0.721083038
ADAM	0.637061269

TANH	
SGD	0.68166557
MOMENTUM	0.685758707
RMSPROP	0.751316617
ADAM	0.625420822

SİGMOİD	
SGD	0.679510864
MOMENTUM	0.680151847
RMSPROP	0.637814887
ADAM	0.617575592

Tablo 2.1'den görüldüğü üzere kayıp (loss) üzerinde bir değerlendirme yapıldığında Sigmoid aktivasyon fonksiyonu – ADAM (Adaptive Moment Estimation) optimizeri ikilisi diğer opsiyonlara göre daha iyi sonuç vermiştir.

Tablo incelendiğinde Sigmoid aktivasyon fonksiyonu diğer optimizerler ile de daha az hata vermiştir. Bunun sebebi, Sigmoid'in düşük boyutlu veri setlerinde düzgün ve açık sınırlar belirleyebilmesidir. Sigmoid, girişteki (input) değerlerini (0,1) arasında tutar. Bu, modelin eğitim esnasında daha dengeli olmasını ve kaybın (loss) daha tutarlı düşmesini sağlar.

Düşük boyutlarda, ReLU değerleri bazen negatif bölgeye düşebilir ve tamamen kaybolabilir. Ağdaki nöronları ciddi bir kesimi bu biçimde öldüğünde Ölü ReLU – Dying ReLU sorunu ortaya çıkar ve model verideki ilişkileri öğrenemez hale gelir. Bu sırada, Sigmoid çok küçük de olsa sinyal üretir ve eğitimin devamlılığını sağlar.

Küçük gradyanlı Sigmoid, ADAM optimizeri ile güçlenir. Çünkü ADAM optimizerinde, her bir parametre önceki gradyanlara ve onların büyüklüklerine bağlı olarak kendi öğrenme oranına sahiptir. Bu sayede, Sigmoid'in çok küçük gradyanlarını dengeleyebilir.

Higgs veri setinin sadece optimizerler üzerindeki etkisinin görülmesi için üç farklı optimizerin değerlendirildiği bir kod yazıldı ve kod çalıştırıldığında aşağıdaki sonuçlar ortaya çıktı.

Tablo 2.2 RCV1 veri seti için aktivasyon fonksiyonu – optimizer ikilisi sonuçları

RELU	
SGD	0.308365312
MOMENTUM	0.317169694
RMSPROP	0.036255712
ADAM	0.070789281

TANH	
SGD	0.308374337
MOMENTUM	0.338488429
RMSPROP	0.077539693
ADAM	0.035696992

SİGMOİD	
SGD	0.322556316
MOMENTUM	0.34979238
RMSPROP	0.091455284
ADAM	0.034036649

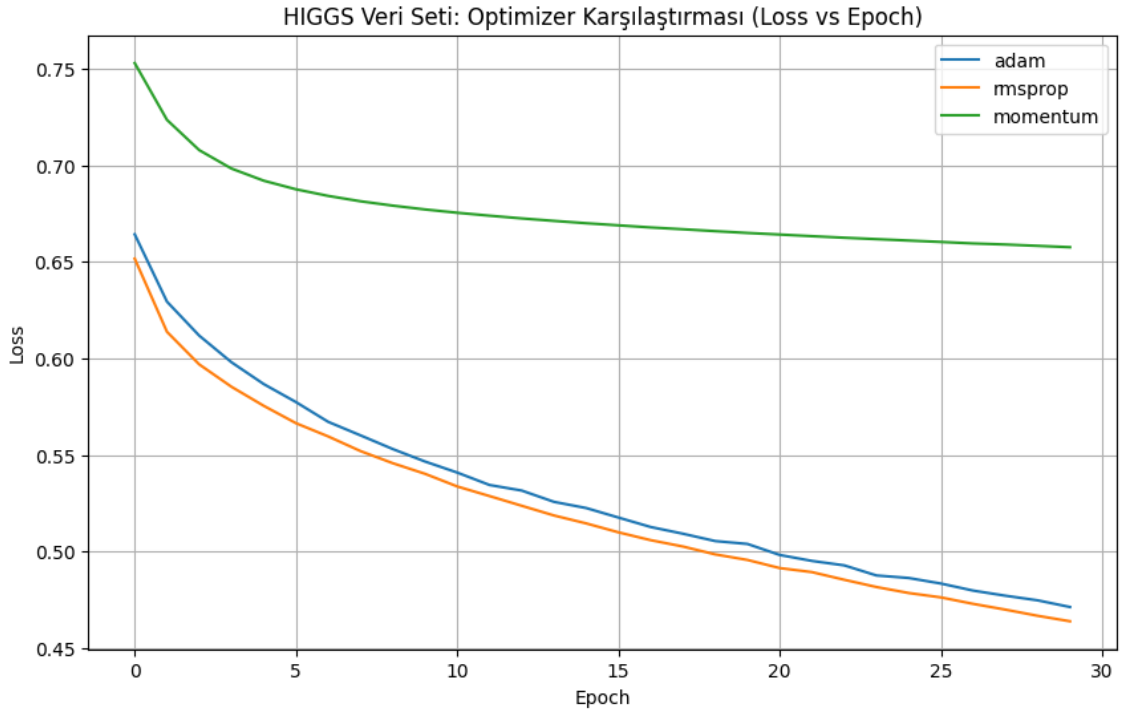
Higgs veri setinde olduğu gibi, RCV1 veri setinde de en başarılı kombinasyon Sigmoid + Adam ikilisir. Bu ikiliyi çok yakın bir değerle Tanh + ADAM takip etmektedir.

