

KOCAELİ ÜNİVERSİTESİ
MÜHENDİSLİK FAKÜLTESİ

BİLGİSAYAR MÜHENDİSLİĞİ BÖLÜMÜ

BİTİRME TEZİ

**MAKİNE ÖĞRENMESİ YÖNTEMLERİ İLE GENETİK VARYANTLARIN
PATOJENİTE ANALİZİ BRCA1, BRCA2 GENLERİ UYGULAMASI**

Muhammed Emre Kara

Onur Kaplan

KOCAELİ 2020

KOCAELİ ÜNİVERSİTESİ

MÜHENDİSLİK FAKÜLTESİ

BİLGİSAYAR MÜHENDİSLİĞİ BÖLÜMÜ

BİTİRME TEZİ

**MAKİNE ÖĞRENMESİ YÖNTEMLERİ İLE GENETİK VARYANTLARIN
PATOJENİTE ANALİZİ BRCA1, BRCA2 GENLERİ UYGULAMASI**

Muhammed Emre KARA

Onur KAPLAN

Prof.Dr. Nevcihan Duru

Danışman, Kocaeli Üniv.

.....

Doç.Dr. Sevinç İlhan Omurca

Jüri Üyesi, Kocaeli Üniv.

.....

Arş. Gör. Hüseyin Aktaş

Jüri Üyesi, Kocaeli Üniv.

.....

Tezin Savunulduğu Tarih: 23.06.2020

ÖNSÖZ VE TEŞEKKÜR

Bu çalışmanın amacı, gen dizilimi verilerinin, tanı, teşhis veya destek amaçlı kullanımına hizmet edebilmek için veri madenciliği yöntemlerinin uygulanmasıdır.

Geliştirilecek uygulamanın detay ve hedeflerine Kocaeli Üniversitesi Tıbbi Genetik Ana Bilim Dalı akademik personellerinin destek ve yardımlarıyla karar verilmiştir.

Tez çalışmamızda desteğini esirgemeyen, çalışmalarımıza yön veren, bize güvenen ve yüreklendiren danışmanımız Doç. Dr. Sevinç İlhan Omurca ‘ya sonsuz teşekkürlerimizi sunarız.

Tez çalışmamızın tüm aşamalarında bilgi ve destekleriyle katkıda bulunan hocamız Doç. Dr. Naci Çine’ye ve Bilge Dursun’a teşekkür ediyoruz.

Hayatımız boyunca bize güç veren en büyük destekçilerimiz, her aşamada sıkıntılarımızı ve mutluluklarımızı paylaşan sevgili ailelerimize teşekkürlerimizi sunarız.

Temmuz – 2020

Muhammed Emre KARA, Onur KAPLAN

Bu dokümandaki tüm bilgiler, etik ve akademik kurallar çerçevesinde elde edilip sunulmuştur. Ayrıca yine bu kurallar çerçevesinde kendime ait olmayan ve kendimin üretmediği ve başka kaynaklardan elde edilen bilgiler ve materyaller (text, resim, şekil, tablo vb.) gerekli şekilde referans edilmiş ve dokümanda belirtilmiştir.

Öğrenci No: 160202094

Adı Soyadı: Muhammed Emre KARA

İmza:

Öğrenci No: 160202061

Adı Soyadı: Onur KAPLAN

İmza:

İÇERİK

ŞEKİLLER DİZİNİ	vi
TABLolar DİZİNİ	vii
DENKLEMLER DİZİNİ	viii
SİMGELER VE KISALTMALAR DİZİNİ	ix
ÖZET	x
ABSTRACT	xi
1. GİRİŞ	12
2. GENEL BİLGİLER	14
2.1. Meme Kanseri	14
2.2. DNA Nedir?	15
2.3. Gen Nedir?	16
2.4. BRCA Nedir?	17
2.5. Gen Dizilime	17
2.6. İnsan Genom Projesi	19
2.7. Varyant Nedir?	20
2.8. BRCA Varyantları	20
2.9. Patojenite Nedir?	21
2.10. CADD	21
2.11. Myvariant.info	22
2.12. Makine Öğrenmesi Nedir	22
2.13. Sınıflandırma nedir	24
2.13.1. Makine öğreniminde sınıflandırma terminolojileri	25
2.13.2. Sınıflandırmada öğrenen türleri	25
2.14. Gradient Boosting	26
2.15. XGBOOST	27
2.15.1. Paralleleştirme	29
2.15.2. Ağaç budaması	30
2.15.3. XGBoost'un avantajları	30
2.15.4. XGBoost'un dezavantajları	30
2.16. LightGBM	31
2.16.1. Light GBM'nin avantajları	32

2.17.	CatBoost.....	33
2.17.1.	CatBoost kütüphanesinin avantajları	37
2.18.	Lojistik Regresyon	37
2.19.	KNN	39
2.20.	SVM	40
2.21.	Karar Ağacı Sınıflandırma	42
2.22.	Naive Bayes	44
2.23.	Random Forest.....	45
3.	UYGULANAN YÖNTEMLER	47
3.1.	Veri Seçimi.....	47
3.2.	Veri Hazırlama.....	48
3.3.	Veri Önleme.....	49
3.3.1.	Krite 1'e göre silinen sütunlar	50
3.3.2.	Kriter 2'ye göre silinen sütunlar	50
3.4.	Basit Sınıflandırma Metotları Uygulaması.....	52
3.4.1.	Logistic Regression	53
3.4.2.	KNN	54
3.4.3.	SVM	55
3.4.4.	Gaussian Naive Bayes.....	56
3.4.5.	Decision Tree Classifier	57
3.4.6.	Random Forest Classifier	58
3.5.	Boosted Tree Sınıflandırma Metotları Uygulaması.....	59
3.5.1.	XGBoost:.....	60
3.5.2.	LightGBM.....	61
3.5.3.	CatBoost.....	63
4.	ÇIKTILAR	65
4.1.	XGBoost:.....	66
4.2.	LightGBM	68
4.3.	CatBoost:.....	70
4.4.	Random Forest Classifier:	72
4.5.	Logistic Regression:.....	73
4.6.	VUS Varyantların Patojenite Tahminleri	73

5. SONUÇLAR VE ÖNERİLER	75
KAYNAKÇA.....	77
ÖZGEÇMİŞ	80

ŞEKİLLER DİZİNİ

Şekil 2.1-1 Kadınlarda En Sık Görülen 10 Kanserin Yaşa Göre Standardize Edilmiş Hızları. (Dünya Standart Nüfusu, 100.000 Kişide) [4]	15
Şekil 2.5-1 Gen dizileme süreçleri. [3].....	18
Şekil 2.14-1 Gradient Boosting ağaç şeması.[15].....	27
Şekil 2.15-1 Ensemble tekniklerin geçmişi.[17]	28
Şekil 2.16-1 LightGBM ağaç şeması.[21]	32
Şekil 2.17-1 CatBoost Sözde Kodu.[24]	34
Şekil 2.17-2 CatBoost temel ağaç simetriği.....	35
Şekil 2.18-1 Logistic Regression kat sayı değişimleri [28]	38
Şekil 2.19-1 KNN temsili komşular gösterimi [28].....	39
Şekil 2.20-1 SVM farklı kernel fonksiyonları temsili [28]	41
Şekil 2.20-2 SVM'de kernel trick [28]	42
Şekil 2.20-3 SVM kernel trick 2 [28]	42
Şekil 2.23-1 Rastgele Orman tahmini temsili [29].....	46
Şekil 3.2-1 Mongo DB'de verinin görünümü	48
Şekil 3.2-2 Veride yapısal dönüşüm	49
Şekil 3.4-1 Logistic Regression Classification Report Çıktısı.....	54
Şekil 3.4-2 KNN Classification Report Çıktısı	55
Şekil 3.4-3 SVM Classification Report Çıktısı.....	56
Şekil 3.4-4 Naive Bayes Classification Report Çıktısı.....	57
Şekil 3.4-5 Decision Tree Classifier Classification Report Çıktısı	58
Şekil 3.4-6 Random Forest Classifier Classification Report Çıktısı	59
Şekil 3.5-1 XGBoost Classification Report Çıktısı.....	60
Şekil 3.5-3 LightGBM Classification Report Çıktısı.....	62
Şekil 3.5-5 CatBoost Classification Report Çıktısı.....	63
Şekil 4.1-1 XGBoost Classification Report Çıktısı - Final.....	66
Şekil 4.1-2 XGBoost Feature Importance - Final	67
Şekil 4.1-3 XGBoost'un VUS varyantlar üzerine tahmininin sınıflara göre dağılımı	68
Şekil 4.2-1 LightGBM Classification Report Çıktısı - Final.....	68
Şekil 4.2-2 LightGBM Feature Importance - Final	69
Şekil 4.2-3 LightGBM'in VUS varyantlar üzerine tahmininin sınıflara göre dağılımı	70
Şekil 4.3-1 CatBoost Classification Report Çıktısı - Final	70
Şekil 4.3-2 CatBoost Feature Importance – Final.....	71
Şekil 4.3-3 CatBoost'un VUS varyantlar üzerine tahmininin sınıflara göre dağılımı	72
Şekil 4.4-1 Random Forest Classification Report Çıktısı - Final	72
Şekil 4.5-1 Logistic Regression Classification Report Çıktısı - Final	73

