

Çanakkale Onsekiz Mart Üniversitesi, Mühendislik Fakültesi,  
Bilgisayar Mühendisliği Akademik Dönem 2022-2023  
Ders: BLM-4014 Yapay Sinir Ağları/Bahar Dönemi  
ARA SINAV SORU VE CEVAP KAĞIDI  
Dersi Veren Öğretim Elemanı: Dr. Öğretim Üyesi Ulya Bayram

Öğrenci Adı Soyadı: Ali Yetgin  
Öğrenci No: 210401006

14 Nisan 2023

### Açıklamalar:

- Vizeyi çözüp, üzerinde aynı sorular, sizin cevaplar ve sonuçlar olan versiyonunu bu formatta PDF olarak, Teams üzerinden açtığım assignment kısmına yüklemeniz gerekiyor. Bu bahsi geçen PDF'i oluşturmak için LaTeX kullandıysanız, tex dosyasının da yer aldığı Github linkini de ödevin en başına (aşağı url olarak) eklerseniz bonus 5 Puan! (Tavsiye: Overleaf)
- Çözümlerde ya da çözümlerin kontrolünü yapmada internetten faydalanmak, ChatGPT gibi servisleri kullanmak serbest. Fakat, herkesin çözümü kendi emeğinden oluşmak zorunda. Çözümlerinizi, cevaplarınızı aşağıda belirttiğim tarih ve saate kadar kimseyle paylaşmayınız.
- Kopyayı önlemek için Github repository'lerinizin hiçbirini **14 Nisan 2023, saat 15:00'a kadar halka açık (public) yapmayınız!** (Assignment son yükleme saati 13:00 ama internet bağlantısı sorunları olabilir diye en fazla ekstra 2 saat daha vaktiniz var. **Fakat 13:00 - 15:00 arası yüklemelerden -5 puan!**)
- Ek puan almak için sağlayacağımız tüm Github repository'lerini **en geç 15 Nisan 2023 15:00'da halka açık (public) yapmış olun linklerden puan alabilmek için!**
- **14 Nisan 2023, saat 15:00'dan sonra gönderilen vizeler değerlendirilmeye alınmayacak, vize notu olarak 0 (sıfır) verilecektir!** Son anda internet bağlantısı gibi sebeplerden sıfır almayı önlemek için assignment kısmından ara ara çözümlerinizi yükleyebilirsiniz yedekleme için. Verilen son tarih/saatte (14 Nisan 2023, saat 15:00) sistemdeki en son yüklü PDF geçerli olacak.
- Çözümlerin ve kodların size ait ve özgün olup olmadığını kontrol eden bir algoritma kullanılacaktır. Kopya çektiği belirlenen vizeler otomatikman 0 (sıfır) alacaktır. Bu nedenle çözümlerinizi ve kodlarınızı yukarıda sağladığım gün ve saatlere kadar kimseyle paylaşmayınız.
- Bu vizeden alınabilecek en yüksek not 100'dür. Toplam aldığımız puan 100'ü geçerse, aldığımız not 100'e sabitlenecektir.
- LaTeX kullanarak PDF oluşturanlar öz geçmişlerine LaTeX bildiklerini de eklemeyi unutmasınlar :)
- Bu vizedeki soruların çözümleri ve tex dosyası için istediğiniz kadar sayıda Github repository'si oluşturabilirsiniz. Sadece yukarıda belirttiğim tarihlerde (14 Nisan 2023 saat 15:00 ile 15 Nisan 2023 saat 15:00 arasında) public yapmayı/halka açmayı ve Github profilinizi de öz geçmişinize eklemeyi unutmayın :)
- Bilgisayarınıza hiçbir program kurmadan, Overleaf, Google Colab, Kaggle gibi ortamlar üzerinden bu vizeyi tamamlamanız mümkün. İyi çalışmalar!!

(Ekstra 5 puan) <https://github.com/aliyetgin/yapaysiniraglarivize>

# 1 (Toplam 10 Puan) Hopfield Networks:

**1.1 (2 Puan) ChatGPT'ye size Hopfield Network'ler ile ilgili detaylı bilgi vermesini isteyin. Sorduğunuz soruyu, ChatGPT'nin çıkarttığı cevabı aşağı koyun. Bu cevapta anlamadığınız terimleri de ChatGPT'ye sorup, sorduğunuz soruları ve ChatGPT'nin verdiği cevapları da sırayla aşağı koyun.**

Soru: Bana Hopfield Network'le ilgili detaylı bilgi verir misin?