Sigmoid'in yapısı gereği kaybolan gradyan riski taşımasına rağmen uygun optimizerle belirli veri setlerinde çok iyi sonuçlar verebildiği görülmüştür.

İki veri seti için de konuşacak olursak en iyi optimizer ADAM'dır. Çünkü aktivasyon fonksiyonlarıyla bir uyum yakalamış ve üç aktivasyon fonksiyonuyla da dengeli ve düşük sonuçlar vermiştir.

2.2. Optimizer Karşılaştırmaları

2.1'de aktivasyon fonksiyonu – optimizer karşılaştırması yapılmıştı ve en iyi optimizer tablo yorumlarına göre ADAM optimizeri seçilmişti. Bu alt başlıkta, öğrenme oranı ve uygulanacak aktivasyon fonksiyonu sabit bırakılarak optimizerlerin davranışları incelenmiştir.



Şekil 2.1 Higgs veri seti üzerinde optimizer karşılaştırması

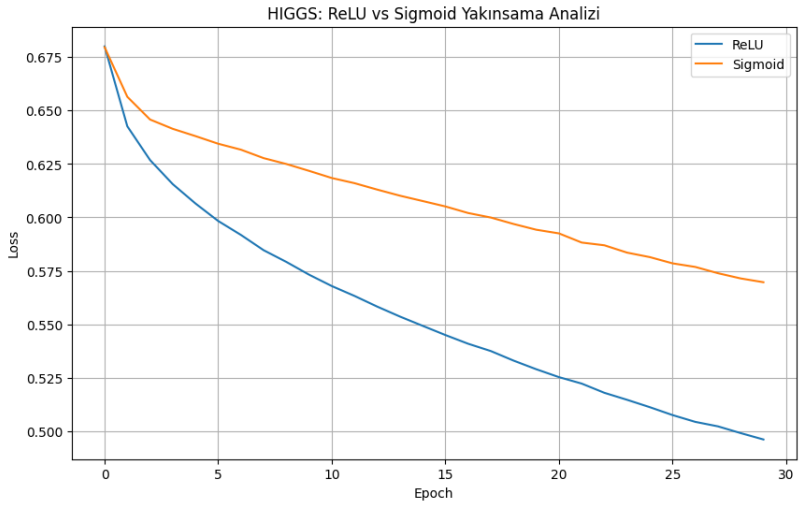
RMSProp tüm epochlar boyunca diğer optimizerlere göre daha düşük kayıp değerlerine ulaşmıştır. ADAM da RMSProp'un biraz gerisinde kalmasına rağmen tatmin edici sonuçlar vermiştir. ADAM ve RMSProp, uyarlanabilir öğrenme oranına sahip optimizerlerdir. Her parametre için ayrı öğrenme oranı sunarlar. Bu da kaybı azaltmıştır.