TABLÖLAR DİZİNİ

Tablo 2.1.1 IARC tarafından yayınlanan Globocan 2012 verilerine göre kadınlarda en sık görülen ilk beş kanserlerin dağılımı. [4]	14
Tablo 3.1.1 Veritabanlarında bulunan varyant sayıları	47
Tablo 3.1.2 dbSNP ve CADD veri tabanlarının özellik sayıları	47
Tablo 3.3.1 Doluluk Oranı sebebiyle elenen sütunlar	50

DENKLEMLER DİZİNİ

(2.14.1).....	27
(2.14.2).....	27
(2.14.3).....	27
(2.17.1).....	35
(2.17.2).....	35
(2.18.1).....	39
(2.19.1).....	40
(2.19.2).....	40
(2.20.1).....	41
(2.20.2).....	41
(2.20.3).....	41
(2.21.1).....	43
(2.21.2).....	43
(2.21.3).....	44
(2.22.1).....	45

SİMGELER VE KISALTMALAR DİZİNİ

SVM	:	Support Vektor Machine
DNA	:	Deoksiribonükleik Asit
KNN	:	K-Nearest Neighbors
KA	:	Karar Ağaçları
REST	:	Representational State Transfer
CADD	:	Combined Annotation Dependent Depletion
ACMG	:	Amerikan Tıbbi Genetik ve Genomik Koleji
SNP	:	Tek Nükleotid Polimorfizmi
IARC	:	Uluslararası Kanseri Ajansı
ACS	:	Amerikan Kanseri Derneği
NLM	:	ABD Ulusal Tıp Kütüphanesi
A/G/S/T	:	Adenin/Guanin/Sitozin/Timin
NBCF	:	ABD Ulusal Meme Kanseri Vakfı
NHGRI	:	Ulusal İnsan Genomu Araştırma Enstitüsü
HGP	:	İnsan Genom Projesi
GRC	:	Genom Referans Konsorsiyum
GBM	:	Gradient Boosting Machines
GBDT	:	Gradient Boosted Decision Trees
MSE	:	Mean Squared Error
CART	:	Classification And Regression Trees
GPU	:	Graphics Processing Unit
API	:	Application Programming Interface
JSON	:	JavaScript Object Notation
VUS	:	Variant of Uncertain Significance
BRCA	:	The Breast Cancer Gene
MEMEDER	:	Meme Sağlığı Derneği

MAKİNE ÖĞRENMESİ YÖNTEMLERİ İLE GENETİK VARYANTLARIN PATOJENİTE ANALİZİ BRCA1, BRCA2 GENLERİ UYGULAMASI

ÖZET

Meme kanseri kadınlarda görülen kanserlerin %33'ünü oluştururken, tüm kanser hastalarının ise %20'sini tehdit etmektedir. BRCA1 ve BRCA2 genleri ise bu kanser türüne kalıtsal yatkınlığı arttırmasıyla biliniyor. Bu nedenle bahsedilen genlere ait varyantların(mutasyonların) patojenitesinin(zararlılığının) anlaşılması kansere yatkınlığın belirlenmesinde kilit bir rol oynamaktadır. Buradaki bilgi eksikliğinin giderilmesi birçok insanın doğru tedavi ve bilinçli yaşam biçimiyle hastalığa yakalanmaktan kurtulmasını veya geciktirmesini sağlayabilir. Ancak bu genlerde şu ana kadar raporlanmış varyantların çoğunluğunun klinik statüsü henüz bilinmemektedir. Makine öğrenimi yöntemlerinin bu soruna etkili bir çözüm sunabileceği fikri, bu çalışmanın temel motivasyon kaynağı olmuştur.

Bu çalışmada amaç, BRCA1 ve BRCA2 genlerine ait varyantların patojenite tahminini yapabilmek üzere makine öğrenmesi modellerinin kullanılmasıdır. Çalışma sonucunda klinik statüsü(patojenite) bilinen varyantlarla eğitilen modelin test verilerinde kabul edilebilir sonuçlar üreterek klinik statüsü bilinmeyen veri setleri üzerinde de uygulanabilir duruma getirilmesi planlanmıştır. Eğitilecek modeller için gerekli veri myvariant.info platformu aracılığıyla CADD veri tabanından edinilmiştir. Çalışma kapsamında KNN, SVC, Logistic Regression gibi çeşitli temel sınıflandırıcıların yanında XGBoost, CatBoost, LightGBM gibi gelişmiş yapılar da kullanılmış ve çıktıları raporlanmıştır.

Uygulama Kocaeli Üniversitesi Tıbbi Genetik Akademisyenlerinin yardımıyla geliştirilmiştir.

Anahtar kelimeler: makine öğrenmesi, brca1, brca2, meme kanseri, biyoinformatik, dna, gen, varyant, patojenite

ANALYSIS OF GENETIC DATA BY MACHINE LEARNING METHODS

ABSTRACT

Breast cancer threatens 33% of cancers in women and 20% of all cancer patients. The BRCA1 and BRCA2 genes are known to increase hereditary susceptibility to this type of cancer. Therefore, understanding the pathogenicity (harmfulness) of variants (mutations) of said genes plays a key role in determining cancer susceptibility. Correcting the lack of information here can help many people to get rid of the disease or delay it with the right treatment and conscious lifestyle. However, the clinical status of the majority of variants reported so far in these genes is not yet known. The idea that machine learning methods can offer an effective solution to this problem has been the main source of motivation for this study.

This study aims to use machine learning models to estimate the pathogenicity of variants of BRCA1 and BRCA2 genes. As a result of the study, it was planned to make the model trained with variants with known clinical status (pathogenicity) acceptable in the test data and to make it applicable on datasets with unknown clinical status. The data required for the models to be trained were obtained from the CADD database via the myvariant.info platform. Within the scope of the study, in addition to various basic classifiers such as KNN, SVC, Logistic Regression, advanced structures such as XGBoost, CatBoost, LightGBM are used and their outputs are reported.

The application was developed with the help of Kocaeli University Medical Genetic Academicians.

Key words: machine learning, brca1, brca2, breast cancer, bioinformatics, dna, gene, variant, pathogenicity

1. GİRİŞ

Bu çalışma, BRCA1 ve BRCA2 genlerine ait varyantların patojenite analizinin yapılması üzerine eğilmiştir. Tüm dünyada kadınlarda en sık görülen kanser türü olan meme kanserinin önlenmesinde etkin rol oynayan bu iki gen üzerinde oluşabilen varyantlar bu genlerin işlevlerini yerine getirememesi ve tümörleri baskılayamaması ile sonuçlanmakta, bu da bireyde kanseri doğurmaktadır. Burada bahsedilen varyantların nispeten küçük bir kısmının klinik statüsü bilinse de önemli bir kısmının klinik statüleri yani patojeniteleri ile ilgili bir bilgi bulunmamaktadır. Bu da bu varyanta sahip hücreler taşıyan bireylerin kanser riski saptamasını oldukça güçleştirmektedir. Bu çalışmada bu sorunu çözmek üzere makine öğrenmesi yöntemlerinden yararlanılması önerilmektedir. Aşağıda tezin bölümleri ile ilgili açıklama yer almaktadır.

- **Genel Bilgiler** bölümünde Meme Kanseri, DNA, Gen, BRCA genleri, Gen Dizileme teknolojisi, İnsan Genom Projesi, Varyantlar, Patojenite, Bu çalışmada kullanılan veri tabanları, Makine Öğrenmesi, Sınıflandırma Konsepti, Çalışmada kullanılan sınıflandırma algoritmaları gibi bu çalışma kapsamında sahip olunması gereken temel bilgilere ait teorik tanım ve açıklamalara yer verilir.
- **Malzemeler ve Yöntem** bölümünde gerekli verilerin elde edilmesi, ön işleme adımları, seçilen algoritmaların uygulamaları ile ilgili teknik detaylar ve alınan kararlar ile ilgili ayrıntılı açıklamalara ve istenilen performansı veremeyen ara çıktılarına yer verilir.
- **Çıktılar** bölümünde geliştirilen Makine Öğrenmesi modelleri ve kullanılan algoritmalara ait deneysel çıktılarına yer verilmektedir
- **Tartışma** bölümünde çalışma boyunca alınan kararlar ve oluşturulan ikiliklerin, karşılaşılan problemler ve bunlara getirilen çözümlerin deneysel çıktıları nasıl etkilediği, farklı tercihlerin nasıl sonuçlar doğurabileceği ve daha iyi çıktılarına nasıl ulaşılabileceği ile ilgili detaylar ortaya konmakta ve bizim düşüncelerimiz gerekçeleri ile paylaşılmaktadır.

