



İSTANBUL SABAHATTİN ZAIM
ÜNİVERSİTESİ

BIM 429 VERİ MADENCİLİĞİ

Türk Televizyon Dizilerinin Devamlılığının Tespiti

Grup No: [BG9]

Hazırlayanlar:
030120080 - Alper KARACA
030120044 - Selçuk YAVAŞ
030120040 - Ferhat TOSON

Teslim Tarihi: January 2, 2024

İçindekiler

1	Giriş	3
2	Literatür Özeti	3
3	Çerçevenin Yapısı, Bileşenler ve Mimari Tasarım	5
4	Yazılımın Kullanılması	8
5	Çerçevenin Algoritmasının Açıklanması	9
5.1	Rassal Orman	9
5.1.1	Çalışma Prensibi ve Adımları	10
5.2	Lojistik Regresyon	10
5.2.1	Çalışma Prensibi ve Adımları	11
5.3	Destek Vektör Makineleri	11
5.4	K En Yakın Komşu	12
5.4.1	Çalışma Prensibi ve Adımları	12
5.5	Karar Ağacı	13
5.5.1	Çalışma Prensibi ve Adımları	13
5.6	Adaptif Güçlendirme	14
5.6.1	Çalışma Prensibi ve Adımları	14
5.7	Gradyan Güçlendirme	15
5.7.1	Çalışma Prensibi ve Adımları	15
5.8	Aşırı Rassal Ağaçlar	16
5.8.1	Çalışma Prensibi ve Adımları	16
5.9	Stokastik Gradyan İnişi	16
5.9.1	Çalışma Prensibi ve Adımları	16
5.10	Ekstrem Gradyan Arttırma	17
5.10.1	Çalışma Prensibi ve Adımları	17
5.11	Hafif Gradyan Arttırma Makineleri	18
5.11.1	Çalışma Prensibi ve Adımları	18
5.12	CatBoost	19
5.12.1	Çalışma Prensibi ve Adımları	19
5.13	Multi Layer Perceptron	19
5.13.1	Çalışma Prensibi ve Adımları	20
5.14	Naive Bayes	20
5.14.1	Çalışma Prensibi ve Adımları	21

6 Çalışma Örnekleri	22
7 Sonuçlar ve Yorumlanması	23
7.1 1997 yılına ait model sonuçları	23
7.2 1998 yılına ait model sonuçları	24
7.3 1999 yılına ait model sonuçları	25
7.4 2000 yılına ait model sonuçları	26
7.5 2001 yılına ait model sonuçları	27
7.6 2002 yılına ait model sonuçları	28
7.7 2003 yılına ait model sonuçları	29
7.8 2004 yılına ait model sonuçları	30
7.9 2005 yılına ait model sonuçları	31
7.10 2006 yılına ait model sonuçları	32
7.11 2007 yılına ait model sonuçları	33
7.12 2008 yılına ait model sonuçları	34
7.13 2009 yılına ait model sonuçları	35
7.14 2010 yılına ait model sonuçları	36
7.15 2011 yılına ait model sonuçları	37
7.16 2012 yılına ait model sonuçları	38
7.17 2013 yılına ait model sonuçları	39
7.18 2014 yılına ait model sonuçları	40
7.19 2015 yılına ait model sonuçları	41
7.20 2016 yılına ait model sonuçları	42
7.21 2017 yılına ait model sonuçları	43
7.22 2018 yılına ait model sonuçları	44
7.23 2019 yılına ait model sonuçları	45
7.24 2020 yılına ait model sonuçları	46
7.25 2021 yılına ait model sonuçları	47
7.26 2022 yılına ait model sonuçları	48
7.27 2023 yılına ait model sonuçları	49
7.28 Sonuç	50
8 Gelecekte Yapılabilecekler	51
9 Kullanılan Kaynaklar	51

1 Giriş

Dizi sektörü, Türkiye'nin en önemli kültürel ve ekonomik sektörlerinden biridir. Her yıl onlarca yeni dizi yayına girmekte ve bu diziler milyonlarca izleyici tarafından takip edilmektedir. Ancak, yeni bir dizinin başarılı olup olamayacağı her zaman kesin olarak bilinmemektedir. Bazen çok iyi kadroya sahip bir dizi yayına girdikten sonra ilgi görmeyebilirken, bazen hiçraigbet görmeyeceği düşünülen bir dizi büyük bir başarı yakalayabilir.

Bu projede, veri madenciliği teknikleri kullanılarak yeni bir Türk dizisinin devamlılığının olup olamayacağına dair tahmin yapılması amaçlanmaktadır. Bu amaçla, geçmiş yıllarda yayınlanan Türk dizilerinin verileri incelenerek ve bu verilerden elde edilen sonuçlar kullanılarak bir tahmin modeli geliştirilecektir.

textbfProjenin ilk aşamasında, yeni bir dizinin devamlılığını etkileyebilecek faktörler belirlenecektir. Bu faktörler, daha önce yapılan araştırmalara dayanılarak belirlenecektir.

İkinci aşamada, belirlenen faktörlerin geçmiş yıllarda yayınlanan Türk dizilerine etkisi incelenecaktır. Bu amaçla, dizilerin yayın tarihleri, oyuncu kadrosu, senaryo, TV kanalı ve yayınlanma günü gibi veriler toplanacak ve bu veriler kullanılarak faktörlerin dizilerin devamlılığına etkisi analiz edilecektir.

Üçüncü aşamada, ikinci aşamada elde edilen sonuçlar kullanılarak bir tahmin modeli geliştirilecektir. Bu model, yeni bir dizinin devamlılığını etkileyebilecek faktörleri dikkate alarak, dizinin devamlılığının olup olamayacağına dair bir tahminde bulunacaktır.

Bu projenin sonucunda, yeni bir Türk dizisinin devamlılığını tahmin etmek için kullanılabilecek bir model geliştirilmesi hedeflenmektedir. Bu model, dizi sektöründe önemli bir ihtiyaç olan dizilerin başarısının önceden tahmin edilmesini mümkün kılacaktır.

2 Literatür Özeti

Deloitte, Türkiye'deki dizi sektörü hakkında kapsamlı bir analiz sunmaktadır. Altı büyük yayıncı kuruluşun (ATV, FOX TV, Kanal D, Show TV, Star TV ve TRT 1) yayın akışları incelenmiş ve 2010-2014 yılları arasındaki dört yayın sezonu boyunca yayınlanan dizilerin reytingleri, süreleri ve maliyetleri hakkında bilgi verilmiştir.

Türkiye'deki dizi sektörünün hızla büyüdüğü ve giderek daha fazla dizi üretiliği belirtilmektedir. 2010-2014 yılları arasında, onde gelen altı kanalda yayınlanan dizi sayısı ortalama olarak 70-80 arasında değişmiştir. Bu dönemde, en az bir dizisi onde gelen TV kanallarında yayınlanan 85 civarında yapım şirketi bulunmaktadır. Ancak, bu yapım şirketlerinin yaklaşık %50'si bu dönemde içerisinde sadece bir dizi yayınladıkları için oldukça parçalı bir pazar yapısı göze çarpmaktadır.

Türkiye'deki dizi sektörünün büyümeye başlaması, 2000'li yılların başında beri yaşanan endüstriyel, ekonomik ve sanatsal değişimlerin etkili olduğu belirtilmektedir. Bu değişimler, Türkiye'deki dizi sektörünün uluslararası alanda daha fazla tanınmasına ve ihracatının artmasına yol açmıştır. Türk dizilerinin Ortadoğu, Balkanlar, Kuzey Afrika ve Latin Amerika gibi birçok ülkede popüler olduğu ve Türk dizilerinin yurt dışında büyük bir izleyici kitlesi tarafından takip edildiği belirtilmektedir.

Türkiye'deki dizi sektörünün büyümeye rağmen, sektör paydaşlarının nitelikli veriye olan ihtiyacının kapsamlı bir doküman ile karşılaşması gereği belirtilmektedir. Sektör paydaşlarının ihtiyaç duyduğu nitelikli verileri sunarak, Türkiye'deki dizi sektörü hakkında kapsamlı bir analiz sunmaktadır.

Türkiye'deki dizi sektörünün maliyetleri hakkında da bilgi verilmektedir. Ortalama olarak bir dizi, reklamlar ve tekrarlar dahil yaklaşık 150-180 dakika sürmektedir. Batı standartları ile karşılaştırıldığında bu süre oldukça yüksektir. Yapım şirketleri, ünlü aktörlerin, senaristlerin ve yapımcıların tercih edilmesi nedeniyle maliyetlerini artırmaktadır. Buna karşılık, RTÜK düzenlemeleri bir dizi yayını içerisinde yer alacak reklam kuşaklarının sayısını ve sürelerini kısıtlamaktadır; bu da yayıcı kuruluşları dizi sürelerini uzatmaya teşvik etmektedir.

Dizilerin tekrarlarına ve özetlerine haftalık program akışlarında önemli bir pay verildiği ve bu tekrarların yayıcı kuruluşlar için önemli ek gelir kaynağı yarattığı belirtilmektedir. Ancak, bazı durumlardaki yayın adedi sınırlamalarına rağmen genellikle TV kanalları tekrarlar için yapımcılara herhangi bir ücret ödememektedirler.

Dizilerin izleyiciler tarafından en fazla tercih edilen program tipi olduğu ve ilk beş programın yaklaşık %50-55'inin dizilerden olduğu belirtilmektedir.

Ancak, reyting performansı düşük olan dizilerin genellikle sonlandırıldığı ve yayınlanan dizilerin yayınlandıkları günlerde ortalama olarak ilk 10'a giremedikleri belirtilmektedir.

Türkiye'deki dizi sektörünün rekabetçi yapısı incelenmektedir. İri ufaklı pek çok yapımcı kuruluşun sayıca az yayıncı kuruluşun kısıtlı yayın saatleri için rekabet etmelerinin yanı sıra sektördeki firmaların kurumsallaşma açısından nispeten daha emekleme döneminde olmaları, yapımcı kuruluşların sürdürülebilir bir yapıda faaliyet göstermelerinin önündeki en büyük engellerden birisidir. Reyting verileri incelendiğinde televizyon pazarının hakimiyeti paylaşılan bir yapıya sahip olduğu gözlemlenmektedir; yapım şirketleri az sayıdaki TV kanalına ürünlerini pazarlamaktadır. TV kanalları, reytingler ile reklam gelirleri arasındaki doğrudan ilişki nedeniyle, yüksek reytingli dizileri tercih etmektedirler.

Türkiye'deki dizi sektörünün yurt dışı pazarlarda rekabet gücünün artırılması için öneriler sunulmaktadır. Bu öneriler arasında, Türk dizilerinin yurt dışında daha fazla tanıtılması, yurt dışı pazarlarda daha fazla Türk dizisi yaylanması, Türk dizilerinin yurt dışında daha fazla ödül kazanması ve Türk dizilerinin yurt dışında daha fazla festivalde katılması yer almaktadır.

Sonuç olarak, Türkiye'deki dizi sektörünün hızla büyüdüğünü ve giderek daha fazla dizi üretildiğini göstermektedir. Ancak, artan maliyetler ve rekabet nedeniyle, düşük reytingli dizilerin genellikle sonlandırıldığı ve yayınlanan dizilerin yayınlandıkları günlerde ortalama olarak ilk 10'a giremedikleri belirtilmektedir. Türkiye'deki dizi sektörünün rekabetçi yapısı, maliyetleri, tekrarları ve özetleri, yurt dışı pazarlarda rekabet gücü ve diğer konular hakkında kapsamlı bir analiz sunulmaktadır.

3 Çerçevenin Yapısı, Bileşenler ve Mimari Tasarım

Projenin başlangıcında, Türk televizyon dizileriyle ilgili verileri elde etmek için web kazıma (web scraping) yöntemi kullanıldı ve bu veriler Wikipedia'dan çekildi. Ardından, elde edilen dizi bilgileriyle ilgili özel sayfalara gidilerek belirli bilgiler toplandı. Bu bilgiler arasında dizi adı, türü, senarist, yönetmen, başrol oyuncuları, bölüm sayısı gibi detaylar bulunmaktaydı. Ayrıca yapımcı, çekim mekanı, gösterim süresi, yayın tarihi gibi veriler de elde edildi. Bu bil-

giler arasında sosyal medya adresleri, yapımcı şirketler ve hangi kanal ya da platformda yayınlandığı gibi detaylar da yer aldı.

İkinci aşamada, elde edilen veri setinde eksik olan bilgilerin tamamlanması için Google Sheets gibi bir çevrimiçi tablo platformuna yüklandı. Ekip üyeleri eksik olan verileri el ile girdi. Ardından, Python kullanılarak satırlarda bulunan parantez içindeki ifadeler temizlendi ve gereksiz noktalama işaretleri düzeltildi.

Üçüncü aşamada, veri seti üzerinde keşifsel veri analizi yapıldı. Bu analizde, veri setinde en sık geçen değerler belirlendi. Veri analizi sonucunda, elde edilen verilerle ilgili özelliklerin belirlenmesine yönelik fikirler oluşturuldu.

Dördüncü aşamada ise özellik çıkarma işlemi gerçekleştirildi. Bu süreçte, veri setinden çeşitli özellikler çıkarıldı ve bu özelliklerin incelenerek hangi bilgilerin daha anlamlı olduğu belirlendi. Veri setinden aşağıdaki özellikler çıkarıldı.