Momentum'un diğerlerine göre çok yüksek kayıpta kalması, sabit öğrenme oranının (learning_rate) bu veri seti için yetersiz olduğu ya da modelin daha fazla epoch'a ihtiyaç duyduğu anlamına gelir.

Ayrıca grafikteki üç çizgide de az dalgalanma vardır. Bu da öğrenme oranlarının uygun seçildiğini, modelin dengeli bir biçimde eğitildiğini gösterir.

2.3. Aktivasyon Fonksiyonları Karşılaştırmaları

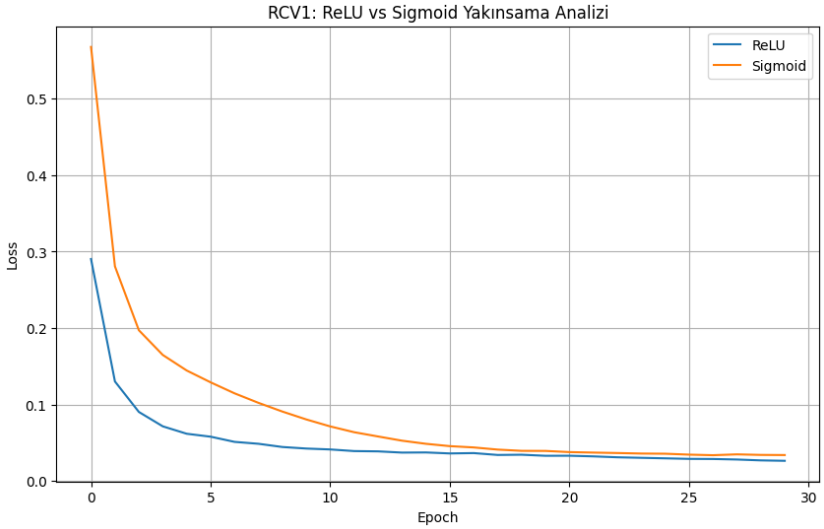
2.1'de aktivasyon fonksiyonu – optimizer karşılaştırması yapılmıştı ve en iyi aktivasyon fonksiyonu tablo yorumlarına göre Sigmoid aktivasyon fonksiyonu seçilmişti. Bu alt başlıkta, öğrenme oranı ve optimizer sabit tutularak Relu ve Sigmoid aktivasyon fonksiyonlarının davranışları incelenecektir.



Şekil 2.2 Higgs veri seti için ReLU ve Sigmoid aktivasyon fonksiyonları grafiği

Eğitim her iki aktivasyon fonksiyonu için aşağı yukarı aynı kayıp değerleriyle başlasa da ReLU genel olarak Sigmoid'e göre daha hızlı düşüş sergilemiştir.

Bunun en önemli sebeplerinden biri Sigmoid'in kaybolan gradyan sorunundan dolayı zayıf kalmasıdır. Sigmoid'in türevleri belirli bir noktadan sonra çok küçülür ve ağırlık güncellemeleri yavaşlar. Kayıpları bir süre sonra zar zor azalması bu yüzdendir. Doğrusal yapısından ötürü ReLU'nun ağırlık güncellemeleri daha dengelidir. Dolayısıyla, ReLU, vanishing gradient engeline takılmadan daha az hatayla öğrenebilir.

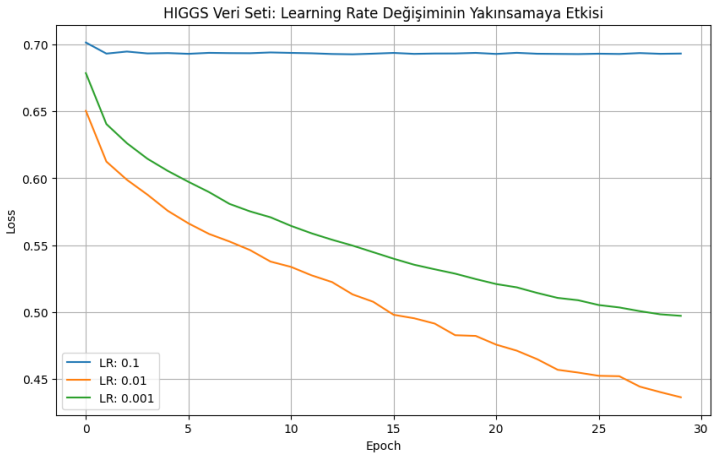


Şekil 2.3 RCV1 veri seti için ReLU ve Sigmoid aktivasyon fonksiyonları grafiği

Eğitim, iki aktivasyon fonksiyonu arasındaki büyük kayıp farkıyla başlamış olsa da Sigmoid son epochlarda ReLU'ya yaklaşmıştır. Higgs veri setindeki gibi açık ara fark bulunmamasına rağmen ReLU, Sigmoid'e göre daha az hata üretmeyi başarmıştır.