- **Sonuç ve Öneriler** bölümünde elde edilen nihai çıktı ve sistemin performansının, öneminin ve potansiyelinin değerlendirmelerini içerir.

Bu çalışmada genler üzerindeki patojenitesi bilinmeyen varyantların patojenitelerini Makine Öğrenmesi yöntemleri ile tahmin etmek hedeflenmekte ve uygulama alanı olarak BRCA1 ve BRCA2 genleri seçilmektedir.

Tezin boyunca bahsi geçecek tüm yazılım, kod ve teknik dökümanlara çalışmaya ait GitHub hesabı üzerinden erişilebilmektedir [32].

2. GENEL BİLGİLER

2.1. Meme Kanseri

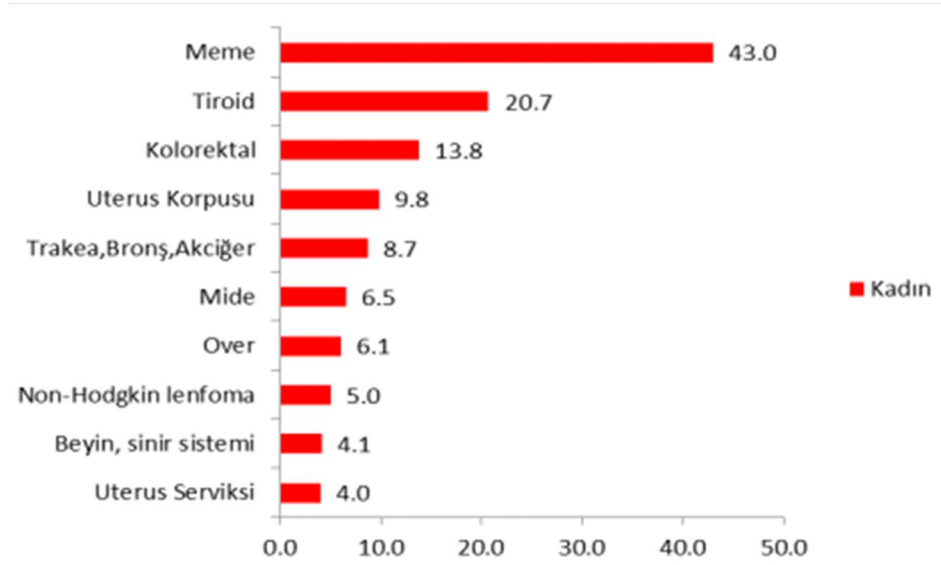
Meme kanseri, kadınlarda en sık görülen invaziv kanserdir ve akciğer kanserinden sonra kadınlarda kanser ölümünün önde gelen ikinci nedenidir. Meme kanseri taraması ve tedavisindeki ilerlemeler 1989'dan beri hayatta kalma oranlarını önemli ölçüde artırmıştır. ACS'e göre, Amerika Birleşik Devletleri'nde 3,1 milyondan fazla, meme kanseri hastası bulunmaktadır. [1]

Tablo 2.1.1 IARC tarafından yayınlanan Globocan 2012 verilerine göre kadınlarda en sık görülen ilk beş kanserlerin dağılımı. [4]

	Türkiye*	Dünya	IARC'a üye 24	AB (28 ülke)	ABD
1	Meme	Meme	Meme	Meme	Meme
2	Tiroid	Kolorektal	Kolorektal	Kolorektal	Akciğer
3	Kolorektal	Uterus serviksi	Akciğer	Akciğer	Kolorektal
4	Uterus korpusu	Akciğer	Uterus serviksi	Uterus korpusu	Tiroid
5	Akciğer	Uterus korpusu	Uterus korpusu	Uterus serviksi	Uterus

* Türkiye Birleşik Veri Tabanı, 2014

Herhangi bir kadının meme kanserinden ölme şansı 38'de 1'dir (%2,6). ACS, 268.600 kadının invaziv meme kanseri teşhisi alacağını ve 62.930 kişinin 2019 yılında noninvaziv kanser teşhisi alacağını tahmin ediyor. Aynı yıl ACS, 41.760 kadının meme kanseri nedeniyle öleceğini bildirdi [1]. Türkiye'de faaliyet gösteren MEMEDER'e göre ise Türkiye'de her yıl 25 bin yeni meme kanseri tanısı konuyor [2]. Bununla birlikte, tedavideki ilerlemeler nedeniyle, meme kanserinden ölüm oranları 1989'dan beri azalmaktadır. Semptomların farkında olma ve tarama ihtiyacı riski azaltmanın önemli yollarındandır. Nadir durumlarda, meme kanseri erkekleri de etkileyebilmektedir. [1]



Şekil 2.1-1 Kadınlarda En Sık Görülen 10 Kanserinin Yaşa Göre Standardize Edilmiş Hızları. (Dünya Standart Nüfusu, 100.000 Kişide) [4]

Meme kanseri risk faktörlerinde en üst sıralarda genetik faktör de yer alır. BRCA1 ve BRCA2 genlerinde belirli mutasyonlar taşıyan kadınların meme kanseri, yumurtalık kanseri veya her ikisini birden geliştirme şansı daha yüksektir. İnsanlar bu genleri ebeveynlerinden miras alırlar. TP53 genindeki mutasyonların ayrıca artan meme kanseri riski ile bağlantıları vardır. Yakın bir akrabası meme kanseri geçirmiş bir kişinin meme kanseri geliştirme şansı artar. Aile öyküsü öncelikle anne, kız ve kız kardeşten oluşan birinci derece akrabaları kapsar. Eğer yakın aile bireylerinden menopoz öncesi ve iki taraflı meme kanseri olan varsa, hayat boyu risk %50'dir. [2]

Bir doktor sıklıkla rutin tarama sonucunda veya hasta semptomları tespit ettikten sonra doktora yaklaştığında meme kanserini teşhis eder[1]. Oysa bu çalışmanın da eğildiği alan olan varyantların klinik önemini tespit edilmesi başarıya ulaşırsa DNA dizileme işleminden sonra varyantları tespit edilmiş insanlar için belirtiler ortaya çıkmadan çok önce hastalığa kalıtsal yatkınlık hakkında bulgular elde edilebilir.

2.2. DNA Nedir?

Bu çalışmaların anlaşılabilmesi için DNA ve Gen kavramlarının anlaşılması elzemdir. NLM DNA'yı şöyle tanımlar:

DNA veya deoksiribonükleik asit, insanlarda ve hemen hemen tüm diğer organizmalarda kalıtsal maddedir. Bir insanın vücudundaki hemen hemen her hücre aynı DNA'ya sahiptir. Çoğu DNA hücre çekirdeğinde, ancak mitokondride az miktarda DNA da bulunabilir. Mitokondri, hücreler içindeki enerjiyi gıdalardan hücrelerin kullanabileceği bir forma dönüştüren yapılardır.[6]

DNA'daki bilgiler dört kimyasal bazdan oluşan bir kod olarak saklanır: A, G, C ve T. İnsan DNA'sı yaklaşık 3 milyar bazdan oluşur ve bu bazların yüzde 99'undan fazlası tüm insanlarda aynıdır. Bu bazların sırası veya sırası, bir organizmanın oluşturulması ve sürdürülmesi için mevcut olan, alfabadeki harflerin kelimeler ve cümleler oluşturmak için belirli bir sırada görünme şekline benzer bilgileri belirler.[6]

DNA bazları, baz çiftleri olarak adlandırılan birimler oluşturmak için birbirleriyle, A ile T ve C ile G'yi birleştirir. Her baz ayrıca bir şeker molekülüne ve bir fosfat molekülüne bağlanır. Birlikte, bir baz, şeker ve fosfata bir nükleotit denir. Nükleotidler, çift sarmal adı verilen bir spiral oluşturan iki uzun iplik halinde düzenlenmiştir. Çift sarmalın yapısı bir merdiven gibidir, taban çiftleri merdivenin basamaklarını ve şeker ve fosfat molekülleri merdivenin dikey yanlarını oluşturur.[6]