Cevap: Hopfield ağı, sınırlı kapasiteli bir yapay sinir ağıdır. Nöronlar, tüm düğümlerin birbirleriyle bağlantılı olduğu bir graf yapısında düzenlenir. Ağ, asenkron bir şekilde çalışır ve nöronların durumlarını sigmoid fonksiyonu kullanarak hesaplar.

Hopfield ağları, hafızayı saklamak, hatırlamak ve yeniden çağırmak için kullanılır. Öğrenme sırasında, Hopfield ağı, veri kümesindeki öğeler arasındaki benzerliklere dayalı olarak nöron bağlantılarını öğrenir. Bu sayede, Hopfield ağı öğrenilen bir deseni hatırlayabilir ve bozukluğu olan bir desene benzer bir desenle eşleştirebilir.

Ayrıca, Hopfield ağları, aynı zamanda birçok uygulama için de kullanılabilir. Örneğin, veri sıkıştırma, veri kurtarma ve kalıpların tanınması işlemlerinde kullanılabilirler.

Hopfield ağı, aşağıdaki özelliklere sahiptir:

- Sınırlı kapasiteli: Hopfield ağı, sınırlı sayıda deseni hatırlayabilir. Bu kapasite, ağın boyutuna ve desenlerin karmaşıklığına bağlıdır. Enerji fonksiyonu: Hopfield ağı, bir enerji fonksiyonu kullanır. Bu fonksiyon, ağın mevcut durumunun bir ölçüsüdür ve hatırlama işleminde kullanılır.
- Asenkron çalışma: Hopfield ağı, nöronları asenkron bir şekilde hesaplar. Bu, ağın daha hızlı çalışmasına ve daha az bellek kullanmasına olanak tanır.
- Geri besleme: Hopfield ağı, geri beslemeli bir yapıya sahiptir. Bu yapı, ağın hatırlama işlemi için önemlidir.

Hopfield ağları, öğrenme sırasında Hebb kuralını kullanırlar. Bu kurala göre, eşzamanlı olarak ateşleyen nöronlar arasındaki bağlantılar güçlenir. Bu sayede, Hopfield ağı, öğrenilen desenleri hatırlayabilir ve bozukluğu olan bir desene benzer bir desenle eşleştirebilir.

Ancak, Hopfield ağları sadece basit desenlerin hatırlanması için kullanılabilir. Daha karmaşık desenlerin hatırlanması için daha büyük ve daha karmaşık ağlar gereklidir.

Sorular:

1-Enerji fonksiyonu nedir ?

Cevap: Enerji fonksiyonu, fiziksel sistemlerin enerji durumunu ifade eden matematiksel bir fonksiyondur. Bu fonksiyon, sistemin içinde bulunduğu durumun enerjisini hesaplar.

Yapay zeka ve makine öğrenmesi gibi alanlarda, enerji fonksiyonu, bir modelin öğrenme hedeflerini ifade etmek için kullanılabilir. Bu amaçla, enerji fonksiyonu, modelin öğrenmek istediği hedeflerin bir ölçüsü olarak kullanılabilir.

Örneğin, Hopfield ağı gibi bazı yapay sinir ağı modelleri, bir enerji fonksiyonu kullanır. Bu ağlar, bir veri örüntüsünü öğrenmek ve daha sonra benzer örüntüleri tanımak için kullanılır. Bu süreçte, enerji fonksiyonu, ağın öğrenme ilerlemesini ölçmek ve doğru sonuçları üretmek için kullanılır.

Enerji fonksiyonu, birçok farklı matematiksel formda ifade edilebilir. Bu formlar, spesifik bir problem için en uygun olanı seçmek için kullanılır. Enerji fonksiyonu, sistemin mevcut durumunu belirleyen değişkenlerin bir fonksiyonu olarak ifade edilir ve bu değişkenlerin değerlerine göre farklı enerji seviyeleri atar. Enerji fonksiyonunun en düşük enerjili durumu, sistemin en istikrarlı durumunu ifade eder.

2-Hebb kuralı nedir ?

Cevap: Hebb kuralı, sinir biliminde kullanılan bir öğrenme kuralıdır. Bu kural, bir nöronun başka

bir nöronla bağlantısının gücünü değiştiren bir öğrenme kuralıdır. Kısaca, "birlikte ateşleyen hücreler birlikte telaffuz edilir" şeklinde özetlenebilir.