2.4. Öğrenme Oranı Karşılaştırmaları

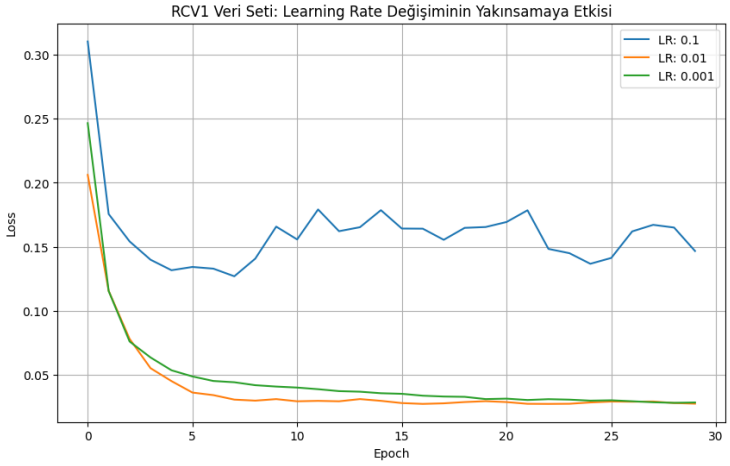
Bu alt başlıkta, değişen öğrenme oranlarına karşılık gelen davranışları incelenecektir.



Şekil 2.4 Higgs veri seti için LR grafiği

Higgs veri seti üzerinde inceleme yapıldığında, LR = 0.1 için modelin kaybı azaltmada başarısız olduğu görülmüştür. Gradyan adımları büyük olduğundan model, minimuma ulaşmadan hedefi ıskalayabilir. LR=0.001 olduğunda ise gradyan adımları çok küçüktür. Epoch sayısına bağlı olarak optimuma erişemeyebilir.

LR=0.01, grafiğe göre en iyi sonucu veren değerdir. Çünkü ne optimum noktasını kaçıracak kadar yüksek ne de optimuma gelemeden eğitimi durduracak kadar küçüktür.