DNA'nın önemli bir özelliği, çoğaltabilmesi veya kendi kopyalarını oluşturabilmesidir. Çift sarmaldaki her DNA ipliği, baz dizisini çoğaltmak için bir desen görevi görebilir. Hücreler bölündüğünde bu önemlidir, çünkü her yeni hücrenin eski hücrede bulunan DNA'nın tam bir kopyasına sahip olması gerekir.[6]

2.3. Gen Nedir?

NBCF ise geni ve BRCA genlerini şöyle tanımlar: Her kişinin DNA'sı, insan vücudunu oluşturmak ve işlevini sürdürmek için kullanılan kodu içerir. Genler, DNA'nın bireysel özellikleri kodlayan küçük bölümleridir. Örneğin, doğal olarak kıvrık saçlı bir kişinin saçının kıvrımına olmasına neden olan bir geni vardır.[6]

Miras alınan tüm özellikler genler aracılığıyla aktarılır. Her insanın her genin iki kopyası vardır: her ebeveyninden bir gen. Her ebeveyn, genlerinin tam yarısını her çocuğa aktardığından, ebeveynin genetik özelliklerinden herhangi birinin yavrularına geçme şansı% 50'dir.[6]

2.4. BRCA Nedir?

“BRCA” adı “BReast CAncer geni” nin kısaltmasıdır. BRCA1 ve BRCA2, bir kişinin meme kanseri geliştirme şansını etkilediği tespit edilen iki farklı genidir.[7]

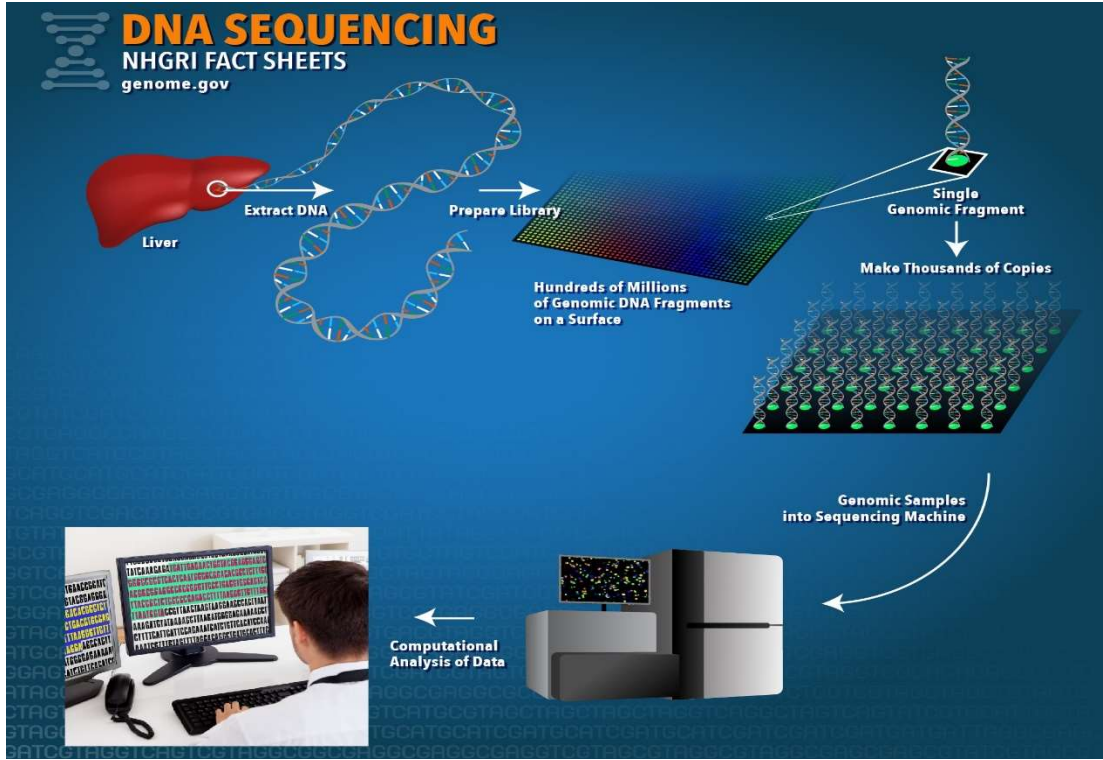
Her insanın hem BRCA1 hem de BRCA2 genleri vardır. İsimlerinin önerebilmesine rağmen, BRCA genleri meme kanserine neden olmaz. Aslında, bu genler normalde meme kanserini önlemede büyük rol oynamaktadır. Kansere ve tümörlerin kontrolsüz büyümesine yol açabilecek DNA kopmalarını onarmaya yardımcı olurlar. Bu nedenle, BRCA genleri tümör baskılayıcı genler olarak bilinir. .[7]

Bununla birlikte, bazı insanlarda bu tümör baskılayıcı genler düzgün çalışmaz. Bir gen değiştiğinde veya bozulduğunda, düzgün çalışmaz. Buna gen mutasyonu denir. .[7]

2.5. Gen Dizilime

Genetik hastalıklar üzerine çalışmaları oldukça hızlandıran ve başka disiplinlerden araştırmacıların bu alanlarda çalışmasını mümkün kılan yenilik gen dizileme olmuştur DNA dizileme olarak da ifade edilebilmektedir.

DNA dizilimi, DNA molekülünü oluşturan "bazlar" adı verilen dört kimyasal yapı bloğunun sırasını belirlemek anlamına gelir. Dizi, bilim insanlarına belirli bir DNA segmentinde taşınan genetik bilgi türünü anlatır. Örneğin, bilim adamları, DNA'nın hangi kısımlarının gen içerdiğini ve hangilerinin genleri açıp kapatarak düzenleyici talimatlar taşıdığını belirlemek için dizi bilgilerini kullanabilirler. Ek olarak ve daha da önemlisi, dizi verileri, bir gendeki hastalığa neden olabilecek değişiklikleri vurgulayabilir. DNA çift sarmalında, dört kimyasal baz her zaman aynı ortakla "baz çiftleri" oluşturmak için bağlanır. A her zaman T ile eşleşir; C her zaman G ile eşleşir. Bu eşleştirme, hücreler bölündüğünde DNA moleküllerinin kopyalandığı mekanizmanın temelidir ve eşleştirme ayrıca çoğu DNA sekanslama(dizileme) deneyinin gerçekleştirildiği yöntemlerin temelini oluşturur.[3]



Şekil 2.5-1 Gen dizileme süreçleri. [3]

Bu işlem oldukça ve maliyetli görünse de İnsan Genom Projesinin tamamlanmasından bu yana teknolojik gelişmeler ile otomasyon hızı artırmış ve maliyetleri ayrı ayrı genlerin rutin olarak dizilenebildiği noktaya kadar düşürmüştür. Hatta günümüzde bazı laboratuvarlar yılda 100.000 milyar bazdan fazla sekans oluşturabilmekte ve bütün bir genom sadece birkaç bin dolara dizilenebilmektedir.[3]

Her ne kadar doktor ofisinde rutin DNA sekanslaması için henüz biraz zaman gerekse de bazı büyük tıp merkezleri birtakım hastalıkları tespit etmek ve tedavi etmek için sekans kullanmaya başladı. Bunlardan birisi de bu çalışma boyunca desteklerini bizden esirgemeyen Kocaeli Üniversitesi Tıbbi Genetik Bölümü'dür. Buralardaki çalışmalar akademik çerçeveden çıkmış ve pratik amaçlarla kullanılmaktadır. Örneğin kanserde, doktorlar bir hastanın sahip olduğu belirli kanser türünü tanımlamak için sekans verilerini giderek daha fazla kullanabilmektedir. Bu, doktorun tedaviler için daha iyi seçimler yapmasını sağlar.