Hebb kuralı, "eşzamanlı ateşleme, birlikte ateşleme" prensibine dayanır. Bu kurala göre, iki nöron eşzamanlı olarak ateşlenirse, bağlantılarını güçlendirmeleri beklenir. Bunun sebebi, bir nöronun diğerini ateşlediği zaman, sinaptik bağlantının güçlenmesiyle iki nöronun birbirine daha sıkı bağlanmasıdır.

Örneğin, bir nöron X, bir diğeri Y ile bağlantılı olsun. Nöron X, nöron Y'yi ateşlediğinde, Hebb kuralına göre X nöronunun Y nöronuyla olan bağlantısı güçlenir. Bu durum, nöron X'in tekrar ateşlediğinde, Y nöronunun ateşlemesine daha yatkın hale gelmesine yol açar.

Hebb kuralı, beyindeki öğrenme sürecinin açıklanmasına yardımcı olur. Bu kural, beyindeki sinir ağları arasındaki bağlantıların nasıl şekillendiği ve nasıl öğrenme süreçleri gerçekleştirdiği konusunda önemli bir ipucu verir. Yapay sinir ağları ve derin öğrenme modellerinde de Hebb kuralı, nöronların bağlantılarını güçlendirmek için kullanılan bir öğrenme kuralıdır.

3-Geri beslemeli yapı(yapay sinir ağları) nedir??

Cevap: Geri beslemeli yapı yapay sinir ağları, bir çıktının aynı zamanda sistemin girdilerine de etki ettiği yapay sinir ağlarıdır. Bu yapıda, ağın çıktısı bir geri besleme döngüsü aracılığıyla tekrar ağın girdisine yönlendirilir. Bu geri besleme döngüsü, ağın davranışını önceden belirlemek için kullanılabilir.

Geri beslemeli yapı yapay sinir ağları, birçok farklı alanda kullanılır. Örneğin, geri beslemeli yapı yapay sinir ağları, kontrol sistemlerinde kullanılabilir. Bu yapay sinir ağları, geri beslemeli bir kontrol döngüsü aracılığıyla sistemin davranışını ayarlamak için kullanılır.

Ayrıca, geri beslemeli yapı yapay sinir ağları, sıralı verileri işlemek için de kullanılabilir. Örneğin, doğal dil işleme ve konuşma tanıma sistemlerinde geri beslemeli yapı yapay sinir ağları kullanılır. Bu yapay sinir ağları, bir önceki adımdan gelen bilgileri kullanarak gelecekteki adımlar için tahminler yapabilirler.

Geri beslemeli yapı yapay sinir ağları, öğrenme sürecinde de kullanılabilirler. Bu yapay sinir ağları, geri beslemeli bağlantıları kullanarak önceki örneklerden öğrenilen bilgileri yeni örnekler için kullanabilirler. Bu özellikleri nedeniyle, geri beslemeli yapı yapay sinir ağları, karmaşık ve dinamik problemler için etkili bir araçtır.

**1.2 (8 Puan) ChatGPT'nin açıklamalarından ne anladığınızı buraya kendi kelimelerinizle özetleyin. Örneğin "Hopfield Network nedir? Ne işe yarar? Neden bilmemiz gerekir bunu? Gerçek hayatta kullanılıyor mu?" gibi. Anlamadığınız kısımlar varsa ve ChatGPT'nin açıklamaları yeterli gelmezse internet dahil farklı kaynaklardan araştırıp, bilgilerin doğruluğunu sorgulamakta serbestsiniz. Konuyu doğru anlayıp anlamamanız puana etki edecektir. Kullandığınız kaynakları da belirtin!**

Cevap buraya gelecek...

## 2 (Toplam 10 Puan) Boltzman Machine:

**2.1 (2 Puan) ChatGPT'ye size Boltzman Machine ilgili detaylı bilgi vermesini isteyin. Sorduğunuz soruyu, ChatGPT'nin çıkarttığı cevabı aşağı koyun. Bu cevapta anlamadığınız terimleri de ChatGPT'ye sorup, sorduğunuz soruları ve ChatGPT'nin verdiği cevapları da sırayla aşağı koyun.**

Cevap buraya gelecek...