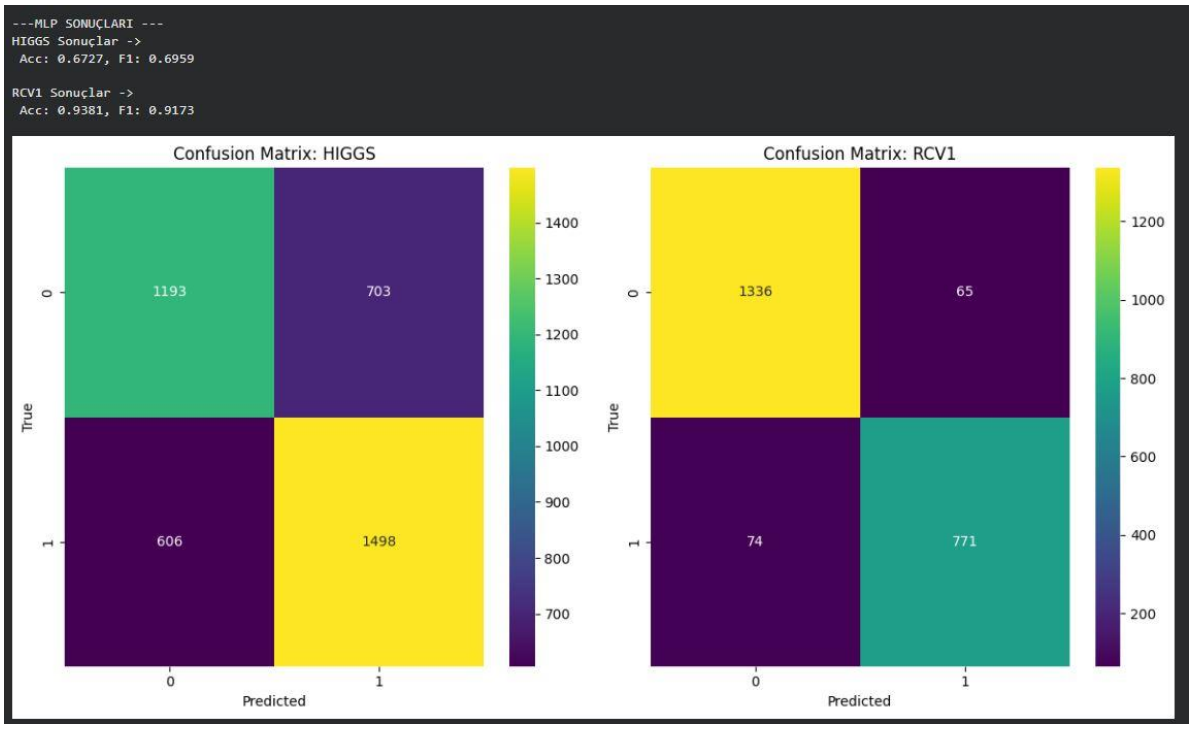


Şekil 2.5 RCV1 veri seti için LR grafiği

RCV1 için de benzer yorumlar yapılabilir. LR = 0.1 için grafiğin fazla zikzaklı olması öğrenme oranının bu veri için çok büyük olduğunu gösterir. RCV1 veri seti üzerinde yapılan hiper parametre testlerinde, 0.01 öğrenme oranı en efektif yakınsamayı sağlamıştır.

2.5. MLP için Performans Metrikleri

MLP için elde edilen metrikler aşağıda verilmiştir.



Şekil 2.6 Verilen datasetleri için MLP performans metrikleri

Doğruluk (accuracy) ve F1 skorları incelendiğinde modelin, RCV1 veri setinde çok başarılı olduğu görülürken, MLP, Higgs veri seti için yetersiz kalmıştır.

RCV1 veri setindeki verilerin neredeyse hepsini doğru tahminleyen MLP, yüksek F1 skoruyla da hassasiyet (precision) ve duyarlılık (recall) dengesinin yüksek olduğunu, modelin nadir sınıfları bile yakalayabildiğini gösterir.

Higgs veri setindeki başarısızlığın nedeni, sınıflar arasındaki yüksek benzerliğe (overlap) işarettir. Higgs veri setinde performansı artırmak için katman sayısı (hidden layer) artırılabilir veya eğitim süresi daha uzun tutulabilir.