NHGRI destekli Teşhis Edilmemiş Hastalıklar Programındaki araştırmacılar, nadir görülen hastalıkların genetik nedenlerini tanımlamak için DNA dizilimi kullanırlar. Bir başka araştırma grubu, yeni doğanların hastalık ve hastalık riski açısından taranmasında kullanımını incelemektedirler. [3]

Dahası, NHGRI ve Ulusal Kanser Enstitüsü tarafından desteklenen Kanser Genom Atlas projesi, yaklaşık 30 kanser türünün genomik detaylarını ortaya çıkarmak için DNA dizilimini kullanmaktadır. Başka bir Ulusal Sağlık Enstitüleri programı, gen aktivitesinin farklı dokularda nasıl kontrol edildiğini ve hastalıkta gen regülasyonunun rolünü inceler. Devam eden ve planlanan büyük ölçekli projeler, kalp hastalığı ve diyabet gibi yaygın ve karmaşık hastalıkların ve fiziksel malformasyonlara, gelişimsel gecikmeye ve metabolik hastalıklara neden olan kalıtsal hastalıkların gelişimini incelemek için DNA dizilimini kullanır. [3]

2.6. İnsan Genom Projesi

Basitçe insan DNA'sındaki genlerin çıkarılması ve insan DNA'sının haritalanması olarak özetlenebilecek projenin NHGRI tarafından tanımı şöyle yapılır: "HGP, insan DNA'sını oluşturan baz çiftlerini belirlemek ve insan genomunun tüm genlerini hem fiziksel hem de işlevsel bir bakış açısıyla tanımlamak ve haritalamak amacıyla uluslararası bir bilimsel araştırma projesiydi.". İnsan referans genomu (referans düzeneği olarak da bilinir), bir türün idealize edilmiş tek bir organizmasında gen kümesinin temsili bir örneği olarak bilim adamları tarafından bir araya getirilen bir dijital nükleik asit sekansı veri tabanıdır. Bu insan genomunun ideal veya sağlıklı referansını belirleyerek karşılaştırma yapmayı mümkün kılar. Ancak burada tek bir referans oluşturma'nın genomun aşırı karmaşık yapısı sebebiyle henüz mümkün olmadığına karar verilmiştir. GRC'ü da İnsan Genom'u için referans genomları şu açıklama ile birlikte sunmaktadır. "GRC, insan için mümkün olan en iyi referans düzeneğini sağlamak için çok çalışıyor. Bunu, tek bir yolla gösterilemeyecek kadar karmaşık bölgeler için birden fazla temsil (alternatif lokus) oluşturarak yaparız. Ayrıca, yama olarak bilinen bölgesel düzeltmeleri de yayınlıyoruz. Bu, belirli bir konumla ilgilenen kullanıcıların, kromozom koordinat stabilitesine ihtiyaç duyan kullanıcıları etkilemeden gelişmiş bir temsil elde etmelerini sağlar." [5]

2.7. Varyant Nedir?

Tüm DNA'ların% 99.5'i tüm insanlarda paylaşılır; tüm farkı yaratan% 0.5'tir. Genetik varyasyonlar veya varyantlar, her bir kişinin genomunu benzersiz kılan farklardır. DNA sekanslaması, bir bireyin DNA sekansını GRC tarafından korunan bir referans genomun DNA sekansı ile karşılaştırarak bir bireyin varyantlarını tanımlar. [8]

Bir kişinin DNA sekansının referans DNA sekansından farklı olabileceği farklı yollar vardır, bazıları: Tek nükleotid polimorfizmleri ("SNP'ler"), tek bir nükleotid referans DNA sekansından farklı olduğunda ortaya çıkan DNA sekansı varyasyonlarıdır.

- Eklemeler, referans sekansa göre bir DNA sekansına ilave nükleotitlerin sokulmasıdır.
- Delesyonlar, referans sekansa göre eksik nükleotitler olduğundadır.
- İkameler, çoklu nükleotitlerin referans sekanstan değiştirildiği zamandır.

Yapısal varyantlar, bir kromozomun büyük bölümlerinin veya hatta tüm kromozomların bir şekilde çoğaltıldığı, silindiği veya yeniden düzenlendiği değişikliklerdir. [8]

Ortalama bir insanın genomunda milyonlarca varyant vardır. Bazı varyantlar genlerde görülür, ancak çoğu genlerin dışındaki DNA dizilerinde görülür. Az sayıda varyant hastalıklarla ilişkilendirilmiştir, ancak çoğu varyantın bilinmeyen etkileri vardır. Bazı varyantlar, farklı göz renkleri ve kan türleri gibi insanlar arasındaki farklılıklara katkıda bulunur. Araştırma topluluğu için daha fazla DNA dizisi bilgisi elde edildikçe, bazı varyantların etkileri daha iyi anlaşılabilir. [8]

2.8. BRCA Varyantları

İnsanların küçük bir yüzdesi (yaklaşık 400'de bir veya nüfusun% 0.25'i) mutasyona uğramış BRCA1 veya BRCA2 genleri taşır. Bir BRCA mutasyonu, geni oluşturan DNA bir şekilde hasar gördüğünde meydana gelir.

Bir BRCA geni mutasyona uğradığında, artık kırık DNA'nın onarılmasında ve meme kanserinin önlenmesinde etkili olmayabilir. Bu nedenle, BRCA gen mutasyonuna sahip kişilerin meme kanseri geliştirme olasılığı daha yüksektir ve daha genç yaşta

kanser geliştirme olasılığı daha yüksektir. Mutasyon geçirmiş genin taşıyıcısı bir gen mutasyonunu yavrularına da geçirebilir. [7]

2.9. Patojenite Nedir?

ACMG, DNA dizisi varyantlarının sınıflandırılması için yönergeler geliştirmiştir: [8]

- **Patojenik (Pathogenic):** daha önce bildirilen ve bozukluğun tanınmış bir nedeni olan bir dizi varyantı.
- **Muhtemelen Patojenik (Likely Pathogenic):** daha önce bildirilmemiş ve bozukluğa neden olması beklenen tipte bir dizi varyantı.
- **Önemi Bilinmeyen Varyantı (VUS):** daha önce bildirilmemiş ve bozukluğa neden olabilecek veya olmayabilecek tipte bir dizi varyantı.
- **Muhtemelen İyi Huylu (Likely Benign):** daha önce bildirilmemiş ve muhtemelen hastalığa neden olmayan bir dizi varyantı.
- **İyi Huylu (Benign):** bir sekans varyantı daha önce bildirilmiştir ve tanınmış bir nötr varyanttır.

Bu çalışmaların sonucunda günümüzde idealize edilmiş İnsan Referans Genom'unu kişinin kendi özel dizilimi ile karşılaştırma imkanı doğmuştur. Bu da herhangi bir insanın DNA'sındaki referans dizilimden farklı olan parçaları yani varyantları saptamayı sağlamaktadır. Bu çalışma ise BRCA1 ve BRCA2 geninde raporlanmış varyantlarınını inceleyerek patojenite sınıfı VUS olan varyantların klinik statülerinin saptanması amacıyla kullanılabilecek bir Makine Öğrenmesi Modeli Geliştirmektedir. Geliştirilecek modelin işleyeceği veriler CADD veri tabanından myvariant.info platformu aracılığıyla edinilmiştir.

2.10. CADD

CADD[31], tekli nükleotid varyantlarının(single nucleotide variants) zararlılığının yanı sıra insan genomundaki yerleştirme / silme (insertion/deletions) varyantlarının skorlanması için bir araçtır. [9]

Birçok varyant ek açıklama ve puanlama aracı etrafta olsa da, ek açıklamaların çoğu tek bir bilgi türünden (ör. Koruma, conservation) yararlanma eğilimindedir ve / veya

kapsamla sınırlıdır (örneğin, missense changes). Bu nedenle, farklı bilgileri objektif olarak ağırlıklandıran ve entegre eden geniş çapta uygulanabilir bir metriğe ihtiyaç vardır. CADD, doğal seçimden kurtulan varyasyonları simüle edilmiş mutasyonlarla karşılaştırarak birden fazla ek açıklamayı bir metriğe entegre eden bir frameworktur. [9]

2.11. Myvariant.info

MyVariant.info, birçok popüler veri kaynağından toplanan varyant anotasyon verilerini sorgulamak / almak için kullanımı kolay REST web hizmetleri sağlar. Varyantlar çeşitli niteliklerine göre sorgulanarak indirilebilir. [10]

2.12. Makine Öğrenmesi Nedir

Makine öğrenimi, sistemlere açıkça programlanmadan deneyimden otomatik olarak öğrenme ve geliştirme yeteneği sağlayan yapay zekanın (AI) bir uygulamasıdır. Makine öğrenimi, verilere erişebilen ve öğrenmeyi kendileri için kullanabilen bilgisayar programlarının geliştirilmesine odaklanır.

Öğrenme süreci, verdiğimiz örneklerle dayanarak verilerdeki kalıpları aramak ve gelecekte daha iyi kararlar vermek için örnekler, doğrudan deneyim veya talimatlar gibi gözlemler veya verilerle başlar. Birincil amaç, bilgisayarların insan müdahalesi veya yardımı olmadan otomatik olarak öğrenmelerini sağlamak ve eylemleri buna göre ayarlamaktır.

Ancak, klasik makine öğrenimi algoritmalarını kullanarak, metin bir anahtar kelimeler dizisi olarak kabul edilir; bunun yerine, anlamsal analize dayanan bir yaklaşım, insanın bir metnin anlamını anlama yeteneğini taklit eder.[11]

Makine öğrenimi algoritmaları genellikle denetlenen veya denetlenmeyen olarak kategorize edilir.