Beşinci aşamada model eğitimi sürecinde adımlar şu şekilde gerçekleşti:

1. **SelectKBest Yöntemiyle Optimum Özellik Seçimi:** İlk olarak, veri setindeki en önemli özelliklerin belirlenmesi için SelectKBest yöntemi kullanıldı. Bu yöntem, ANOVA F-değeri metoduyla özelliklerin hedef değişkenle ilişkisini değerlendirir. Eğer özellikler sayısal ise, her bir özelliğin hedef değişkenle ilişkisini ifade eden ANOVA F-değerleri hesaplandı. Bu değerler, özelliğin hedef değişkeni ne kadar iyi açıkladığını ölçer.
2. **Optimum Hiperparametrelerin Belirlenmesi:** Optimum özellik sayısını belirledikten sonra, modelin hiperparametrelerini belirlemek için Optuna gibi bir araç kullanıldı. Optuna, modellerin performansını artırmak için hiperparametre araması yapar. Bayesian Optimization yöntemini kullanarak, model için en iyi parametre setini belirlemek için rastgele parametre kombinasyonları denenir.
3. **Modelin Oluşturulması ve Kalibrasyonu:** Model oluşturulduktan sonra, modelin çıktılarının daha doğru olması için kalibrasyon işlemi yapıldı. Bu aşamada, "sigmoid" ve "isotonic" gibi kalibrasyon yöntemleri

kullanıldı. Sigmoid yöntemi, modelin olasılık tahminlerini sigmoid fonksiyonuyla düzenlerken; isotonic yöntemi ise monoton artış gösteren bir fonksiyonla modelin çıktılarını düzenler.

4. **HuggingFace Platformu Üzerinden Canlıya Alma:** Son olarak, kalibre edilmiş modeller HuggingFace platformu üzerinde kullanıma hazır hale getirildi. Bu platform, NLP (Doğal Dil İşleme) ve makine öğrenimi modellerini paylaşmak, kullanmak ve dağıtmak için kullanılır.

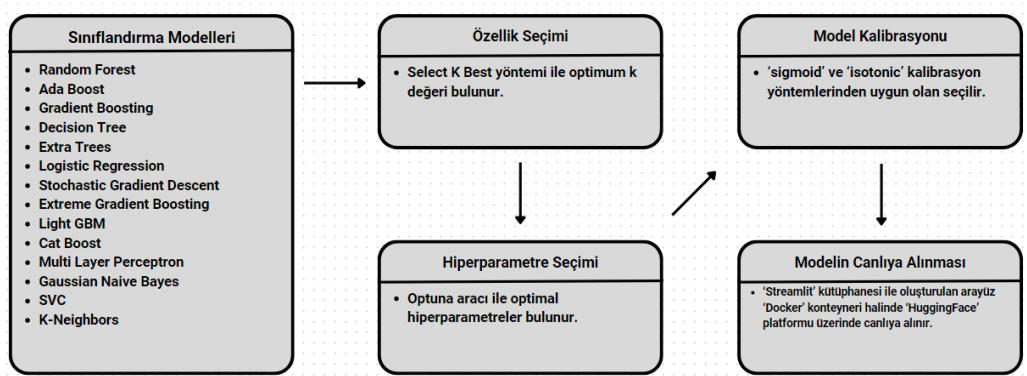


Figure 1: Yazılım çerçevesi.

Table 1: Veri setinden çıkarılan özellikler.

Özellik Adı	Özellik Açıklaması
Tarih	Dizinin çekildiği yıl.
Dizi adı uzunluğu	Dizi adının uzunluğu.
Dizi adındaki kelime sayısı	Dizi adındaki toplam kelime sayısı.
Dizi adında bağlaç	Dizi adındaki bağlaç sayısı.
Dizi adında yer ismi	Dizi adındaki yer ismi bulunuyor mu ?
Dizi adında özel isim	Dizi adındaki özel isim bulunuyor mu ?
Dizinin türü	Dizi Aile, Aksiyon, Aşk, Bilim Kurgu, Dram, Gençlik, Gerilim, Komedi, Polisiye, Romantik ve Tarihi türlerine sahip mi ?
Dizinin yaymlandığı kanal	Dizi TRT 1, Kanal D, atv, Star TV, Show TV, FOX, Samanyolu TV, TV8 kanallarında yayılmış mı ?
Çekim yeri	Dizi Türkiye'de mi Yurtdışında mı çekilmiş ?
Uyarlama	Dizi bir uyarlama mı ?
Başrol sayısı	Dizideki başrol sayısı.
Kanal fiyatı	Dizinin yaymlandığı kanal ücretli mi ücretsiz mi ?
Yayınlandığı gün	Dizinin yaymlandığı gün.
Yaz dizisi	Temmuz veya Ağustos ayında mı yayılmış ?
Gösterim süresi	Dizinin bir bölümünün ortalaması süresi.
Ödüllü	Dizi Altın Kelebek ödülü almış mı ?
Senaristlerin yazdığı ortalama dizi sayısı	Senaristlerin o diziden önce yazdığı ortalama dizi sayısı.
Ödüllü senarist sayısı	Dizideki ödülü senarist sayısı.
Senaristlerin aldığı ortalama ödül sayısı	Senaristlerin aldığı ortalama ödül sayısı.
Başrollerin oynadığı ortalama dizi sayısı	Dizideki başrollerin o diziden önce oynadığı ortalama dizi sayısı.
Ödüllü başrol sayısı	Dizideki ödülü başrol sayısı.
Başrollerin aldığı ortalama ödül sayısı	Başrollerin aldığı ortalama ödül sayısı.
Yönetmenlerin yaptığı ortalama dizi sayısı	Yönetmenlerin o diziden önce yaptığı ortalama dizi sayısı.
Ödüllü yönetmen sayısı	Dizideki ödülü yönetmen sayısı.
Yönetmenlerin aldığı ortalama ödül sayısı	Yönetmenlerin aldığı ortalama ödül sayısı.
Görüntü yönetmenlerinin ortalama yaptığı dizi sayısı	Görüntü yönetmenlerinin o diziden önce yaptığı dizi sayısı.
Bestecilerin bestelediği ortalama dizi sayısı	Bestecilerin o diziden önce bestelediği ortalama dizi sayısı.
Yapım şirketlerinin yaptığı ortalama dizi sayısı	Yapım şirketlerinin o diziden önce yaptıkları ortalama dizi sayısı.
Sosyal medya hesapları	İziler YouTube, Facebook, Twitter ve Instagram hesabına sahip mi ?

4 Yazılımın Kullanılması

Yazılım, **Google Colab** ortamında geliştirilmiştir. Yazılımı çalıştırmak için aşağıdaki kütüphanelerin kurulması gereklidir. Kurulum klasörü içinde bulunan 'requirements.txt' dosyasını kullanarak `pip install -r requirements.txt` komutu çalıştırılarak gerekli kütüphanelerin kurulması sağlanır.

```
imbalanced_learn==0.11.0
ipython==8.14.0
lightgbm==4.2.0
matplotlib==3.8.1
mlxtend==0.23.0
optuna==3.5.0
pandas==2.1.4
scikit_learn==1.3.2
```

```
scikit_learn_intelex==2024.0.1
scikit_plot==0.3.7
seaborn==0.13.0
tensorflow==2.15.0.post1
tensorflow_cpu==2.15.0.post1
numpy==1.23.5
streamlit==1.29.0
lightgbm==4.2.0
xgboost==2.0.3
catboost==1.2.2
folium==0.15.1
spacy==3.7.2
```

5 Çerçevenin Algoritmasının Açıklanması

Kullanılan algoritmanın detaylı açıklaması, nasıl çalıştığı ve hangi problemleri çözdüğü...

5.1 Rassal Orman

Rassal orman (Random Forest), bir ensemble (bir araya getirme) algoritmasıdır. Hiper parametre seçimi yapmadan da iyi sonuçlar vermesinden dolayı popülerdir. Random Forest genellikle sınıflandırma ve regresyon problemlerini çözmek için kullanılır. Birden fazla alt ağaçlar oluşturarak overfitting'in önüne geçer. Her bir dal için Gini değişkeni hesaplanır. Alt daldaki Gini değişkeni, bir üst daldaki gini değişkeninden az ise o dal başarılıdır demektir. Yani en üstten en alta doğru gini değişkeni azalır. Hem sınıflandırma hem de regresyon problemlerini çözmek için kullanılır. Gözlemler bootstrap yöntemi ile, değişkenler Random subspace yöntemini kullanır. Random subspace teknigi, P adet değişkeni olan bir veri setinde, P 'den daha az sayıda bir rastgele değişken seçerek her bir ağaç için, dallanmaların bu değişkenler üzerinden yapılması sağlanır. Böylece rastgelelik elde edilmiş olur. Bootstrap, benzer örnek oluşturmak için tek bir veri setini yeniden örnekleyen istatistiksel bir prosedürdür. Rastgele olacak şekilde asıl veri setinin $2/3$ 'ü seçilir. Bootstrap veri setinde olmayan $1/3$ 'luk kısıma "Out-of-Bag Dataset" denir. Random Forest, ensemble teknigini kullanır. Ensemble, bir grup zayıf öğrenicinin güçlü bir öğrenici oluşturmak için bir araya gelmesidir. Ensemble,

temelinde 2 yaklaşım vardır:

- **Bagging:** Birçok zayıf öğrenme modelini bağımsız olarak eğitip birleştirir. Ana fikir, veri kümesinin rastgele alt küme (bootstrap) kullanılarak farklı modellerin eğitilmesi ve bu modellerin sonuçlarının birleştirilmesidir. Temel amacı varyansı azaltmaktadır.
- **Boosting:** Zayıf öğrenme modellerini ardışık olarak eğtiir ve her bir model, önceki modelin hatalarına odaklanır. Yani, hatalı sınıflandırılan örnekler üzerinde daha fazla vurgu yaparak performansı artırır. Boosting'in temel amacı bias'ı azalmaktır.

5.1.1 Çalışma Prensibi ve Adımları

1. **Veri Toplama:** Özellikler ve hedef değişken belirlenir.
2. **Ağaç Oluşturma:** Birden fazla karar ağaçları oluşturur. Her ağaç, rastgele seçilen alt örneklem verileri üzerinde kurulur. Aynı zamanda her düğümde rastgele seçilen bir alt küme özellik kullanılır. Bu, her ağaçın farklı olduğu ve çeşitliliği artırdığı anlamına gelir.
3. **Ağaçlar Arası Bağımsızlık:** Her ağaç bağımsız olarak kurulur ve tahmin yapar. Bu, tek bir ağaç hatalı tahminlerde bulunursa, diğer ağaçlar da bu hatayı dengeleyebilir.
4. **Tahmin Yapma:** Her ağaçın verilen bir örneklemeye için tahmin yapar. Sınıflandırma problemlerinde en çok oy alan sınıfı seçer. Regresyon problemlerinde tüm ağaçların tahminin ortalaması alınır.

5.2 Lojistik Regresyon

İsmi matematikteki, lojistik (sigmoid) fonksiyondan alır. Bir bağımlı değişkenin (genellikle kategorik) olasılığını tahmin etmek veya sınıflandırmak için kullanılan bir istatistiksel modeldir. Temel amacı, bir girdi örneğini belirli bir sınıfa ait olma olasılığını tahmin etmektir. Lineer regresyon ile arasındaki fark, Lineer regresyon optimum çizgiyi çizmek için "En Küçük Kareler Yöntemi (Least Squares)", lojistik regresyon "Maksimum Olabilirlik (Maksimum Likelihood)" kullanır. Sigmoid Fonksiyonu, verileri 0 ile 1 arasında sıkıştırmak için kullanılan fonksiyondur.

5.2.1 Çalışma Prensibi ve Adımları

$$P(y=1|x) = \frac{1}{1+e^{-(\beta_0+\beta_1x_1+\beta_2x_2+\dots+\beta_nx_n)}} \text{ Bu denklemde:}$$

- $P(Y=1)$, bağımlı değişkenin 1 (başarı, pozitif sınıf) olma olasılığını temsil eder.
- e, Euler sayısı (yaklaşık olarak 2.71828) olarak bilinir.
- z, lojit (log-odds) değeridir ve şu şekilde hesaplanır: $z = b_0 + b_1x_1 + b_2x_2 + \dots + b_n * x_n$
 - b_0 , kesme terimidir.
 - b_1, b_2, \dots, b_n , bağımsız değişkenlerin katsayılarıdır.
 - x_1, x_2, \dots, x_n , bağımsız değişkenlerin değerleridir.

1. **Örnekleri Sınıflandırma:** Her bir örneği sınıflandırmak için bağımsız değişkenlerin ağırlıklı toplamını ve lojistik fonksiyonu kullanır. Bu sonuç, 0 ile 1 arasında bir olasılık değeridir.
2. **Eşik Değer Karşılaştırması:** Elde edilen olasılık değeri, belirli bir eşik değer (0.5) üzerindeyse örnek "1" sınıfına atanır, değilse "0" sınıfına atanır.
3. **Model Eğitimi:** Model, eğitim verileri üzerinde eğitilir.

5.3 Destek Vektör Makineleri

Temel amacı, verileri bir hiperdüzlem üzerinde sınıflandırmak veya regresyon analizi yapmaktadır. SVM, bu hiperdüzlemi oluştururken, sınıflar arasındaki en büyük marjı (boşluk) bulmaya çalışır. Marj, hiperdüzleme en yakın veri noktalarından uzaklık olarak tanımlanır ve bu noktalara "destek vektörleri" denir. Hiperdüzlem, bu destek vektörleri arasında yer alır ve iki sınıf arasındaki boşluğu maksimize eder. SVM'nin çalışma prensibi matematiksel olarak iki sınıf için aşağıdaki şekilde ifade edilir:

Veri noktası: (x, y) Hiperdüzlem: $w * x + b = 0$

- w : Hiperdüzlemi belirleyen normal vektör (ağırlıklar).
- x : Veri noktasının özellik vektörü.

- b: Bias terimi.

Sınıf tahminlemesi yapmak için, veri noktasını hiperdüzlem formülüne yerleştiririz:

- Eğer $w * x + b > 0$ ise, veri noktası sınıf +1'e aittir.
- Eğer $w * x + b < 0$ ise, veri noktası sınıf -1'e aittir.

5.4 K En Yakın Komşu

KNN, veri noktalarının komşularına dayalı olarak sınıflandırılmasını veya tahmin edilmesini sağlayan bir algoritmadır. Temel fikir, bir veri noktasının sınıfını veya değerini belirlemek için, bu noktanın en yakın komşularının sınıfını veya değerlerini kullanmaktadır.