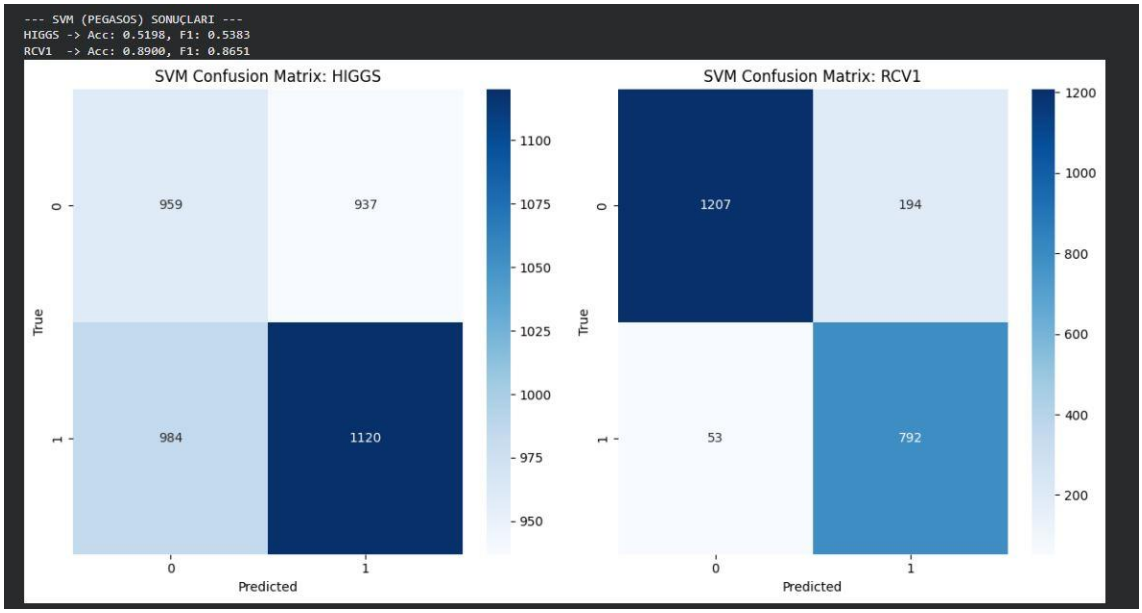
3. SVM (PEGASOS) ANALİZİ

3.1. SVM Performans Metriklerinin Değerlendirilmesi

Pegasos (Primal Estimated sub-GrAdient SOlver for SVM), Destek Vektör Makineleri-DVM (Support Vector Machines – SVM) için geliştirilen iteratif çözümleme yöntemidir.

SVM'nin aksine Pegasos SGD (Stochastic Gradient Descent) kullanarak primal denklemleri üzerinde çalışır.

Avantajı: Özellikle büyük veri setlerinde (Big Data), verinin tamamını belleğe yüklemeden verimli bir şekilde eğitilebilir. "Hızlı ve etkili" olmasıyla bilinir.



Şekil 3.1 SVM performans metrikleri

Doğrusal bir ayırıcı olan SVM (Pegasos) algoritması, MLP algoritmasında olduğu gibi RCV1 veri seti için daha iyi sonuçlar elde etmiştir. Küçük boyutlu, karmaşık ve doğrusal olmayan (non-lineer) verilerden olan Higgs için sonuçlar, algoritmanın neredeyse 2 tahminden 1'ini yanlış yaptığını gösterir.

3.2. MLP ve SVM karşılaştırması

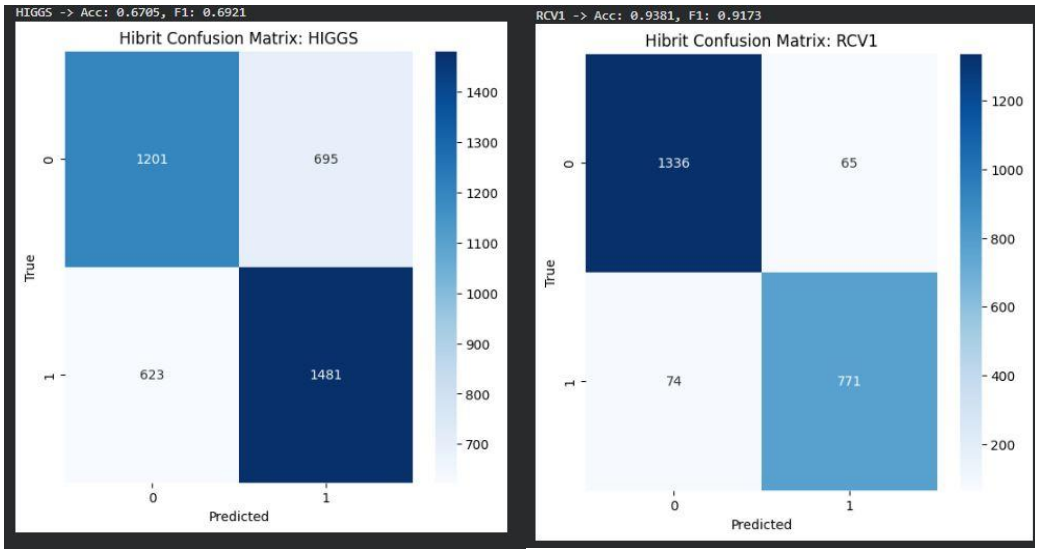
Şekil 3.1'deki SVM performans metriğindeki değerler de incelendiğinde hem Higgs hem de RCV1 veri seti için doğruluk ve F1 skoru değerleri MLP'nin altında kalmıştır.

Bunun sebebi, MLP'nin gizli katmanlar sayesinde verideki karmaşık ilişkileri çözümleyerek modeli eğitebilmesidir.