**2.2 (8 Puan) ChatGPT'nin açıklamalarından ne anladığınızı buraya kendi kelimelerinizle özetleyin. Örneğin "Boltzman Machine nedir? Ne işe yarar? Neden bilmemiz gerekir bunu? Gerçek hayatta kullanılıyor mu?" gibi. Anlamadığınız kısımlar varsa ve ChatGPT'nin açıklamaları yeterli gelmezse internet dahil farklı kaynaklardan araştırıp,**

bilgilerin doğruluğunu sorgulamakta serbestsiniz. Konuyu doğru anlayıp anlamamanız puana etki edecektir. Kullandığınız kaynakları da belirtin!

Cevap buraya gelecek...

### 3 (Toplam 10 Puan) Markov Assumption ve Markov Chain:

**3.1 (2 Puan)** ChatGPT'ye size Markov assumption ve Markov Chain ilgili detaylı bilgi vermesini isteyin. Sorduğunuz soruyu, ChatGPT'nin çıkarttığı cevabı aşağı koyun. Bu cevapta anlamadığınız terimleri de ChatGPT'ye sorup, sorduğunuz soruları ve ChatGPT'nin verdiği cevapları da sırayla aşağı koyun.

Cevap buraya gelecek...

**3.2 (8 Puan)** ChatGPT'nin açıklamalarından ne anladığınızı buraya kendi kelimelerinizle özetleyin. Örneğin “Markov assumption ve Markov Chain nedir? Ne işe yarar? Neden bilmemiz gerekir bunu? Gerçek hayatta kullanılıyor mu?” gibi. Anlamadığınız kısımlar varsa ve ChatGPT'nin açıklamaları yeterli gelmezse internet dahil farklı kaynaklardan araştırıp, bilgilerin doğruluğunu sorgulamakta serbestsiniz. Konuyu doğru anlayıp anlamamanız puana etki edecektir. Kullandığınız kaynakları da belirtin!

Cevap buraya gelecek...

### 4 (Toplam 20 Puan) Feed Forward:

- Forward propagation için, input olarak şu X matrisini verin (tensöre çevirmeyi unutmayın):

$$X = \begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \end{bmatrix} \text{ Satırlar veriler (sample'lar), kolonlar öz nitelikler (feature'lar).}$$

- Bir adet hidden layer olsun ve içinde tanh aktivasyon fonksiyonu olsun
- Hidden layer'da 50 nöron olsun
- Bir adet output layer olsun, tek nöronu olsun ve içinde sigmoid aktivasyon fonksiyonu olsun

Tanh fonksiyonu:

$$f(x) = \frac{\exp(x) - \exp(-x)}{\exp(x) + \exp(-x)}$$

Sigmoid fonksiyonu:

$$f(x) = \frac{1}{1 + \exp(-x)}$$

Pytorch kütüphanesi ile, ama kütüphanenin hazır aktivasyon fonksiyonlarını kullanmadan, formülünü verdiğim iki aktivasyon fonksiyonunun kodunu ikinci haftada yaptığımız gibi kendiniz yazarak bu yapay sinir ağını oluşturun ve aşağıdaki üç soruya cevap verin.

**4.1 (10 Puan)** Yukarıdaki yapay sinir ağını çalıştırmadan önce pytorch için Seed değerini 1 olarak set edin, kodu aşağıdaki kod bloğuna ve altına da sonucu yapıştırın:

```
kod_buraya = None
if kod_buraya:
    devam_ise_buraya = 0

print(devam_ise_buraya)
```

Sonuç da buraya

**4.2 (5 Puan) Yukarıdaki yapay sinir ağını çalıştırmadan önce Seed değerini öğrenci numaranız olarak değiştirip, kodu aşağıdaki kod bloğuna ve altına da sonucu yapıştırın:**

```
kod_buraya = None
if kod_buraya:
    devam_ise_buraya = 0

print(devam_ise_buraya)
```

Sonuç da buraya

**4.3 (5 Puan) Kodlarınızın ve sonuçlarınızın olduğu jupyter notebook'un Github repository'sindeki linkini aşağıdaki url kısmının içine yapıştırın. İlk sayfada belirttiğim gün ve saate kadar halka açık (public) olmasın:**

[www.githublinkiburaya.com](http://www.githublinkiburaya.com)

## 5 (Toplam 40 Puan) Multilayer Perceptron (MLP):

**Bu bölümdeki sorularda benim vize ile beraber paylaştığım Prensesi İyileştir (Cure The Princess) Veri Seti parçaları kullanılacak. Hikaye şöyle (soruyu çözmek için hikaye kısmını okumak zorunda değilsiniz):**

“Bir zamanlar, çok uzaklarda bir ülkede, ağır bir hastalığa yakalanmış bir prenses yaşamış. Ülkenin kralı ve kraliçesi onu iyileştirmek için ellerinden gelen her şeyi yapmışlar, ancak denedikleri hiçbir çare işe yaramamış.