5.4.1 Çalışma Prensibi ve Adımları

Tahmin yapılacak yeni veri noktası ile eğitim verilerindeki diğer noktalar arasındaki uzaklığı hesaplanır. Genellikle kullanılan uzaklık ölçütleri;

- **Euclidean (Öklidyen):** İki nokta arasındaki doğru mesafeyi ölçer.
- **Manhattan:** İki nokta arasındaki yolların toplam uzunluğunu ölçer.
- **Chebyshev:** Vektörler arasındaki mutlak farkın maksimumunu alır. Daha sonra en yakın "k" komşusu seçilir. k, bir hiperparametredir ve kullanıcı tarafından belirlenmelidir.
- **Sınıflandırma Problemi:** Eğer KNN sınıflandırma amaçlı kullanılıyorsa, en yakın k komşunun sınıfları incelenir ve yeni veri noktasının sınıfı, bu k komşunun sınıfının çoğunluğu (modu) olarak tahmin edilir. Örneğin, eğer 3 en yakın komşu sınıfları "A", "A", ve "B" ise, yeni veri noktasının tahmini sınıfı "A" olur.
- **Regresyon Problemi:** Eğer KNN regresyon amaçlı kullanılıyorsa, en yakın k komşunun hedef değerleri kullanılarak yeni veri noktasının hedef değeri tahmin edilir. Genellikle bu değerlerin ortalaması veya ağırlıklı ortalaması kullanılır.

5.5 Karar Ağacı

Karar ağacı, bir veri setini, bir dizi karar uygulayarak daha küçük kümelere bölmek için kullanılan bir yapıdır. Karar ağacında ilk bölünmenin başladığı yere kök, dalların uzaması ile gelişen kısımlara düğüm, alt ugulara ise yaprak denir. Karar ağaçları, en iyi bölünmeyi seçmek için farklı kriterler kullanır. Daha uzun ağaçlar yerine daha kısa ağaçlar tercih edilir. En yaygın olarak kullanılan kriterler. Bilgi Kazancı (Information Gain): Bir düğümün bölünmesinin ne kadar bilgi kazandıracağını ölçer. Bilgi kazancı, her alt düğümün belirli bir özellikle ne kadar homojen olduğunu gösteren bir metriktir. Bilgi kazancı, bir özellikle yapılan bölünmenin önceki duruma göre ne kadar daha az belirsizlik (bilgi eksikliği) getirdiğini ölçer. Bu, ağacın daha homojen alt gruplara bölünmesine ve iyi tahminler yapmasına yardımcı olur. Sırasıyla şu adımlar izlenir:

1. **Entropi (Entropy):** Bir veri kümelerinin ne kadar homojen veya heterojen olduğunu ölçen bir kavramdır. Daha yüksek entropi, daha fazla belirsizlik veya bilgi eksiliği anlamına gelir. Entropi şu formülle hesaplanır:
$$H(X) = -\sum_{i=1}^n P(x_i) \cdot \log_2 P(x_i)$$
, burada p_1, p_2, \dots, p_k , S' deki her sınıfın olasılıklarıdır.
2. **Dallanma (Split):** Bir özellikle seçilir ve veri kümesi bu özelliğe göre alt gruplara bölünür. Her alt grup için entropi hesaplanır ve bu alt grupların ağırlıklı ortalaması (weighted average) alınır.
3. **Bilgi Kazancı (Information Gain):** Entropinin önceki durum ile yeni durum arasındaki farkını ölçer. Information Gain yüksekse, o özellik daha fazla bilgi kazandırır ve ağacın bölünmesinde daha önemli bir rol oynar.

5.5.1 Çalışma Prensibi ve Adımları

1. **Veri Toplama:** Özellikler ve hedef değişken belirlenir.
2. **Ağacı Oluşturma:** Ağacın kök düğümü seçilir ve veriler bu düğümde belirli bir özelliğe göre bölünürl. Ardından, her alt düğüm için aynı adım tekrarlanır.

3. **Düğüm Bölünmesi:** Düğümler, en iyi bölünme kriterine göre alt düğümlere bölünür. Bölünme kriterleri genellikle bilgi kazancı, gini indeksi veya ortalama hata gibi metrikler kullanılarak belirlenir.
4. **Ağaç Düzenleme:** Ağaç gereğinden fazla dallanma yapabilir. Bu nedenle gereksiz dallar kaldırılmalıdır. Budama (pruning), arar ağacında tahmine yeterince katkı yapmayan dallarda tahmin edici değişkenlerin modelden çıkarılması işlemidir. Post ve Pre olarak ikiye ayrılır. Prepruning, tahmin edici değişkenleri teker teker ele alarak modelin tahmin gücü için hangisinin etkili olacağı kararlaştırılarak adım adım dallanmaların iletirtilmesidir. Postpruning, tamamlanmış bir karar ağacından modele katkı yapmayan dalların tespit edilip modelden çıkarılmasıdır.
5. **Tahmin:** Yeni veriler için tahminler yapılır.

5.6 Adaptif Güçlendirme

AdaBoost, düşük başarılı modeller olarak adlandırılan zayıf öğrenicileri bir araya getirerek daha güçlü bir öğrenici oluşturan bir toplu öğrenme algoritmasıdır. Temel amacı, zayıf modellerin sınırlı başarılarını birleştirerek genelde daha güçlü ve genelleyici bir model elde etmektir. Bu algoritma, tek başına başarılı olamayabilecek basit modellerin gücünü birleştirerek genellemeye yeteneğini artırır ve aşırı öğrenmeyi azaltır.

5.6.1 Çalışma Prensibi ve Adımları

1. **Başlangıç Modeli Seçimi:** İlk olarak, veri seti üzerinde bir zayıf öğrenici (örneğin, bir karar ağıacı) seçilir ve eğitilir.
2. **Hata Hesaplama:** Modelin üzerinde çalıştığı örnekler arasındaki hatalar hesaplanır, yanlış sınıflandırılan örneklerin ağırlığı arttırılır.
3. **Ağırlıklı Veri Seti Oluşturma:** Hataların ağırlıklarına göre ayarlandığı yeni bir ağırlıklı veri seti oluşturulur. Bu, daha fazla vurgu gerektiren hatalı örnekleri içerir.
4. **Yeni Modelin Eğitimi:** Yeni ağırlıklı veri seti üzerinde bir sonraki zayıf öğrenici eğitilir. Bu öğrenici, önceki hatalara daha fazla odaklanarak modelin zayıflıklarını düzeltmeye çalışır.

5. **Ağırlık Güncelleme:** Her öğrenici eklenikten sonra, modelin performansına göre her öğrencinin ağırlığı belirlenir. Daha başarılı modeller daha fazla ağırlığa sahip olur.
6. **Sonuç Oluşturma:** Tüm zayıf öğrencilerin bir araya getirilmesiyle güçlü bir öğrencisi elde edilir. Her öğrencinin katkısının ağırlığına göre final tahminler yapılır.

5.7 Gradyan Güçlendirme

Gradient Boosting, zayıf öğrencilere bir araya getirerek güçlü bir öğrencisi oluşturan bir makine öğrenimi yöntemidir. Adım adım hatayı düzelterek modeli iyileştirir. Bu teknik, özellikle sayıları tahmin etme veya nesneleri sınıflandırma gibi görevlerde kullanılmıştır. Ancak, model çok karmaşıksa, aşırı uyuma dikkat etmek önemlidir.

5.7.1 Çalışma Prensibi ve Adımları

1. **Başlangıç Modeli:** İlk olarak, bir başlangıç modeli (genellikle bir karar ağaçları) seçilir ve eğitilir.
2. **Hata Hesaplama:** Başlangıç modelinin tahminlerindeki hatalar hesaplanır.
3. **Hata Temelinde Yeni Model:** Hatalar üzerine odaklanarak yeni bir model daha eklenir. Bu model, önceki modelin hatalarını düzeltmeye çalışır.
4. **Ağırlıklı Hata:** Her öğrencinin hatalar düzeltildikçe, her veri örneğinin hata oranlarına göre ağırlıkları güncellenir. Hatalı tahminlere daha fazla ağırlık verilir.
5. **Toplam Tahmin Oluşturma:** Tüm öğrencilerin tahminleri bir araya getirilerek toplam tahmin oluşturulur. Bu, her öğrencinin katkısının önceki hataları düzeltmeye odaklandığı bir şekilde gerçekleşir.
6. **İteratif Süreç:** Bu süreç belirli bir iterasyon sayısına veya belirli bir hata eşigine ulaşana kadar devam eder. Her iterasyonda, yeni eklenen öğrencisi, öncekilerin hatalarını düzeltmeye çalışarak modeli iyileştirir.

5.8 Aşırı Rassal Ağaçlar

Extra Trees (Extremely Randomized Trees), bir makine öğrenimi algoritmasıdır ve özellikle sınıflandırma ve regresyon problemleri için kullanılır. Extra Trees, Random Forest algoritmasına benzer bir yaklaşımı benimser ancak bazı farklar vardır. Extra Trees, karar ağaçları oluştururken daha fazla rastgelelik ekleyerek, çeşitli modeller oluşturur ve bu sayede genelleme yeteneğini artırır. Ancak, bu ekstra rastgelelik nedeniyle her bir ağaçın ayrı ayrı ayarlanması veya yüksek varyanslı veri setlerinde başarılı olabilmesi mümkün değildir.

5.8.1 Çalışma Prensibi ve Adımları

- Alt Örneklemler ve Özellik Seçimi:** Her bir ağaç için, veri setinden rastgele bir alt örneklem ve rastgele bir özellik alt kümesi seçilir.
- Karar Ağacı Oluşturma:** Seçilen alt örneklem ve özellik alt kümesi üzerinde bir karar ağacı oluşturulur. Bu ağaç, geleneksel karar ağaçlarından farklı olarak, düğümlerdeki bölgemeleri rastgele seçer.
- Ağaçların Birleştirilmesi:** Belirlen sayıda karar ağacı oluşturulduktan sonra, bu ağaçlar bir araya getirilir ve toplu bir model oluşturulur.
- Oylama veya Ortalama:** Sınıflandırma problemlerinde genellikle çokunluk oylaması, regresyon problemlerinde ise ağaçların tahminlerinin ortalaması kullanılarak final tahmin yapılır.

5.9 Stokastik Gradyan İnişi

SGD, bir optimizasyon algoritmasıdır ve özellikle büyük veri setleri üzerinde eğitim yaparken etkilidir. Genellikle makine öğrenimi modellerini eğitmek için kullanılır. SGD, özellikle büyük veri setleri üzerinde etkilidir, çünkü her eğitim örnekleri üzerinde ağırlık güncellemesi yaparak hızlı bir şekilde öğrenmeyi sağlar. Ayrıca, modellerin online eğitimini ve gerçek zamanlı uygulamalarda kullanımı kolaylaştırır.

5.9.1 Çalışma Prensibi ve Adımları

- Başlangıç Parametreleri:** Modelin başlangıç parametreleri (ağırlıklar ve biaslar) rastgele veya belirli bir stratejiye dayanarak belirlenir.

2. **Veri Setinin Karıştırılması:** Veri seti, modelin eğitimini daha iyi ve daha hızlı hale getirmek için karıştırılır. Bu, her bir eğitim döngüsünde farklı örneklerin model tarafından işlenmesini sağlar.
3. **Stokastik Gradyan Hesaplaması:** Her bir örnek üzerinde modelin gradyanı (eğiminin) hesaplanır. Bu, maliyet fonksiyonunun modelin mevcut parametreleriyle ne kadar değiştirilmesi gerektiğini gösterir.
4. **Parametre Güncellemesi:** Hesaplanan gradyan kullanılarak modelin parametreleri güncellenir. SGD, her örnek üzerinde tek bir güncelleme yapar, bu nedenle "stokastik" olarak adlandırılır.
5. **Maliyet Fonksiyonunun Hesaplanması:** Güncellenmiş parametrelere maliyet fonksiyonu hesaplanır. Maliyet, modelin tahminlerinin gerçek değerlerden ne kadar sapma gösterdiğini ölçer.
6. **Konverjans Kontrolü:** Belirli bir konverjans kriterine veya belirli bir epoch (eğitim döngüsü) sayısına ulaşınca kadar bu adımlar tekrarlanır.

5.10 Ekstrem Gradyan Arttırma

XGBoost, ağaç tabanlı modelleme yaklaşımı kullanarak özellikle sınıflandırma ve regresyon problemleri için kullanılan bir makine öğrenimi algoritmasıdır. Gradient Boosting yöntemini temel alan XGBoost, önceki öğrenicilerin hatalarını düzeltmeye odaklanarak, bir dizi zayıf öğreniciyi birleştirir ve güçlü bir öğrenici oluşturur. XGBoost, performansı ve hızı artırmak için bir dizi optimize edilmiş algoritma ve özellik içerir. Ayrıca, özellik seçimi, aşırı öğrenme kontrolü, eksik veri yönetimi ve hızlı eğitim gibi avantajlara sahiptir.

5.10.1 Çalışma Prensibi ve Adımları

1. **Başlangıç Modeli:** İlk öğrenici, genellikle küçük bir karar ağaçıdır. Bu ağaç, veri setini sınıflandırmaya veya bir çıktı değeri tahmin etmeye çalışır.
2. **Hataların Hesaplanması:** İlk öğrenici tarafından yapılan tahminlerin hataları hesaplanır. Hatalar, gerçek değerler ile önceki tahminler arasındaki farkları ifade eder.

3. **İkinci Öğrenici:** İlk öğrencinin hatalarını düzeltmeye odaklanarak ikinci bir öğrenci eklenir. İkinci öğrenci, hataları minimize etmeye çalışır.
4. **Ağırlıklı Toplama:** İlk ve ikinci öğrencilerin tahminleri, hatalarına göre ağırlıklı bir şekilde toplanır. Bu, her bir öğrencinin katkısının belirlenmesine yardımcı olur.
5. **Devam Eden İterasyonlar:** Hata düzeltme ve ağırlıklı toplama işlemleri belirli bir iterasyon (epoch) sayısına veya hata eşliğine ulaşılınca kadar devam eder. Her bir iterasyonda, yeni öğrenciler eklenir ve model geliştirilir.