4. HİBRİT MODEL VE HİBRİT MODELİN TEKİL MODELLERLE KARŞILAŞTIRILMASI

Hibrit model kurulurken model istifleme (model stacking) kullanıldı. Model stacking'te öncelikle karıştırılacak modeller ayrı ayrı eğitilirler. Sonrasında ise doğrulama setleri için MLP'den olasılık değerleri, SVM'den ise karar skorları alınır. Bu değerlerden yola çıkılarak [MLP_Score, SVM_Score] biçiminde iki boyutlu yeni bir öznitelik (feature) vektörü oluşturulur. Bu yeni öznitelik vektörüne Meta-Öznitelik vektörü denir.

Oluşturulan yeni veri kümesi Logistik Regresyon kullanılarak eğitilir ve sonuçlar elde edilir.



Şekil 4.1 Hibrit model performans metrikleri

Hibrit modelin metrik değerleri incelendiğinde MLP'ye yakın sonuçlar görülür. Hatta RCV1 için hibrit model, MLP'nin RCV1 veri seti için ürettiği değerlerin aynısını üretmiştir.

Makine öğrenmesi literatürü bu durumu Performans Doygunluğu (Performance Saturation) ve Model Baskınlığı (Model Dominance) olarak adlandırır. Yüksek boyutlu ve seyrek (sparse) yapıya sahip RCV1'in sınıfların doğrusal ayrılabilirliğini artırması, MLP modelinin veri setindeki tüm farklı örnekleme kalıplarını en yüksek seviyede kendi başına yakalamasına olanak tanımaktadır.

Meta-model olarak işlev gören Lojistik Regresyon, SVM'den gelen karar skorlarının, MLP'nin olasılıksal skorlarına kıyasla yeni bir bilgi kazancı sağlayamamış olabilir ve buna bağlı olarak ağırlıklar MLP'nin lehine güncellenmiş olabilir.

Sonuç olarak, MLP modelini RCV1 üzerinde doyum noktasına ulaşması, hibrit modelde yapının SVM bileşeninin ek hata iyileştirmesi için gereksiz hale gelmesine neden olmuş, böylece her iki modelin çıktıları metrik bazda eşitlenmiştir.

Özetle, tüm metrikler iki veri seti için de değerlendirildiğinde en iyi sonuçları MLP vermiştir. SVM, hibrit model ve MLP'nin çok gerisinde kalmıştır. Hibrit modelin RCV1 veri seti için MLP ile aynı değerleri üretmesi en iyi model olarak öne çıkmasına yetmemiştir.

Yapay Zekâ ve Dış Kaynak Kullanım Beyanı

Kullandığım Yapay Zekâ Araçları: Google Gemini

Kullanım Amacı: Kod içindeki hatayı ayıklama, bilgi eksikliği nedeniyle yazılamayan kod bloklarının yazılması, raporun daha teknik temelle hazırlanabilmesi için kavram öğrenme, raporun dilinin düzgün olması amacıyla paraphrase şartıyla rapora yazılabilecek teknik paragraflar oluşturma ve paragrafları inceleme.

Kullandığım Dış Kaynaklar:

- **GitHub:** MLP'yi kodlamaya başlamadan önce nasıl yapıldığı hakkında bilgi edinmek üzere kullanıldı.

https://github.com/Fodark/mlp-python (Erişim Tarihi: 02/01/2026)

- **Medium:** GitHub ile aynı amaçla kullanıldı.

https://elcaiseri.medium.com/building-a-multi-layer-perceptron-from-scratch-with-numpy-e4cee82ab06d (Erişim Tarihi: 02/01/2026)

- **GeeksforGeeks:** Raporlamada yardımcı olması adına bazı optimizerlerin tanımının öğrenilmesi amacıyla kullanıldı.

https://www.geeksforgeeks.org/deep-learning/adam-optimizer/
https://www.geeksforgeeks.org/deep-learning/rmsprop-optimizer-in-deep-learning/
(Erişim Tarihi: 05/01/2026)

- **TTIC> Publications:** Raporlamada yardım olması adına SVM (Pegasos) tanımı incelendi.

https://home.ttic.edu/~nati/Publications/PegasosMPB.pdf (Erişim Tarihi: 06/01/2026)