Denetimli makine öğrenme algoritmaları : Denetimli öğrenme, daha sonra yeni örnekler için genelleştirebileceğiniz bir işlevi eğitmek için etiketli eğitim verilerini kullanabileceğiniz bir yöntemdir. Eğitim, işlevin ne zaman doğru olup olmadığını belirleyebilecek bir eleştirmen içerir ve ardından doğru sonucu üretmek için işlevi

değiştirir. Klasik örnekler, geri yayılma algoritması tarafından eğitilmiş sinir ağlarını içerir, ancak diğer birçok algoritma vardır. Bu eğitici, öğrenme uygulamalarında SVM'ler ve olasılık sınıflandırıcıları (Naive Bayes) gibi diğer yaklaşımları araştırmaktadır.

Denetimli öğrenmede, girdi verileri ve istenen çıktıdan oluşan etiketli eğitim verilerini kullanarak bir işlev (veya model) oluşturursunuz. Denetim, istenen çıktı biçiminde gelir; bu da işlevi, ürettiği gerçek çıktıya göre ayarlamanıza olanak tanır. Eğitildiğinde, ideal olarak doğru tepki veren bir çıktı (tahmin veya sınıflandırma) üretmek için bu işlevi yeni gözlemlere uygulayabilirsiniz.

Yarı denetimli makine öğrenme algoritmaları: Denetimli ve denetimsiz öğrenme arasında bir yere düşer, çünkü eğitim için hem etiketli hem de etiketsiz verileri kullanırlar tipik olarak az miktarda etiketlenmiş veri ve büyük miktarda etiketlenmemiş veri. Bu yöntemi kullanan sistemler öğrenme doğruluğunu önemli ölçüde geliştirebilir. Genellikle, yarı denetimli öğrenme, elde edilen etiketli veriler, onu eğitmek / ondan öğrenmek için yetenekli ve ilgili kaynaklar gerektirdiğinde seçilir. Aksi takdirde, etiketlenmemiş verilerin edinilmesi genellikle ek kaynak gerektirmez.

Denetimsiz makine öğrenme algoritmaları: Denetimsiz öğrenmede, bir algoritma, verilerdeki bazı gizli özelliklere dayanarak verilerin etiketinin kaldırıldığı bir veri kümesindeki verileri ayırır. Bu işlev, verinin gizli yapısını keşfetmek ve anomali tespiti gibi görevler için yararlı olabilir. Bu eğitici, denetimsiz öğrenmenin arkasındaki fikirleri ve uygulamalarını açıklar ve daha sonra bu fikirleri veri keşfi bağlamında gösterir.

Denetimsiz öğrenme algoritmaları, etiketlenmemiş bir veri kümesindeki verileri, verilerdeki temel gizli özelliklere göre gruplandırır. Etiket olmadığı için, sonucu değerlendirmenin bir yolu yoktur (denetimli öğrenme algoritmalarının önemli bir farkı). Verileri gözetimsiz öğrenme yoluyla gruplandırarak, aksi halde görülemeyen ham veriler hakkında bir şeyler öğrenirsiniz. Çok boyutlu veya büyük veri setlerinde bu sorun daha da belirgindir.

Takviye makinesi öğrenme algoritmaları: Takviye öğrenme ilginç bir öğrenme modelidir, sadece bir girdinin bir çıktıya nasıl haritalanacağını öğrenmekle kalmaz, aynı zamanda bir dizi girdiyi bağımlılıklara sahip çıktılarla eşleştirebilir (örneğin Markov karar süreçleri). Takviye öğrenimi bir ortamdaki durumlar ve belirli bir durumda yapılabilecek eylemler bağlamında mevcuttur. Öğrenme işlemi sırasında, algoritma bazı ortamlardaki durum-eylem çiftlerini rastgele (bir durum-eylem çifti tablosu oluşturmak için) araştırır, daha sonra öğrenilen bilgilerin pratiğinde durumu kullanır. İşlem çifti, belirli bir durum için bazı hedef durumlara yol açan en iyi işlemi seçmeyi ödüllendirir.[12]

2.13. Sınıflandırma nedir

Makine öğrenimi ve istatistiklerinde sınıflandırma, bilgisayar programının kendisine verilen verilerden öğrendiği ve yeni gözlemler veya sınıflandırmalar yaptığı denetimli bir öğrenme yaklaşımıdır.

Sınıflandırma, belirli bir veri kümesini sınıflara ayırma işlemidir, hem yapılandırılmış hem de yapılandırılmamış veriler üzerinde gerçekleştirilebilir. Süreç, verilen veri noktalarının sınıfını tahmin etmekle başlar. Sınıflara genellikle hedef, etiket veya kategoriler denir.

Sınıflandırma öngörülü modelleme, giriş değişkenlerinden ayrık çıkış değişkenlerine eşleme fonksiyonunun yakınlştırılması görevidir. Ana hedef, yeni verilerin hangi sınıfa / kategoriye gireceğini belirlemektir.

Kalp hastalığı tespiti bir sınıflandırma problemi olarak tanımlanabilir, bu sadece iki sınıf olabileceğinden, yani kalp hastalığı veya kalp hastalığı olmadığından ikili bir sınıflandırmadır. Bu durumda sınıflandırıcı, verilen girdi değişkenlerinin sınıfla nasıl ilişkili olduğunu anlamak için eğitim verilerine ihtiyaç duyar. Sınıflandırıcı doğru bir şekilde eğitildikten sonra, kalp hastalığının belirli bir hasta için olup olmadığını tespit etmek için kullanılabilir.

Sınıflandırma bir tür denetimli öğrenme olduğundan, hedeflere bile girdi verileri sağlanır.

2.13.1. Makine öğreniminde sınıflandırma terminolojileri

- Classifier: Giriş verilerini belirli bir kategoriye eşlemek için kullanılan bir algoritmadır.
- Classification Model: Model, eğitim için verilen giriş verilerini tahmin eder veya bir sonuç çıkarır, veriler için sınıfı veya kategoriye tahmin eder.
- Feature - Özellik, gözlenen olayın bireysel olarak ölçülebilir bir özelliğidir.
- Binary Classification - İki sonucu olan bir sınıflandırma türüdür, örneğin - doğru veya yanlış.
- Multi-Class Classification: İki'den fazla sınıf içeren sınıflandırma, çok sınıflı sınıflandırmada her örnek bir ve yalnızca bir etikete veya hedefe atanır.
- Multi-label Classification: Bu, her örneğin bir etiket veya hedef kümesine atandığı bir sınıflandırma türüdür.
- Initialize: Bu öge için kullanılacak sınıflandırıcıyı atamaktır
- Train the Classifier: Verinin eğitilmesidir.
- Predict the Target: Eğitilen veri kullanılarak, etiketlenmemiş verileri etiketlemek için yapılan tahmindir.
- Evaluate: Bu temel olarak modelin değerlendirilmesi, yani sınıflandırma raporu, doğruluk puanı vb. anlamına gelir.

2.13.2. Sınıflandırmada öğrenen türleri

1. Lazy Learners: Tembel öğrenciler sadece eğitim verilerini saklar ve bir test verisi görünene kadar bekler. Sınıflandırma, saklanan eğitim verilerindeki en ilgili veriler kullanılarak yapılır. İstekli öğrencilere kıyasla daha fazla tahmin süresi var. Örneğin - k-en yakın komşu, vaka temelli akıl yürütme.
2. Eager Learners: İstekli öğrenciler tahminler için veri almadan önce verilen eğitim verilerine dayalı bir sınıflandırma modeli oluşturur. Tüm alan için işe yarayacak tek bir hipotez taahhüt edebilmelidir. Bu nedenle, eğitimde çok zaman alırlar ve bir tahmin için daha az zaman alırlar. Örneğin - Karar Ağacı, Naif Bayes, Yapay Sinir Ağları.[13]

2.14. Gradient Boosting

Gradient boosting, sınıflandırma veya regresyon tahmini modelleme problemleri için kullanılabilen bir grup makine öğrenme algoritmaları sınıfını ifade eder. Gradient boosting, gradient tree boosting, stochastic gradient boosting (bir uzantı) ve gradient boosting machines veya kısaca GBM olarak da bilinir. Topluluklar karar ağacı modellerinden yapılır. Ağaçlar topluluğa her seferinde bir tane eklenir ve önceki modeller tarafından yapılan tahmin hatalarını düzeltmek için uygundur. Bu, destekleme olarak adlandırılan bir tür topluluk makine öğrenme modelidir.