5.11 Hafif Gradyan Arttırma Makineleri

LightGBM, özellikle büyük veri setleri, yüksek boyutlu özellik uzayları ve hızlı model eğitimi gerektiren durumlar için uygundur. LightGBM, histogram tabanlı öğrenme ve düzenlendirilmiş öğrenme algoritmalarını kullanarak efektif bir şekilde çalışır. Modelin performansını artırır ve overfitting'i kontrol altında tutar. LightGBM, sınıflandırma, regresyon ve sıralama gibi çeşitli problemleri çözmek için kullanılır.

5.11.1 Çalışma Prensibi ve Adımları

1. **Histogram Tabanlı Öğrenme:** LightGBM, veri setini önceden oluşturulmuş histogramlara dönüştürerek çalışır. Bu, özellik değerlerini bölme noktalarına dönüştürmek ve veriyi daha hızlı işlemek için kullanılır.
2. **Yerleşik Paralel İşleme:** LightGBM, büyük veri setlerini paralel olarak işleyebilen dağıtılmış bir yapıya sahiptir. Bu, eğitim süresini hızlandırır ve ölçeklenebilirliği artırır.
3. **Düzenlendirilmiş Öğrenme:** Düzenlendirilmiş öğrenme algoritmaları, LightGBM'nin aşırı uyumu kontrol etmesine yardımcı olur. Bu, ağaçların karmaşıklığını sınırlar ve genelleme yeteneğini artırır.
4. **Ağaç Büyüütme ve Eğitim:** LightGBM, öğrenme işlemi sırasında ağaçları aşama aşama büyütür. Her ağaç, önceki ağaçların hatalarını düzeltmeye odaklanarak eklenir.

5. **Gradient Boosting:** LightGBM, önceki öğrenicilerin hatalarını azaltmaya çalışarak ağaçları birleştirir. Bu, gradient boosting yöntemini temel alır.

5.12 CatBoost

CatBoost, özellikle kategorik değişkenlerle çalıştığımızda, hızlı model eğitimi ve güçlü performans sağlamak için tasarlanmıştır. CatBoost, sınıflandırma, regresyon ve sıralama gibi çeşitli problemleri çözmek için kullanılır.

5.12.1 Çalışma Prensibi ve Adımları

1. **Kategori Özelliklerinin Otomatik İşlenmesi:** CatBoost, kategorik özellikleri doğrudan kullanabilir ve bu özelliklerin içerdiği bilgileri otomatik olarak işleyebilir. Bu, veri mühendisliği aşamasında kategori özelliklerinin dönüştürülmesi veya kodlanması gerekliliğine anlamlı gelir.
2. **Başlangıç Ağacı Modeli:** CatBoost, genellikle bir başlangıç ağıacı modeli ile başlar. Bu ağaç, basit bir modelle başlayarak gradient boosting algoritmasıyla birleştirilir.
3. **Ağaç Büyütme ve Düzeltme:** Her bir öğrenici, önceki öğrenicilerin hatalarını düzeltmeye odaklanarak eklenir. Bu, gradient boosting'in temel prensibidir.
4. **Simetrik Yapı:** CatBoost, önceki ağaçlardan gelen hataların düzeltilmesi için simetrik ağaç yapısı kullanır.
5. **Dengeli Örnek Ağırlıkları:** CatBoost, sınıflandırma problemlerinde dengesiz veri setlerini ele almak için özel bir ağırlıklandırma stratejisi kullanır.
6. **Kategorik Değişkenlere Özel Optimizasyon:** CatBoost, kategorik değişkenlerin kullanımını optimize etmek ve ağırlıklarını otomatik olarak ayarlamak için özel bir içsel optimizasyon stratejisi kullanır.

5.13 Multi Layer Perceptron

MLP, yapay sinir ağlarının bir türüdür ve özellikle derin öğrenme alanında kullanılan bir modeldir. MLP, en azından bir giriş katmanı, bir veya daha

fazla gizli katman ve bir çıkış katmanından oluşur. Her katman, birbirine bağlı nöronlardan oluşur ve bu nöronlar, ağırlıklar ve aktivasyon fonksiyonları ile ilişkilidir.

5.13.1 Çalışma Prensibi ve Adımları

1. **Giriş Katmanı:** İlk katman, veri setindeki özelliklerini temsil eder. Her bir özellik, giriş katmanındaki bir nörona bağlanır.
2. **Gizli Katmanlar:** Bir veya daha fazla gizli katman, öğrenme ve özellik öğrenme süreçlerini gerçekleştirir. Her gizli katmanın nöronları, önceki katmandaki nöronlardan gelen ağırlıklı toplamları ve bir aktivasyon fonksiyonu tarafından işlenmiş çıktıları alır.
3. **Ağırlıkların ve Bias'ın Güncellenmesi:** Eğitim sürecinde, ağırlıklar ve bias'lar, modelin hata oranını minimize etmek amacıyla geri yayılım (backpropagation) algoritması ile güncellenir. Bu süreç, optimizasyon algoritmaları (örneğin, gradyan iniş) kullanılarak gerçekleştirilir.
4. **Çıkış Katmanı:** Son katman, istenen çıktıyı üretir. Probleme bağlı olarak, bu katman bir tek nöron (regresyon problemleri) veya birden fazla nöron (sınıflandırma problemleri) içerebilir. Sınıflandırma problemlerinde genellikle softmax aktivasyon fonksiyonu kullanılır.

5.14 Naive Bayes

Naive Bayes algoritması, olasılık teorisine dayalı bir makine öğrenimi ve istatistiksel sınıflandırma algoritmasıdır. Temel olarak, bir nesnenin belirli bir sınıfı ait olma olasılığını tahmin etmek için kullanılır. Bu sınıflandırma algoritması, Bayes Teoremi'ne dayanır ve "naif" (ingenuous) olarak adlandırılır, çünkü sınıf tahminindeki özellikler arasındaki bağımsızlık varsayımlı yapar, yani her özelliğin sınıfı etkileme olasılığı birbirinden bağımsızdır. 3 türü vardır;

- **Gaussian Naive Bayes:** Özellikler sürekli değer (continuous value) ise bu değerlerin bir gauss dağılımı veya diğer bir değişle normal dağılımdan örneklenliğini varsayar.
- **Multinomial Naive Bayes:** Çok sınıfı kategorileri sınıflandırmak için kullanılır.

- **Bernoulli Naive Bayes:** Multinomial Naive Bayes'e benzer şekilde sınıflandırma yapar. Ancak tahminler sadece ikili şekildedir.

5.14.1 Çalışma Prensibi ve Adımları

$$P(C_k|x) = \frac{P(x|C_k) \cdot P(C_k)}{P(x)}$$

- $P(A|B)$, B koşulu altında A'nın olasılığını temsil eder.
- $P(A)$, A'nın önceden bilinen veya gözlemlenen olasılığını temsil eder.
- $P(B|A)$, A'nın verildiği durumda B'nin olasılığını temsil eder.
- $P(B)$, B'nin önceden bilinen veya gözlemlenen olasılığını temsil eder.

6 Çalışma Örnekleri

Yazılımı kullanmak için ilk olarak 'yıl' bilgisi seçilir. Daha sonra o yıla ait modeli çalıştırabilmek için girilmesi gereken özellikler kullanıcidan istenecektir. Bilgiler girildikten sonra en altta bulunan 'Tahmin Et' butonu ile tahmin işlemi gerçekleştirilir.



Turkish TV Series Classification

Yayın tarihi:

1997

SVC + RandomOverSampler + k=23 + isotonic

Dizi adı uzunluğu:

0,00

Dizi adındaki kelime sayısı:

0,00

Aile türünde mi ? (0-Hayır, 1-Evet):

0,00

Figure 2: Tahmin ekranı.

7 Sonuçlar ve Yorumlanması

7.1 1997 yılına ait model sonuçları

1997 yılı için en iyi performansı %60 sınıma doğruluğu, %50 F1 skoru, %46.66 kesinlik skoru ve %46.66 duyarlılık skoru ile SVC modeli vermiştir.

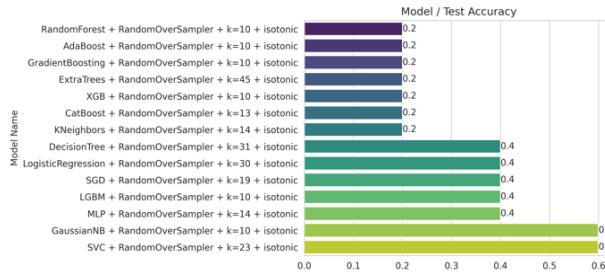


Figure 3: 1997 yılına ait model test doğrulukları.

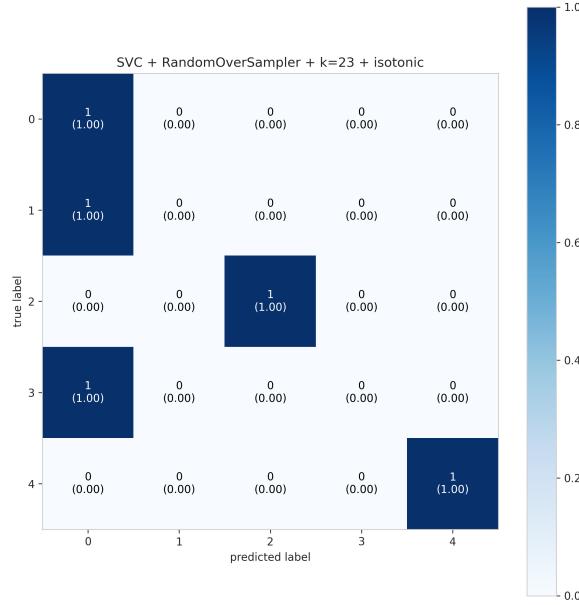


Figure 4: SVC modeline ait karmaşıklık matrisi

7.2 1998 yılına ait model sonuçları

1998 yılı için en iyi performansı %60 sinama doğruluğu, %50 F1 skoru, %46.66 kesinlik skoru ve %46.66 duyarlılık skoru ile XGB modeli vermiştir.

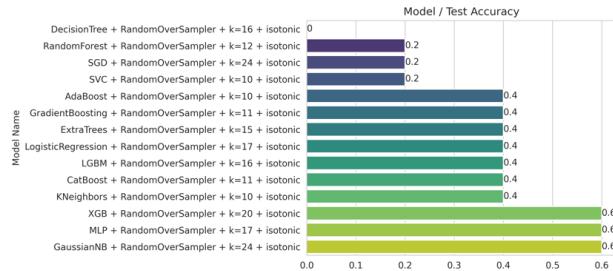


Figure 5: 1998 yılına ait model test doğrulukları.

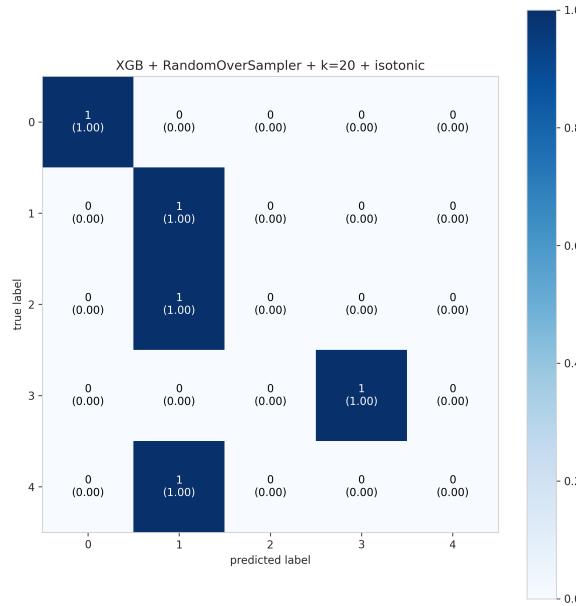


Figure 6: XGB modeline ait karmaşıklık matrisi

7.3 1999 yılına ait model sonuçları

1999 yılı için en iyi performansı %60 sinama doğruluğu, %53.33 F1 skoru, %50 kesinlik skoru ve %50 duyarlılık skoru ile Gaussian Naive Bayes modeli vermiştir.

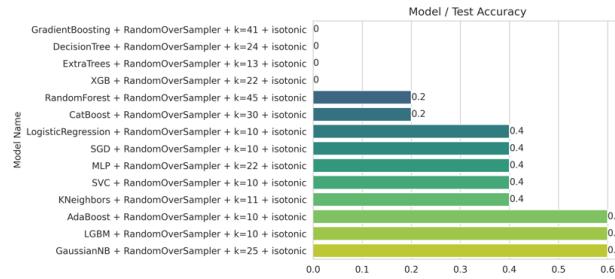


Figure 7: 1999 yılına ait model test doğrulukları.

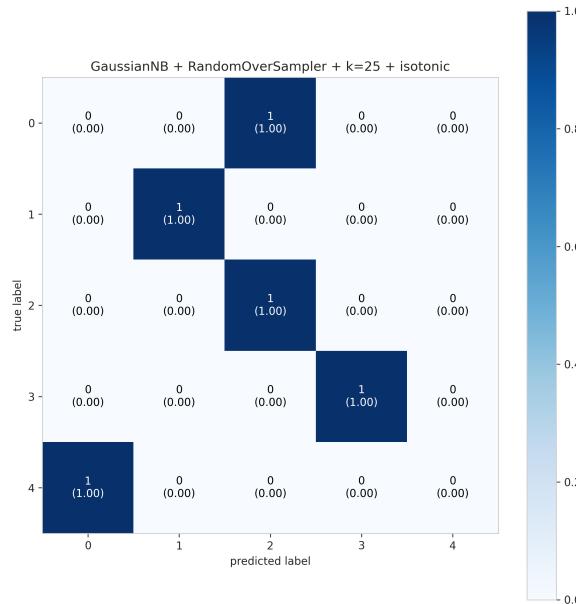


Figure 8: Gaussian Naive Bayes modeline ait karmaşılık matrisi

7.4 2000 yılına ait model sonuçları

2000 yılı için en iyi performansı %40 sinama doğruluğu, %32 F1 skoru, %26.66 kesinlik skoru ve %26.66 duyarlılık skoru ile SVC modeli vermiştir.