Yerel bir grup köylü, herhangi bir hastalığı iyileştirmek için gücü olduğu söylenen bir dizi sihirli malzemeden bahsederek kral ve kraliçeye yaklaşmış. Ancak, köylüler kral ile kraliçeyi, bu malzemelerin etkilerinin patlayıcı olabileceği ve son zamanlarda yaşanan kuraklıklar nedeniyle bu malzemelerden sadece birkaçının herhangi bir zamanda bulunabileceği konusunda uyarılmışlar. Ayrıca, sadece deneyimli bir simyacı bu özelliklere sahip patlayıcı ve az bulunan malzemelerin belirli bir kombinasyonunun prensesi iyileştireceğini belirleyebilecekmiş.

Kral ve kraliçe kızlarını kurtarmak için umutsuzlar, bu yüzden ülkedeki en iyi simyacıyı bulmak için yola çıkmışlar. Dağları tepeleri aşmışlar ve nihayet "Yapay Sinir Ağları Uzmanı" olarak bilinen yeni bir sihirli sanatın ustası olarak ün yapmış bir simyacı bulmuşlar.

Simyacı önce köylülerin iddialarını ve her bir malzemenin alınan miktarlarını, ayrıca iyileşmeye yol açıp açmadığını incelemiş. Simyacı biliyormuş ki bu prensesi iyileştirmek için tek bir şans varmış ve bunu doğru yapmak zorundaymış. (Original source: <https://www.kaggle.com/datasets/unmoved/cure-the-princess>)

(Buradan itibaren ChatGPT ve Dr. Ulya Bayram'a ait hikayenin devamı)

Simyacı, büyülü bileşenlerin farklı kombinasyonlarını analiz etmek ve denemek için günler harcamış. Sonunda birkaç denemenin ardından prensesi iyileştirecek çeşitli karışım kombinasyonları bulmuş ve bunları bir veri setinde toplamış. Daha sonra bu veri setini eğitim, validasyon ve test setleri olarak üç parçaya ayırmış ve bunun üzerinde bir yapay sinir ağı eğiterek kendi yöntemi ile prensesi iyileştirme ihtimalini hesaplamış ve ikna olunca kral ve kraliçeye haber vermiş. Heyecanlı ve umutlu olan kral ve kraliçe, simyacının prensese hazırladığı ilacı vermesine izin vermiş ve ilaç işe yaramış ve prenses hastalığından kurtulmuş.

Kral ve kraliçe, kızlarının hayatını kurtardığı için simyacıya krallıkta kalması ve çalışmalarına devam etmesi için büyük bir araştırma bütçesi ve çok sayıda GPU'su olan bir server vermiş. İyileşen prenses de kendisini iyileştiren yöntemleri öğrenmeye merak salıp, krallıktaki üniversitenin bilgisayar mühendisliği bölümüne girmiş ve mezun olur olmaz da simyacının yanında, onun araştırma grubunda çalışmaya başlamış. Uzun yıllar birlikte krallıktaki insanlara, hayvanlara ve doğaya faydalı olacak yazılımlar

geliştirmişler, ve simyacı emekli olduğunda prenses hem araştırma grubunun hem de krallığın lideri olarak hayatına devam etmiş.

Prenses, kendisini iyileştiren veri setini de, gelecekte onların izinden gidecek bilgisayar mühendisi prensler ve prensesler başkalarına faydalı olabilecek yapay sinir ağları oluşturmayı öğretilenler diye halka açmış ve sınavlarda kullanılmasını salık vermiş.”