Gradient boosting etkili bir makine öğrenme algoritmasıdır ve genellikle tablo ve benzer yapılandırılmış veri kümelerinde makine öğrenme yarışmalarını (Kaggle gibi) kazanmak için kullanılan ana veya ana algoritmalarından biridir. Algoritma, belirli bir veri kümesi için ayarlanması ve belki de ayarlanması gereken hiperparametreler sağlar. Ayarlanacak birçok hiperparametre olmasına rağmen, belki de en önemlileri şunlardır:

- Modeldeki ağaç veya tahminci sayısı.
- Modelin öğrenme oranı.
- Stokastik modeller için satır ve sütun örnekleme hızı.
- Maksimum ağaç derinliği.
- Minimum ağaç ağırlığı.
- Düzenleme terimleri alfa ve lambda.

Python'da gradyan artırma algoritmasının birçok uygulaması vardır. Belki de en çok kullanılan uygulama scikit-learn kütüphanesi ile sağlanan sürümdür.

Uygulamada genellikle daha iyi sonuçlar elde eden algoritmanın hesaplamalı olarak verimli alternatif uygulamalarını sağlayan ek üçüncü taraf kütüphaneleri mevcuttur. Örnekler arasında XGBoost kütüphanesi, LightGBM kütüphanesi ve CatBoost kütüphanesi sayılabilir.[14]

Herhangi bir denetimli öğrenme algoritmasının amacı bir kayıp fonksiyonunu tanımlamak ve bunu en aza indirmektir. Gradient Boosting algoritması için matematiğin nasıl çalıştığını görelim. Diyelim ki kayıp olarak MSE var:

$$MSE = \sum (y_i - y_i^p)^2 \quad (2.14.1)$$

Tahminlerimizi istiyoruz, böylece kayıp fonksiyonumuz MSE minimumdur. Gradient descent kullanarak ve tahminlerimizi bir öğrenme oranına göre güncelleyerek MSE'nin minimum olduğu değerleri bulabiliriz.

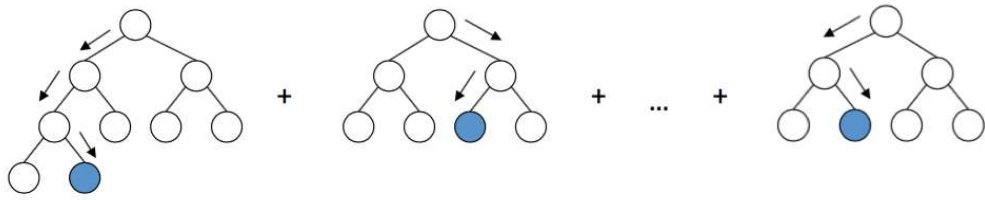
$$y_i^p = y_i^p + \alpha * \delta \sum (y_i - y_i^p)^2 / \delta y_i^p \quad (2.14.2)$$

Genel Denklemi burada:

$$y_i^p = y_i^p - \alpha * 2 * \sum (y_i - y_i^p) \quad (2.14.3)$$

Olur. Burada α öğrenme oranı ve $\sum (y_i - y_i^p)$ artıkların toplamıdır

Dolayısıyla, tahminlerimizi, artıklarımızın toplamı 0'a (veya minimum) yakın ve öngörülen değerler gerçek değerlere yeterince yakın olacak şekilde güncelliyoruz.[15]



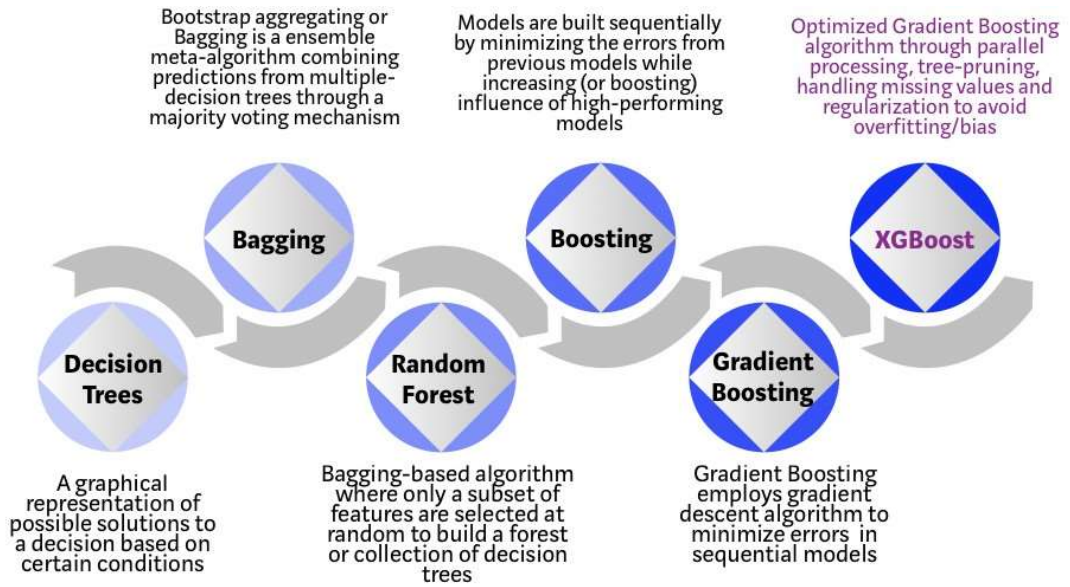
Şekil 2.14-1 Gradient Boosting ağaç şeması.[15]

2.15. XGBOOST

XGBoost, yüksek verimli, esnek ve taşınabilir olacak şekilde tasarlanmış, optimize edilmiş dağıtılmış bir gradient boosting kütüphanesidir. Gradient Boosting çerçevesi

altında makine öğrenme algoritmaları uygular. XGBoost, birçok veri bilimi problemini hızlı ve doğru bir şekilde çözen paralel bir ağaç güçlendirmesi (GBDT, GBM olarak da bilinir) sağlar. Aynı kod, büyük dağıtılmış ortamda (Hadoop, SGE, MPI) çalışır ve milyarlarca örneğin ötesinde sorunları çözebilir.[117]

eXtreme Gradient Boosting (XGBoost), etkinlik, hesaplama hızı ve model performansı için tasarlanmış gradient boosting algoritmasının (terminoloji uyarısı) ölçeklenebilir ve geliştirilmiş bir sürümüdür. Açık kaynaklı bir kütüphane ve Dağıtılmış Makine Öğrenimi Topluluğunun bir parçasıdır. XGBoost, mevcut boosting tekniklerini en kısa sürede hassas bir şekilde geliştirmek için tasarlanmış yazılım ve donanım yeteneklerinin mükemmel bir karışımıdır.[17]



Şekil 2.15-1 Ensemble tekniklerin geçmişi.[17]

Regresyon, sınıflandırma, sıralama ve kullanıcı tanımlı tahmin problemlerini çözmek için kullanılabilir.

- Windows, Linux ve OS X üzerinde sorunsuz çalışır.
- C ++, Python, R, Java, Scala ve Julia dahil tüm önemli programlama dillerini destekler.
- AWS, Azure ve İplik kümelerini destekler ve Flink, Spark ve diğer ekosistemlerle iyi çalışır. [16]

Boosting ve gradient boosting arasındaki en büyük fark, her iki algoritmanın da modeli (zayıf öğrenenler) yanlış tahminlerden nasıl güncellediği. Gradyan yükseltme, ağırlıkları güncelleyerek modelin kaybını tekrarlayan şekilde optimize eden Gradient Descent adlı bir algoritma kullanarak gradient (kayıp fonksiyonunda bir yön) kullanarak ağırlıkları ayarlar.[17]

XGBoost ve GBM'ler, gradient descent mimarisini kullanarak zayıf öğrenenleri (genel olarak CART'lar) artırma prensibini uygulayan topluluk ağacı yöntemleridir. Bununla birlikte, XGBoost, sistem optimizasyonu ve algoritmik geliştirmeler yoluyla temel GBM çerçevesini geliştirir.