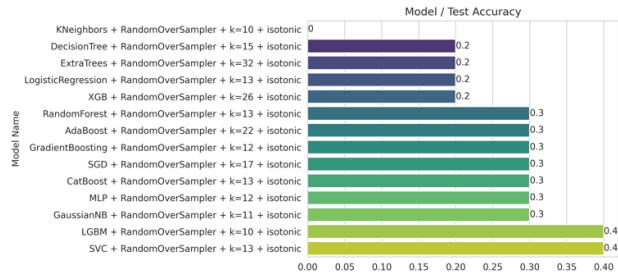


Figure 9: 2000 yılına ait model test doğrulukları.

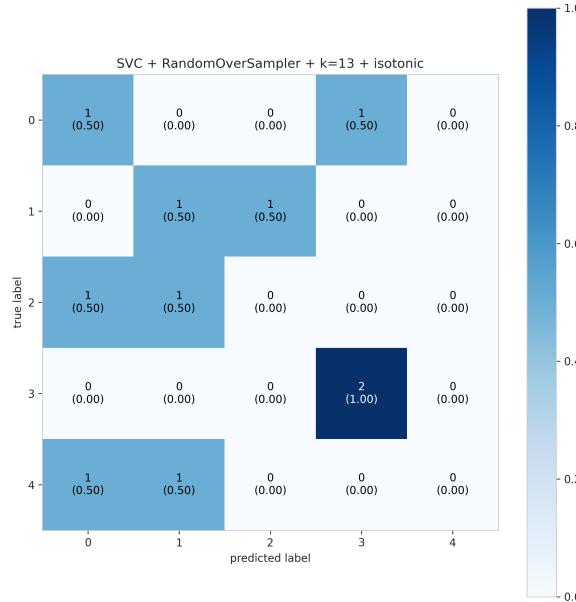


Figure 10: SVC modeline ait karmaşaklı matrisi

7.5 2001 yılına ait model sonuçları

2001 yılı için en iyi performansı %50 sinama doğruluğu, %44.76 F1 skoru, %48 kesinlik skoru ve %48 duyarlılık skoru ile CatBoost modeli vermiştir.

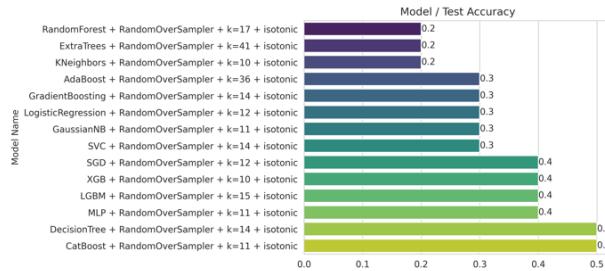


Figure 11: 2001 yılına ait model test doğrulukları.

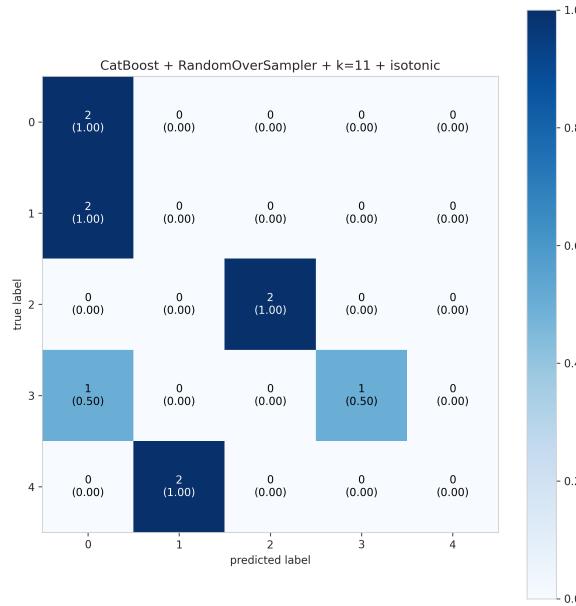


Figure 12: CatBoost modeline ait karmaşıklık matrisi

7.6 2002 yılına ait model sonuçları

2002 yılı için en iyi performansı %60 sinama doğruluğu, %61.33 F1 skoru, %76.66 kesinlik skoru ve %76.66 duyarlılık skoru ile Logistic Regression modeli vermiştir.

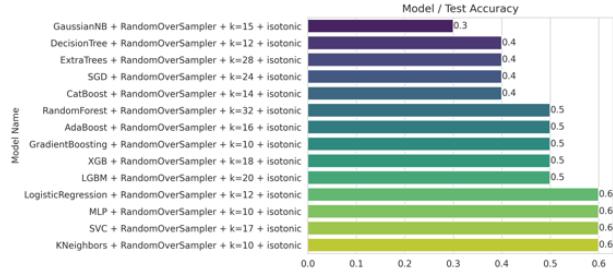


Figure 13: 2002 yılına ait model test doğrulukları.

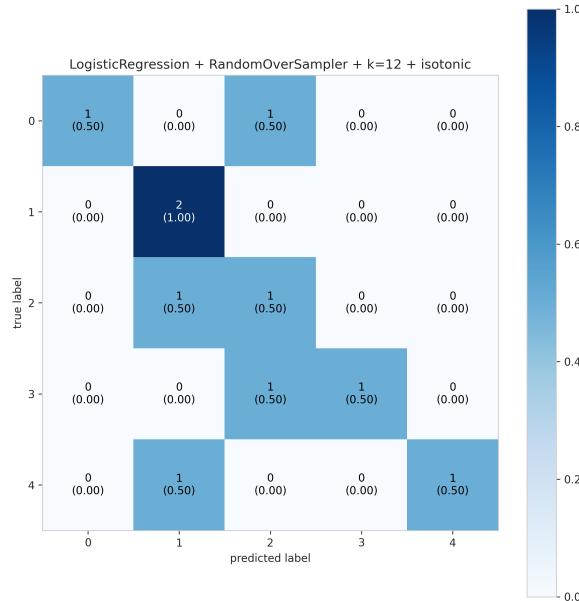


Figure 14: Logistic Regression modeline ait karmaşıklık matrisi

7.7 2003 yılına ait model sonuçları

2003 yılı için en iyi performansı %70 sinama doğruluğu, %62 F1 skoru, %56.66 kesinlik skoru ve %56.66 duyarlılık skoru ile SVC modeli vermiştir.

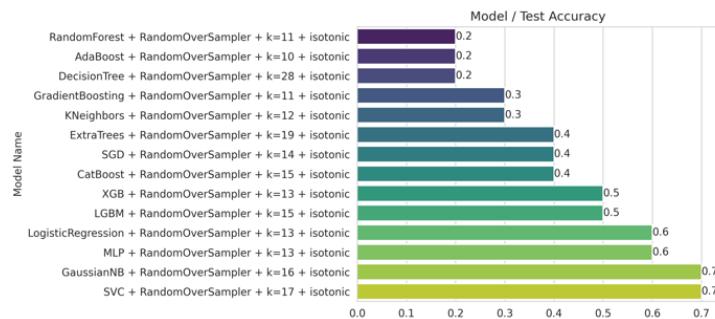


Figure 15: 2003 yılına ait model test doğrulukları.

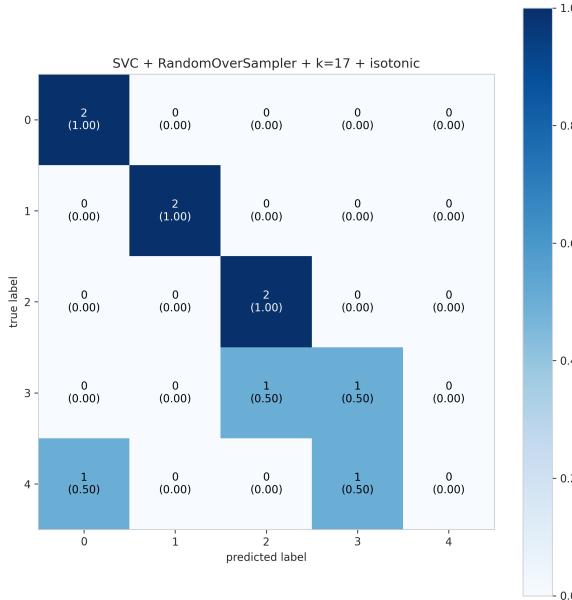


Figure 16: SVC modeline ait karmaşıklık matrisi

7.8 2004 yılına ait model sonuçları

2004 yılı için en iyi performansı %40 sinama doğruluğu, %40.38 F1 skoru, %50.66 kesinlik skoru ve %50.66 duyarlılık skoru ile SVC modeli vermiştir.

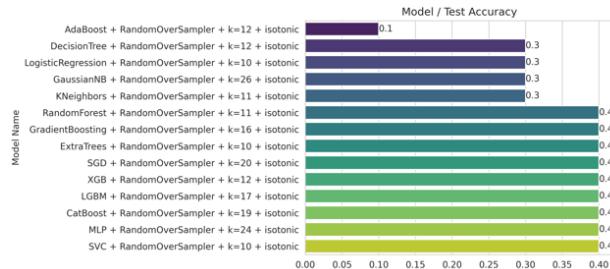


Figure 17: 2004 yılına ait model test doğrulukları.

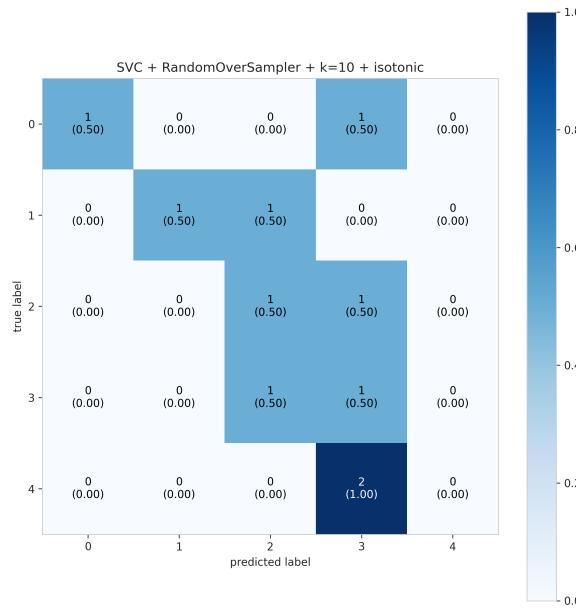


Figure 18: SVC modeline ait karmaşılık matrisi

7.9 2005 yılına ait model sonuçları

2005 yılı için en iyi performansı %60 sinama doğruluğu, %47.42 F1 skoru, %41.33 kesinlik skoru ve %41.33 duyarlılık skoru ile Gradient Boosting modeli vermiştir.

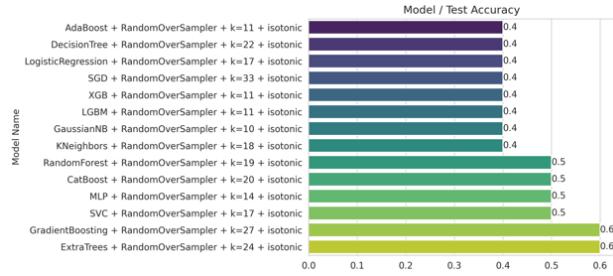


Figure 19: 2005 yılına ait model test doğrulukları.

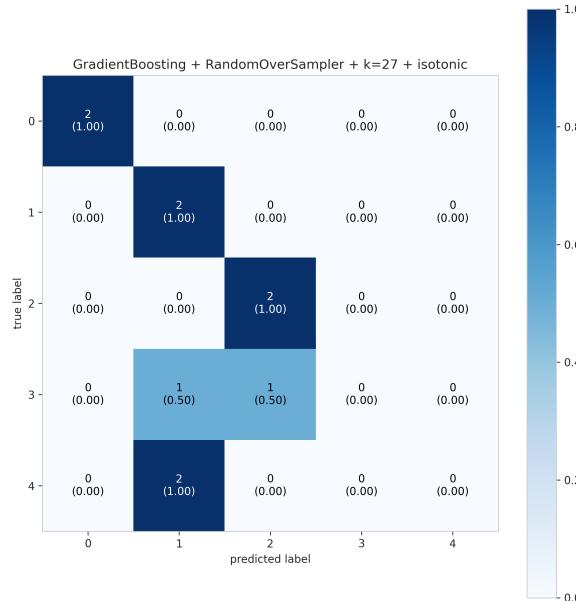


Figure 20: Gradient Boosting modeline ait karmaşıklık matrisi

7.10 2006 yılına ait model sonuçları

2006 yılı için en iyi performansı %60 sinama doğruluğu, %52.66 F1 skoru, %53.33 kesinlik skoru ve %53.33 duyarlılık skoru ile Logistic Regression modeli vermiştir.

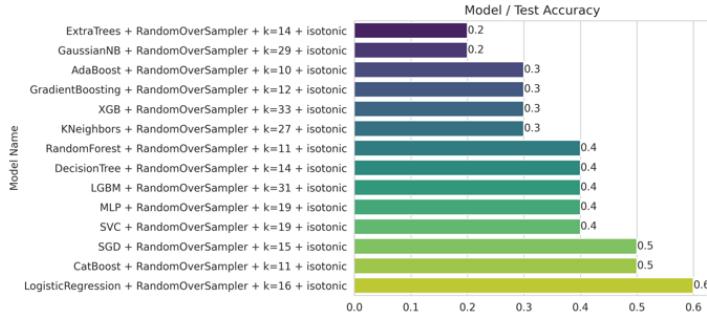


Figure 21: 2006 yılına ait model test doğrulukları.