**İki hidden layer’lı bir Multilayer Perceptron (MLP) oluşturun beşinci ve altıncı haftalarda yaptığımız gibi. Hazır aktivasyon fonksiyonlarını kullanmak serbest. İlk hidden layer’da 100, ikinci hidden layer’da 50 nöron olsun. Hidden layer’larda ReLU, output layer’da sigmoid aktivasyonu olsun.**

Output layer’da kaç nöron olacağını veri setinden bakıp bulacaksınız. Elbette bu veriye uygun Cross Entropy loss yöntemini uygulayacaksınız. Optimizasyon için Stochastic Gradient Descent yeterli. Epoch sayınızı ve learning rate’i validasyon seti üzerinde denemeler yaparak (loss’lara overfit var mı diye bakarak) kendiniz belirleyeceksiniz. Batch size’ı 16 seçebilirsiniz.

**5.1 (10 Puan) Bu MLP’nin pytorch ile yazılmış class’ının kodunu aşağı kod bloğuna yapıştırın:**

```
kod_buraya = None
if kod_buraya:
    devam_ise_buraya = 0

print(devam_ise_buraya)
```

**5.2 (10 Puan) SEED=öğrenci numaranız set ettikten sonra altıncı haftada yazdığımız gibi training batch’lerinden eğitim loss’ları, validation batch’lerinden validasyon loss değerlerini hesaplayan kodu aşağıdaki kod bloğuna yapıştırın ve çıkan figürü de alta ekleyin.**

```
kod_buraya = None
if kod_buraya:
    devam_ise_buraya = 0

print(devam_ise_buraya)
```

**5.3 (10 Puan) SEED=öğrenci numaranız set ettikten sonra altıncı haftada ödev olarak verdiğim gibi earlystopping’deki en iyi modeli kullanarak, Prensesi İyileştir test setinden accuracy, F1, precision ve recall değerlerini hesaplayan kodu yazın ve sonucu da aşağı yapıştırın. %80’den fazla başarı bekliyorum test setinden. Daha düşükse başarı oranınız, nerede hata yaptığınızı bulmaya çalışın. %90’dan fazla başarı almak mümkün (ben dedim).**

```
kod_buraya = None
if kod_buraya:
    devam_ise_buraya = 0

print(devam_ise_buraya)
```

Sonuçlar buraya

**5.4 (5 Puan) Tüm kodların CPU’da çalışması ne kadar sürüyor hesaplayın. Sonra to device yöntemini kullanarak modeli ve verileri GPU’ya atıp kodu bir de böyle çalıştırın ve ne kadar sürdüğünü hesaplayın. Süreleri aşağıdaki tabloya koyun. GPU için Google Colab ya da Kaggle’ı kullanabilirsiniz, iki ortam da her hafta saatlerce GPU hakkı veriyor.**

Tablo 1: Buraya bir açıklama yazın

Ortam	Süre (saniye)
CPU	kaç?
GPU	kaç?

**5.5 (3 Puan)** Modelin eğitim setine overfit etmesi için elinizden geldiği kadar kodu gereken şekilde değiştirin, validasyon loss'unun açıkça yükselmeye başladığı, training ve validation loss'ları içeren figürü aşağı koyun ve overfit için yaptığınız değişiklikleri aşağı yazın. Overfit, tam bir çanak gibi olmalı ve yükselmeli. Ona göre parametrelerle oynayın.

Cevaplar buraya

**5.6 (2 Puan)** Beşinci soruya ait tüm kodların ve cevapların olduğu jupyter notebook'un Github linkini aşağıdaki url'e koyun.

[www.benimgithublinkim.com](http://www.benimgithublinkim.com)

## 6 (Toplam 10 Puan)

Bir önceki sorudaki Prensisi İyileştir problemindeki yapay sinir ağınıza seçtiğiniz herhangi iki farklı regülerizasyon yöntemi ekleyin ve aşağıdaki soruları cevaplayın.

**6.1 (2 puan)** Kodlarda regülerizasyon eklediğiniz kısımları aşağı koyun:

```
kod_buraya = None
if kod_buraya:
    devam_ise_buraya = 0

print(devam_ise_buraya)
```

**6.2 (2 puan)** Test setinden yeni accuracy, F1, precision ve recall değerlerini hesaplayıp aşağı koyun:

Sonuçlar buraya.

**6.3 (5 puan)** Regülerizasyon yöntemi seçimlerinizin sebeplerini ve sonuçlara etkisini yorumlayın:

Yorumlar buraya.

**6.4 (1 puan)** Sonucun github linkini aşağıya koyun:

[www.benimgithublinkim2.com](http://www.benimgithublinkim2.com)