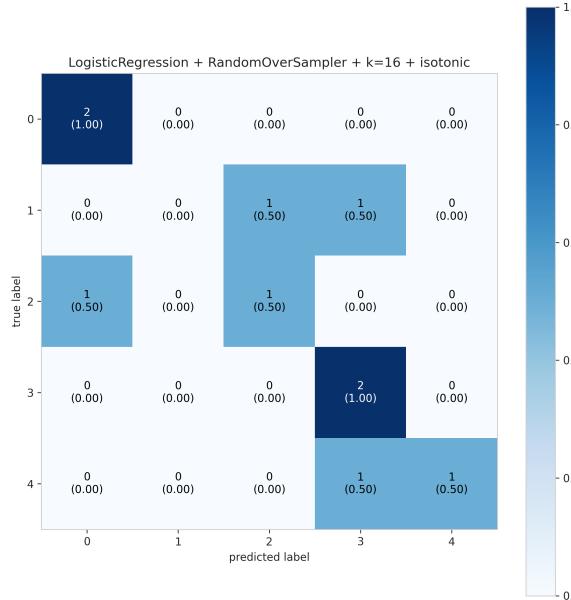


Figure 22: Logistic Regression modeline ait karmaşıklık matrisi

7.11 2007 yılına ait model sonuçları

2007 yılı için en iyi performansı %60 sinama doğruluğu, %61.33 F1 skoru, %76.66 kesinlik skoru ve %76.66 duyarlılık skoru ile MLP modeli vermiştir.

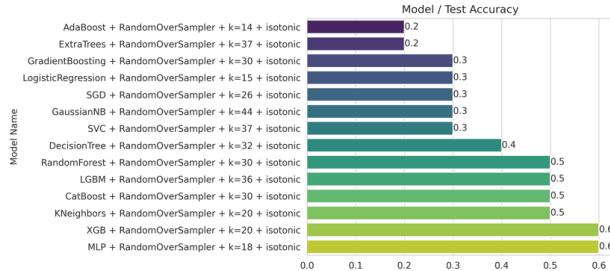


Figure 23: 2007 yılına ait model test doğrulukları.

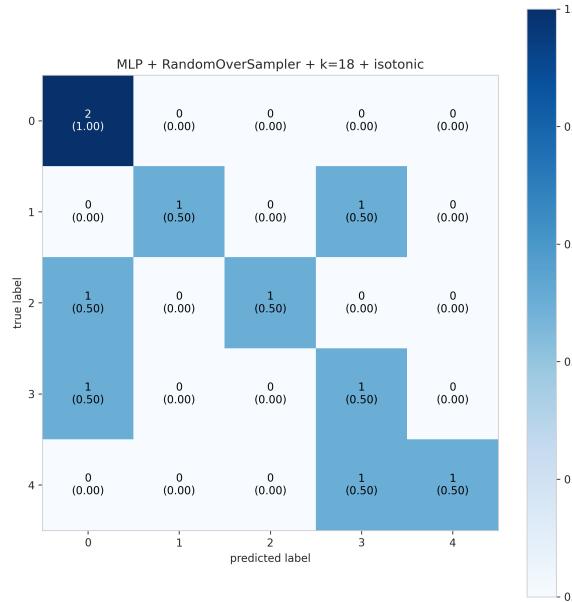


Figure 24: MLP modeline ait karmaşıklık matrisi

7.12 2008 yılına ait model sonuçları

2008 yılı için en iyi performansı %50 sinama doğruluğu, %44.76 F1 skoru, %48 kesinlik skoru ve %48 duyarlılık skoru ile XGB modeli vermiştir.

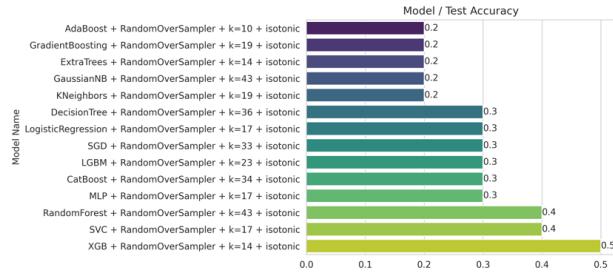


Figure 25: 2008 yılına ait model test doğrulukları.

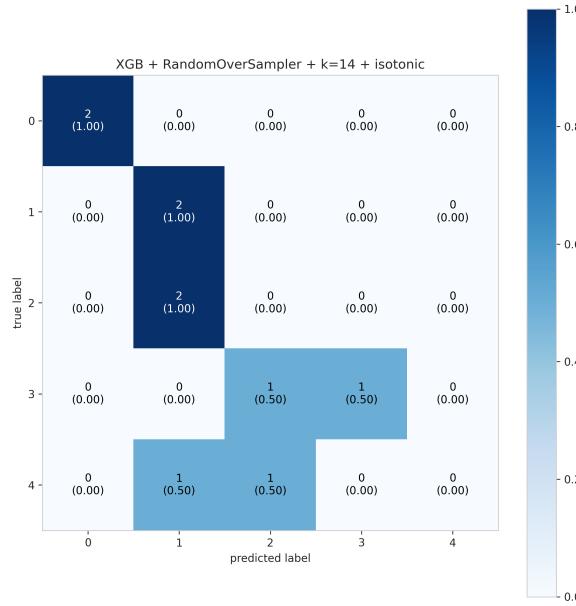


Figure 26: XGB modeline ait karmaşıklık matrisi

7.13 2009 yılına ait model sonuçları

2009 yılı için en iyi performansı %60 sinama doğruluğu, %58.09 F1 skoru, %68 kesinlik skoru ve %68 duyarlılık skoru ile Decision Tree modeli vermiştir.

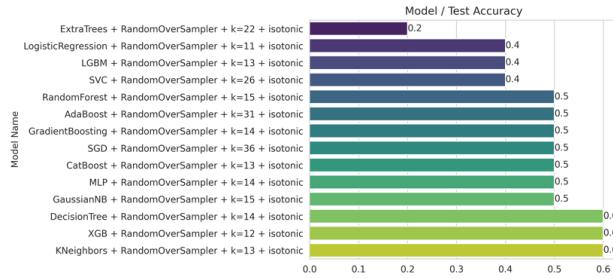


Figure 27: 2009 yılına ait model test doğrulukları.

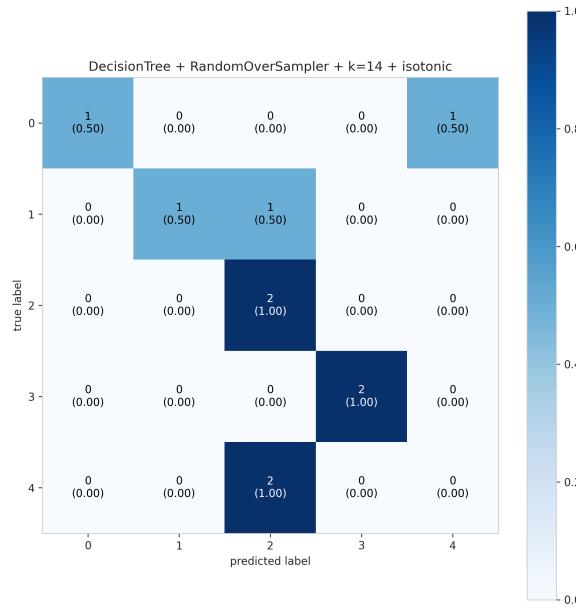


Figure 28: Decision Tree modeline ait karmaşıklık matrisi

7.14 2010 yılına ait model sonuçları

2010 yılı için en iyi performansı %60 sinama doğruluğu, %61.42 F1 skoru, %78 kesinlik skoru ve %78 duyarlılık skoru ile Logistic Regression modeli vermiştir.

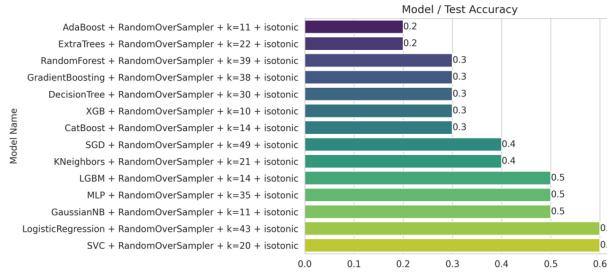


Figure 29: 2010 yılına ait model test doğrulukları.

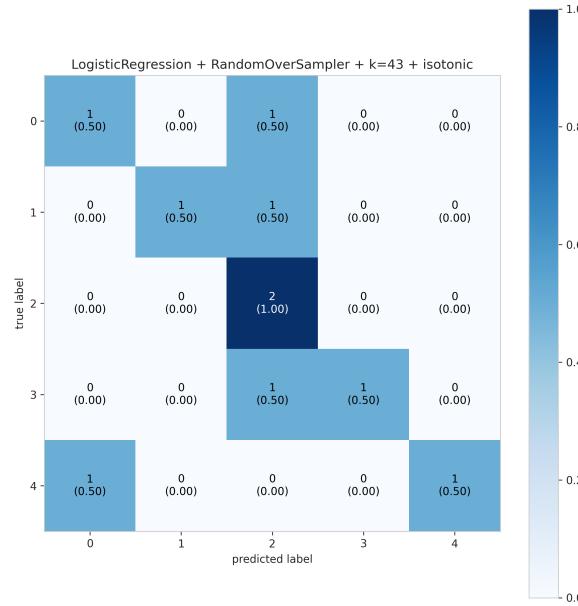


Figure 30: Logistic Regression modeline ait karmaşılık matrisi

7.15 2011 yılına ait model sonuçları

2011 yılı için en iyi performansı %40 sınıma doğruluğu, %32 F1 skoru, %26.66 kesinlik skoru ve %26.66 duyarlılık skoru ile Logistic Regression modeli vermiştir.

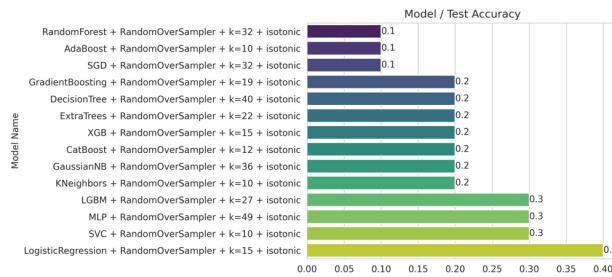


Figure 31: 2011 yılına ait model test doğrulukları.

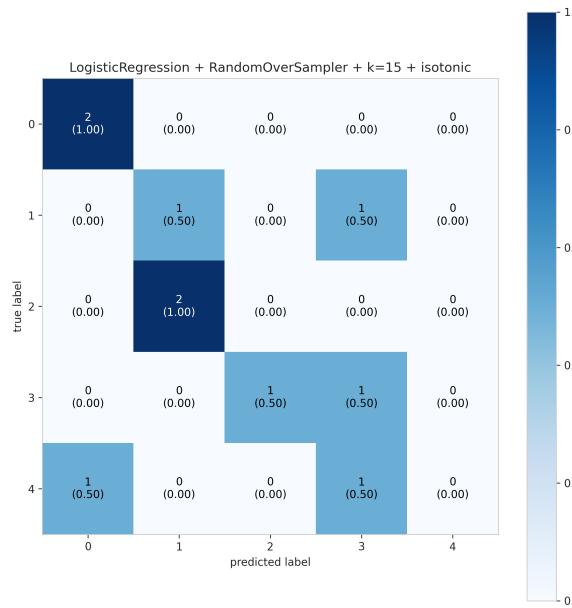


Figure 32: Logistic Regression modeline ait karmaşılık matrisi

7.16 2012 yılına ait model sonuçları

2012 yılı için en iyi performansı %33.33 sinama doğruluğu, %34.66 F1 skoru, %52.38 kesinlik skoru ve %52.38 duyarlılık skoru ile MLP modeli vermiştir.

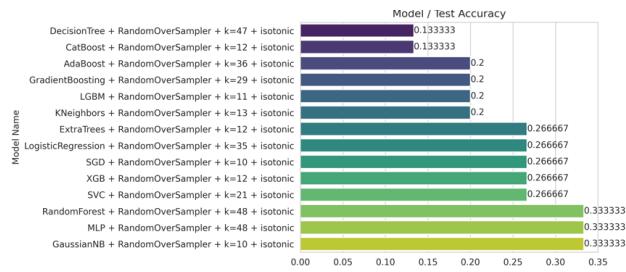


Figure 33: 2012 yılına ait model test doğrulukları.

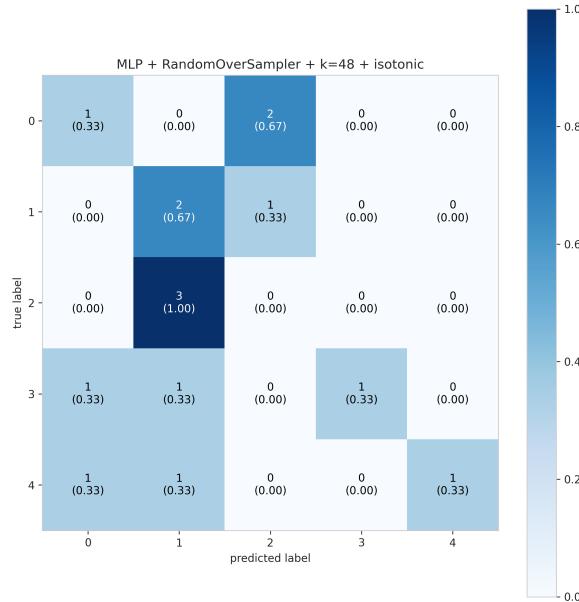


Figure 34: MLP modeline ait karmaşılık matrisi

7.17 2013 yılına ait model sonuçları

2013 yılı için en iyi performansı %53.33 sinama doğruluğu, %49.57 F1 skoru, %49 kesinlik skoru ve %49 duyarlılık skoru ile SVC modeli vermiştir.

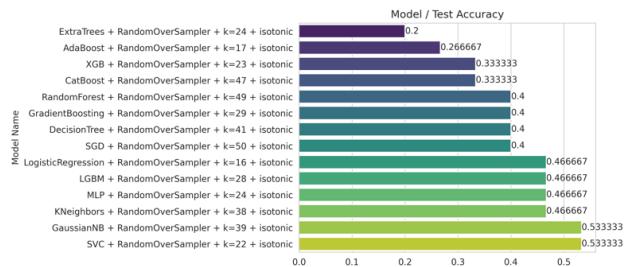


Figure 35: 2013 yılına ait model test doğrulukları.

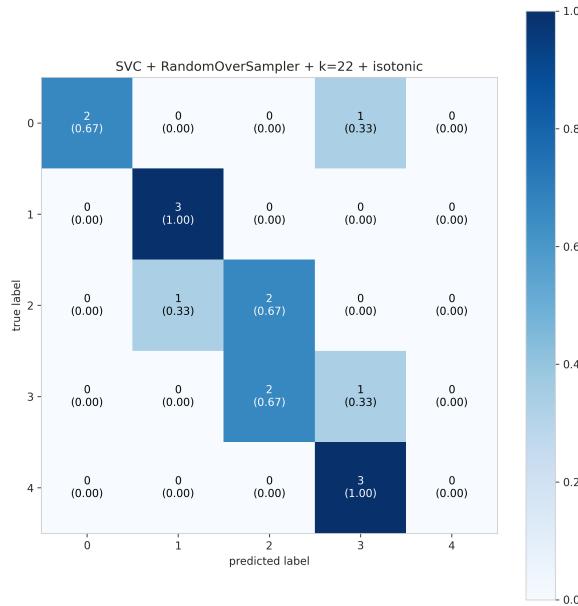


Figure 36: SVC modeline ait karmaşılık matrisi

7.18 2014 yılına ait model sonuçları

2014 yılı için en iyi performansı %40 sinama doğruluğu, %38 F1 skoru, %46.66 kesinlik skoru ve %46.66 duyarlılık skoru ile Gaussian Naive Bayes modeli vermiştir.

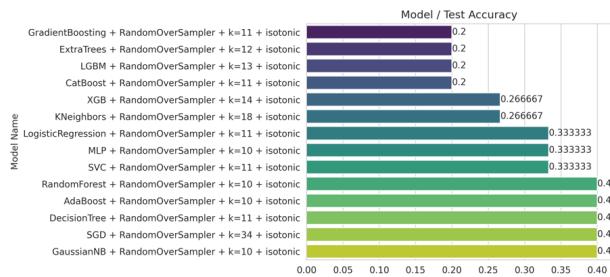


Figure 37: 2014 yılına ait model test doğrulukları.

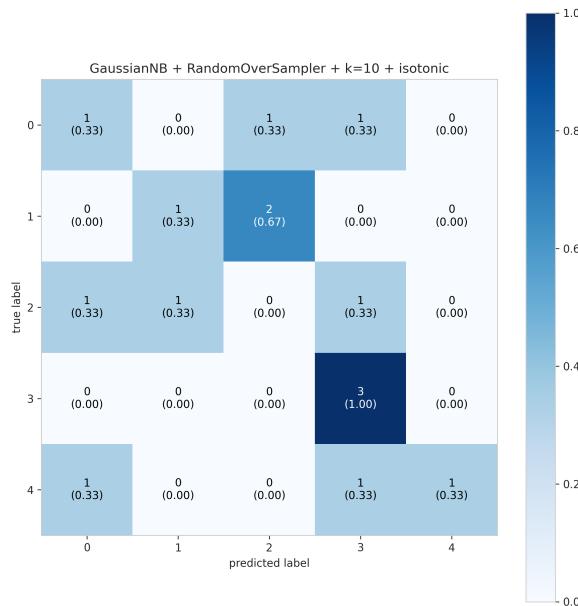


Figure 38: Gaussian Naive Bayes modeline ait karmaşılık matrisi

7.19 2015 yılına ait model sonuçları

2015 yılı için en iyi performansı %60 sinama doğruluğu, %52.14 F1 skoru, %47 kesinlik skoru ve %47 duyarlılık skoru ile MLP modeli vermiştir.

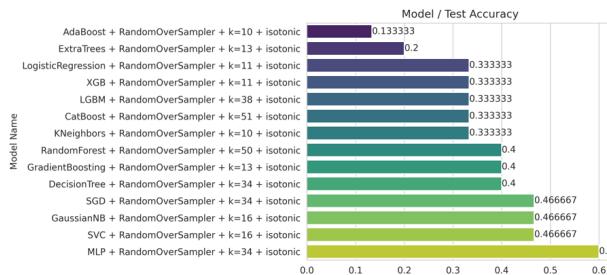


Figure 39: 2015 yılına ait model test doğrulukları.

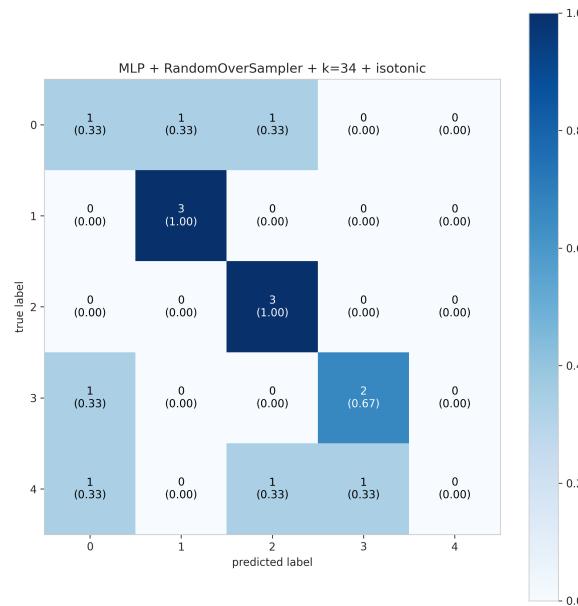


Figure 40: MLP modeline ait karmaşılık matrisi

7.20 2016 yılına ait model sonuçları

2016 yılı için en iyi performansı %66.66 sinama doğruluğu, %65.23 F1 skoru, %71.66 kesinlik skoru ve %71.66 duyarlılık skoru ile SVC modeli vermiştir.

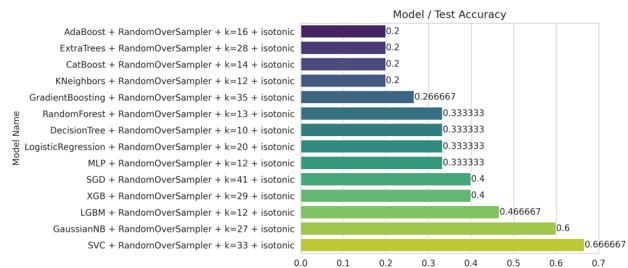


Figure 41: 2016 yılına ait model test doğrulukları.

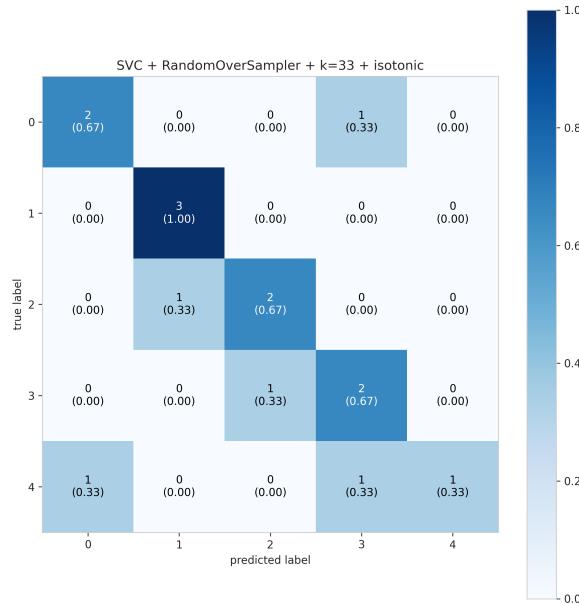


Figure 42: SVC modeline ait karmaşaklı matrisi

7.21 2017 yılına ait model sonuçları

2017 yılı için en iyi performansı %53.33 sinama doğruluğu, %45.90 F1 skoru, %59.5 kesinlik skoru ve %59.5 duyarlılık skoru ile MLP modeli vermiştir.

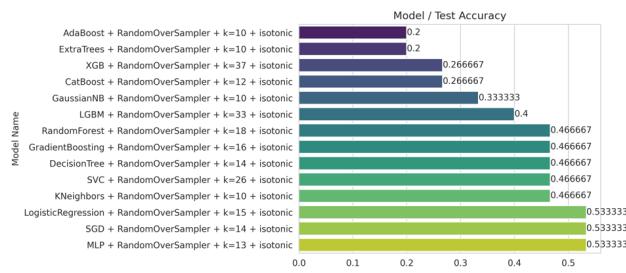


Figure 43: 2017 yılına ait model test doğrulukları.

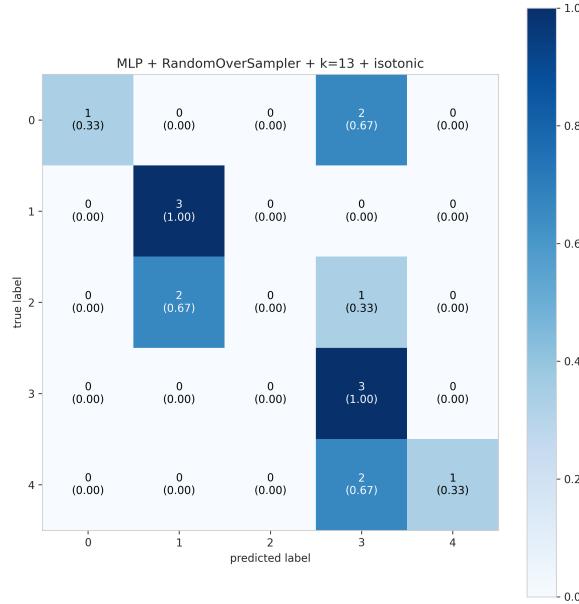


Figure 44: MLP modeline ait karmaşılık matrisi

7.22 2018 yılına ait model sonuçları

2018 yılı için en iyi performansı %53.33 sinama doğruluğu, %53.65 F1 skoru, %70 kesinlik skoru ve %70 duyarlılık skoru ile KNeighbors modeli vermiştir.

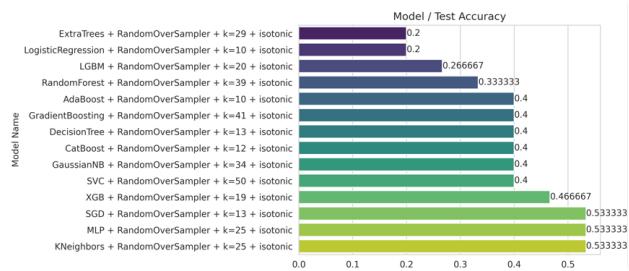


Figure 45: 2018 yılına ait model test doğrulukları.

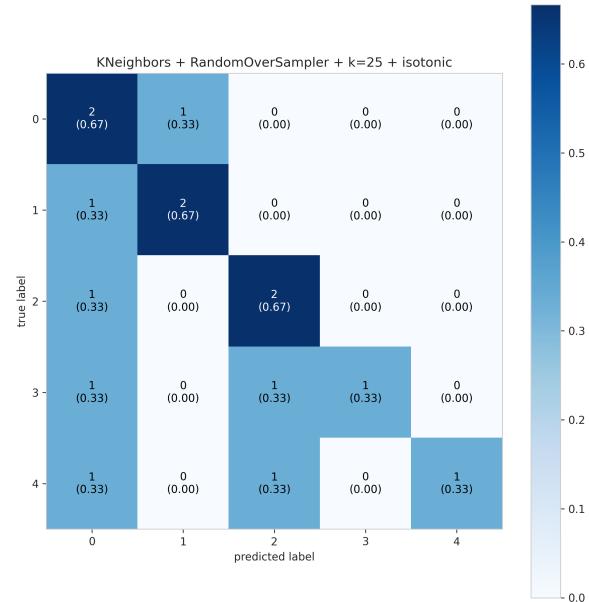


Figure 46: KNeighbors modeline ait karmaşılık matrisi

7.23 2019 yılına ait model sonuçları

2019 yılı için en iyi performansı %53.33 sınıma doğruluğu, %52 F1 skoru, %68.57 kesinlik skoru ve %68.57 duyarlılık skoru ile XGB modeli vermiştir.

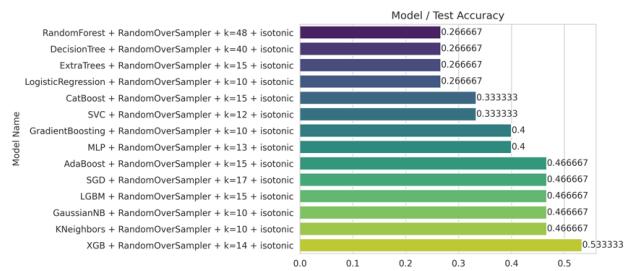


Figure 47: 2019 yılına ait model test doğrulukları.

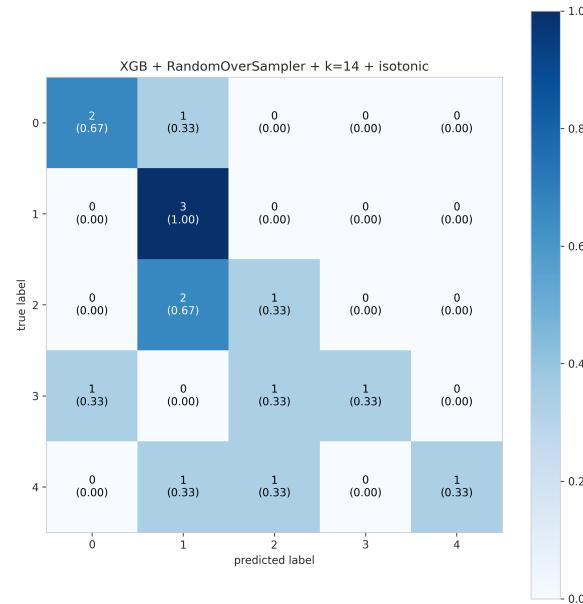


Figure 48: XGB modeline ait karmaşılık matrisi

7.24 2020 yılına ait model sonuçları

2020 yılı için en iyi performansı %60 sinama doğruluğu, %58.33 F1 skoru, %73.33 kesinlik skoru ve %73.33 duyarlılık skoru ile MLP modeli vermiştir.

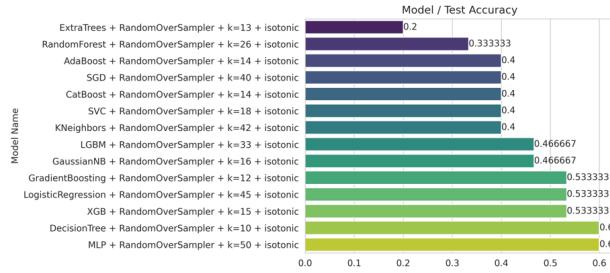


Figure 49: 2020 yılına ait model test doğrulukları.

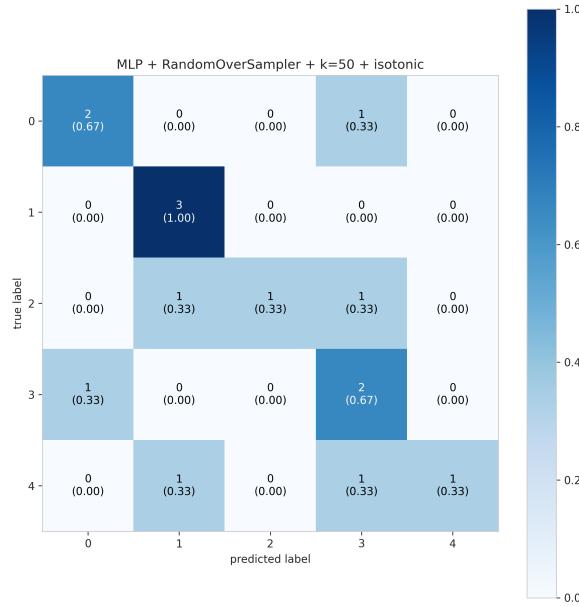


Figure 50: MLP modeline ait karmaşılık matrisi

7.25 2021 yılına ait model sonuçları

2021 yılı için en iyi performansı %53.33 sinama doğruluğu, %49.71 F1 skoru, %53.57 kesinlik skoru ve %53.57 duyarlılık skoru ile LGBM modeli vermiştir.

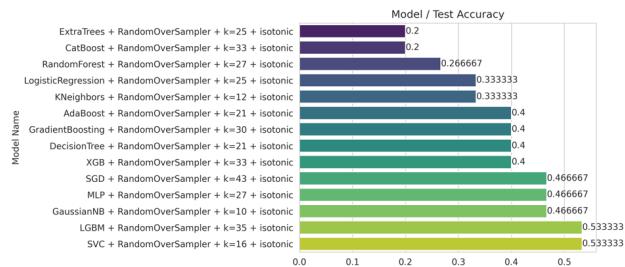


Figure 51: 2021 yılına ait model test doğrulukları.

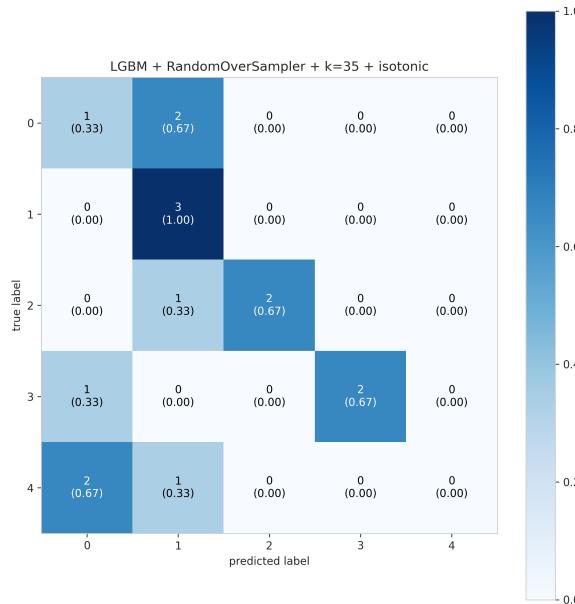


Figure 52: LGBM modeline ait karmaşılık matrisi

7.26 2022 yılına ait model sonuçları

2022 yılı için en iyi performansı %33.33 sınıma doğruluğu, %27.5 F1 skoru, %44.61 kesinlik skoru ve %44.61 duyarlılık skoru ile KNeighbors modeli vermiştir.

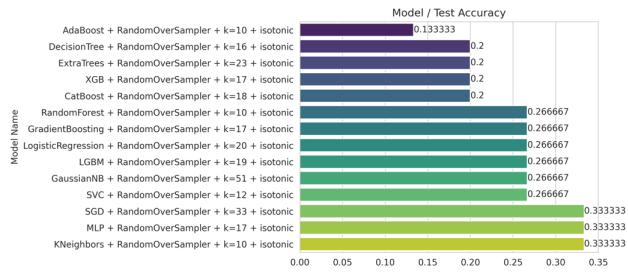


Figure 53: 2022 yılına ait model test doğrulukları.

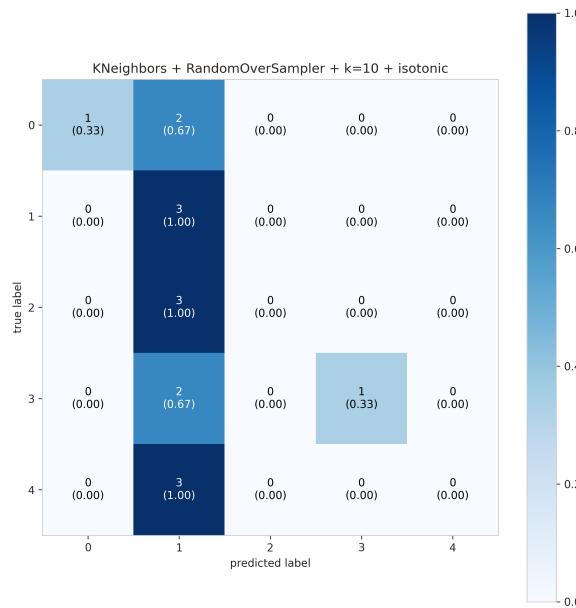


Figure 54: KNeighbors modeline ait karmaşılık matrisi

7.27 2023 yılına ait model sonuçları

2023 yılı için en iyi performansı %46.66 sinama doğruluğu, %39.76 F1 skoru, %35.33 kesinlik skoru ve %35.33 duyarlılık skoru ile Random Forest modeli vermiştir.

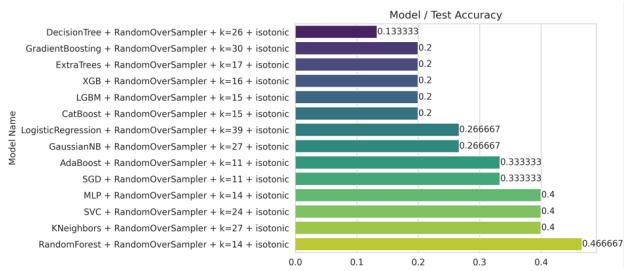


Figure 55: 2023 yılına ait model test doğrulukları.

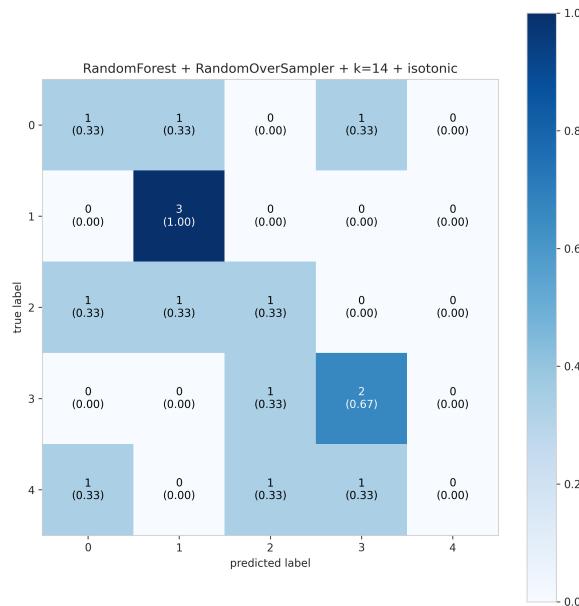


Figure 56: Random Forest modeline ait karmaşıklık matrisi

7.28 Sonuç

Yıllara göre en iyi modeller ve skorları aşağıdaki tabloda verilmiştir.

Table 2: Yıllara göre en iyi modeller.

Yıl	Model Adı	ES	SS	ED	SD	F1	PRE	REC
1997	SVC + ROS + k=23 + isotonic	0.0489	0.0144	0.97	0.6	0.5	0.4666	0.4666
1998	XGB + ROS + k=20 + isotonic	0.4152	0.0087	0.95	0.6	0.5333	0.4666	0.4666
1999	GaussianNB + ROS + k=25 + isotonic	0.0248	0.005	0.8666	0.6	0.5333	0.5	0.5
2000	SVC + ROS + k=13 + isotonic	0.0621	0.0154	0.8666	0.4	0.32	0.2666	0.2666
2001	CatBoost + ROS + k=11 + isotonic	5.0491	0.0091	1.0	0.5	0.4476	0.48	0.48
2002	LogisticRegression + ROS + k=12 + isotonic	0.0750	0.0045	0.72	0.6	0.6133	0.7666	0.7666
2003	SVC + ROS + k=17 + isotonic	0.0335	0.0060	0.9935	0.7	0.62	0.5666	0.5666
2004	SVC + ROS + k=10 + isotonic	0.0304	0.005	0.8972	0.4	0.4038	0.5066	0.5066
2005	GradientBoosting + ROS + k=27 + isotonic	4.9267	0.0075	1.0	0.6	0.4742	0.4133	0.4133
2006	LogisticRegression + ROS + k=16 + isotonic	0.1633	0.0058	0.7493	0.6	0.5266	0.5333	0.5333
2007	MLP + ROS + k=18 + isotonic	11.3033	0.0178	0.7594	0.6	0.6133	0.7666	0.7666
2008	XGB + ROS + k=14 + isotonic	0.5731	0.0089	0.9569	0.5	0.4476	0.48	0.48
2009	DecisionTree + ROS + k=14 + isotonic	0.0449	0.0063	0.9211	0.6	0.5809	0.6799	0.6799
2010	LogisticRegression + ROS + k=43 + isotonic	0.1901	0.0045	0.7697	0.6	0.6142	0.78	0.78
2011	LogisticRegression + ROS + k=15 + isotonic	0.1639	0.0047	0.6832	0.4	0.32	0.2666	0.2666
2012	MLP + ROS + k=48 + isotonic	10.1280	0.0103	1.0	0.33	0.3466	0.5238	0.5238
2013	SVC + ROS + k=22 + isotonic	0.5853	0.0099	0.6185	0.5333	0.4957	0.49	0.49
2014	GaussianNB + ROS + k=10 + isotonic	0.0345	0.0062	0.5855	0.4	0.38	0.4666	0.466
2015	MLP + ROS + k=34 + isotonic	30.1362	0.0074	0.9740	0.6	0.5214	0.47	0.47
2016	SVC + ROS + k=13 + isotonic	1.5583	0.0142	0.5848	0.6666	0.6523	0.7166	0.7166
2017	MLP + ROS + k=13 + isotonic	13.2263	0.0073	0.6733	0.5333	0.4590	0.5950	0.5950
2018	KNeighbors + ROS + k=25 + isotonic	0.1011	0.0116	0.8458	0.5333	0.5365	0.7	0.7
2019	XGB + ROS + k=14 + isotonic	1.3034	0.0105	0.9093	0.5333	0.52	0.6857	0.6857
2020	MLP + ROS + k=50 + isotonic	18.1571	0.0173	0.7064	0.6	0.5833	0.7333	0.7333
2021	LGBM + ROS + k=35 + isotonic	0.9594	0.0101	0.8786	0.5333	0.4971	0.5357	0.5357
2022	KNeighbors + ROS + k=10 + isotonic	0.0988	0.0088	0.9984	0.3333	0.275	0.4461	0.4461
2023	RandomForest + ROS + k=14 + isotonic	3.5568	0.0676	0.7268	0.4666	0.3976	0.3533	0.3533

8 Gelecekte Yapılabilecekler

Proje için gelecekteki adımların yapılması hedeflenmektedir:

- Veri setinin daha geniş ve temsilci olabilmesi için yeni verilerin eklenmesi.
- Dizi başarısını etkileyebilecek daha fazla özellik eklenmesi.
- Modelin gerçek dünya uygulama alanları için endüstriyle daha sıkı bir entegrasyonunu araştırma.
- Modelin güvenilirliğini ve güvenliğini artırmak için önlemler alma.
- Kullanıcı dostu bir arayüz oluşturma.
- Kullanıcı geri bildirimlerini düzenli olarak toplama ve modele entegre etme.

9 Kullanılan Kaynaklar

References

- [1] Deloitte *tr-media-tv-report*. 2014. Erişim Tarihi: 2024-01-02.
<https://www2.deloitte.com/content/dam/Deloitte/tr/Documents/technology-media-telecommunications/tr-media-tv-report.pdf>
- [2] Türk dizileri listesi *tr-media-tv-report*. Temmuz 2020. Erişim Tarihi: 2024-01-02.
https://tr.wikipedia.org/wiki/T%C3%BCrk_dizileri_listesi