XGBoost'un başarısının arkasındaki en önemli faktör, tüm senaryolarda ölçeklenebilirliğidir. Sistem, tek bir makinedeki mevcut popüler çözümlerden on kat daha hızlı çalışır ve dağıtılmış veya bellek sınırlı ayarlarda milyarlarca örneğe ölçeklendirilir. XGBoost'un ölçeklenebilirliği birkaç önemli sistemden ve algoritmik optimizasyondan kaynaklanmaktadır. Bu yenilikler şunları içerir: seyrek verileri işlemek için yeni bir ağaç öğrenme algoritması; teorik olarak doğrulanmış weighted quantile sketch prosedürü yaklaşık ağaç öğrenmesinde örnek ağırlıklarının kullanılmasını sağlar. Paralel ve dağıtılmış bilgi işlem, öğrenmeyi daha hızlı hale getirir ve bu da model araştırmasını hızlandırır. [18]

2.15.1. Paralleleştirme

XGBoost, paralelleştirilmiş uygulama kullanarak sıralı ağaç oluşturma sürecine yaklaşır. Bu, temel öğrenenler oluşturmak için kullanılan döngülerin değiştirilebilir doğası nedeniyle mümkündür; bir ağacın yaprak düğümlerini numaralandıran dış halka ve özellikleri hesaplayan ikinci iç halka. Döngülerin bu şekilde iç içe yerleştirilmesi paralelleştirmeyi sınırlar, çünkü iç döngüyü tamamlamadan (ikisinden daha hesaplama gerektiren), dış döngü başlatılamaz. Bu nedenle, çalışma süresini iyileştirmek için, döngülerin sırası, tüm örneklerin küresel bir taramasıyla başlatma ve paralel iş parçacıkları kullanılarak sıralama kullanılarak değiştirilir. Bu anahtar, hesaplamadaki herhangi bir paralelleştirme ek yükünü dengeleyerek algoritmik performansı artırır.

2.15.2. Ağaç budaması

GBM çerçevesi içinde ağaç ayrılması için durma kriteri doğada açgözlüdür ve bölünme noktasındaki negatif kayıp kriterine bağlıdır. XGBoost, önce ölçüt yerine 'max_depth' parametresini kullanır ve ağaçları geriye doğru budamaya başlar. Bu "önce derinlik" yaklaşımı, hesaplama performansını önemli ölçüde artırır.

Donanım Optimizasyonu: Bu algoritma, donanım kaynaklarını verimli kullanmak için tasarlanmıştır. Bu, degrade istatistiklerini saklamak için her iş parçacığındaki dahili arabellekleri ayırarak önbellek farkındalığıyla gerçekleştirilir. 'Çekirdek dışı' bilgi işlem gibi diğer geliştirmeler, belleğe sığmayan büyük veri çerçevelerini işlerken kullanılabilir disk alanını optimize eder.[16]

2.15.3. XGBoost'un avantajları

- Tahmin oluşturmada çok hızlı ve çoğu öğrenme algoritmasına kıyasla eğitilmesi nispeten hızlıdır.
- Özellikle, - bir dereceye kadar - paralelleştirilebilir, daha büyük modellerin birden fazla işlemci arasında eğitimine izin verir.
- Birçok algoritmanın mücadele ettiği seyrek verileri işleyebilir.
- Çok çeşitli koşullarda gözle görülür derecede iyi sonuç verir.
- Bir uygulama açık kaynak olarak mevcuttur ve hem R hem de Python tarafından iyi bir şekilde desteklenir.

2.15.4. XGBoost'un dezavantajları

- Bu bir kara kutu: herhangi bir tahminin arkasındaki mantığı belirlemek nispeten zordur (kutuyu açmaya yönelik yöntemler mevcut olsa da).
- Özellikle, tahminler süresizdir - hem açıklayıcı hem de hedef değişkenler sürekli olsa bile, bölme noktalarında bir değerden diğerine atlarlar.
- Microsoft'un LightGBM ve CatBoost gibi yeni rakiplerinden biraz daha yavaş.[19]

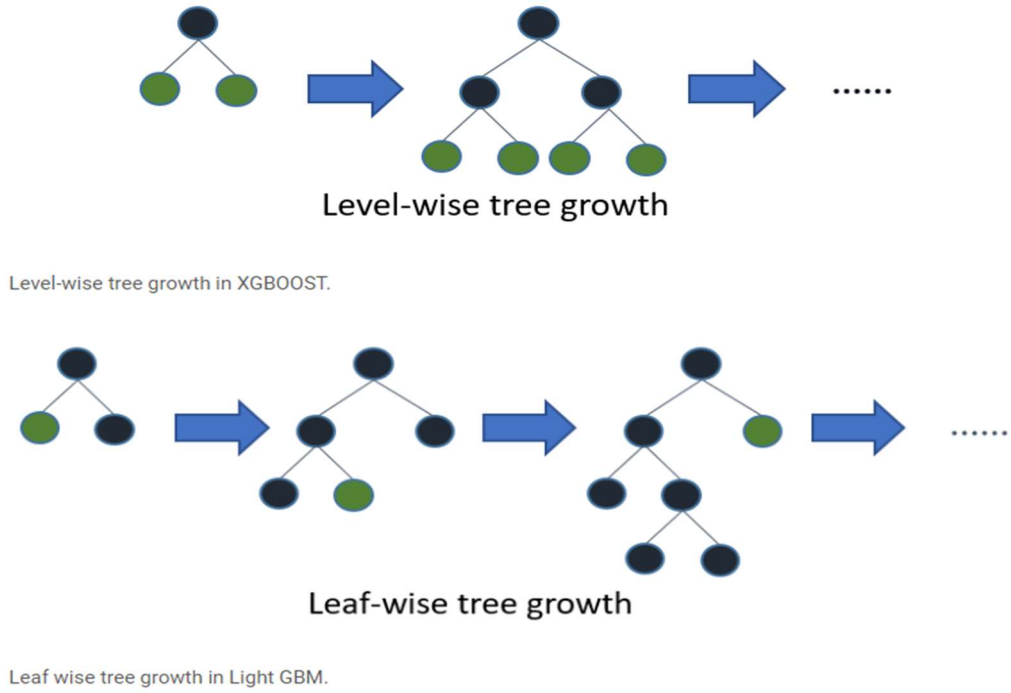
2.16. LightGBM

GBDT, verimliliği, doğruluğu ve yorumlanabilirliği nedeniyle yaygın olarak kullanılan bir makine öğrenme algoritmasıdır. GBDT, çok sınıflı sınıflandırma, tıklama tahmini ve sıralamayı öğrenme gibi birçok makine öğrenimi görevinde son teknoloji performanslara ulaşır. Son yıllarda, büyük verilerin ortaya çıkmasıyla (hem özellik sayısı hem de örnek sayısı açısından), GBDT özellikle doğruluk ve verimlilik arasındaki dengede yeni zorluklarla karşı karşıyadır. GBDT'nin geleneksel uygulamalarının, her özellik için, olası tüm ayrılma noktalarının bilgi kazanımını tahmin etmek için tüm veri örneklerini taraması gerekir. Bu nedenle, hesaplama karmaşıklıkları hem özellik sayısı hem de örnek sayısı ile orantılı olacaktır. Bu, büyük verileri işlerken bu uygulamaları çok zaman alıcı hale getirir.[20]

Verilerin boyutu gün geçtikçe artmaktadır ve geleneksel veri bilimi algoritmalarının daha hızlı sonuç vermesi zorlaşmaktadır. LightGBM, yüksek hızı nedeniyle "Light" olarak eklenir. LightGBM büyük boyutlu verileri işleyebilir ve çalışması için daha az bellek alır. LightGBM'nin popüler olmasının bir başka nedeni, sonuçların doğruluğuna odaklanmasıdır. LGBM ayrıca GPU öğrenimini de destekler ve bu nedenle veri bilimcileri veri bilimi uygulama geliştirme için LGBM'yi yaygın olarak kullanırlar.[21]

LightGBM, sıralama, sınıflandırma ve diğer birçok makine öğrenimi görevi için kullanılan karar ağacı algoritmasına dayanan hızlı, dağıtılmış, yüksek performanslı bir gradient boosting çerçevedir.[22]

LightGBM, dikey olarak ağacı büyütür, diğer algoritma ise ağaçları yatay olarak büyütür; diğer bir deyişle, lightGBM leaf-wise olarak büyür, diğer algoritmalar ise level-wise olarak büyür. Büyümek için maksimum delta kaybı olan yaprağı seçecektir. Aynı yaprağı büyütürken, leaf-wise algoritması, level-wise bir algorithmadan daha fazla kaybı azaltabilir.[21]



Şekil 2.16-1 LightGBM ağaç şeması.[21]

Leaf-wise bölünmeler karmaşıklıkta artışa neden olur ve overfitting sebebiyet verebilir. Fakat bölünmenin meydana geleceği derinliği belirten 'max_depth' parametresi belirlenerek üstesinden gelinebilir.

2.16.1. Light GBM'nin avantajları

- Daha hızlı egzersiz hızı ve daha yüksek verimlilik: LightGBM, histogram tabanlı algoritma kullanır, yani sürekli özellik değerlerini, eğitim prosedürünü sabitleyen ayrı kutulara koyar.
- Daha az bellek kullanımı: Sürekli değerleri daha düşük bellek kullanımına neden olan ayrık kutulara değiştirir.
- Diğer herhangi bir yükseltme algoritmasından daha iyi doğruluk: Daha yüksek doğruluk elde etmenin ana faktörü olan seviye bilge bir yaklaşım yerine yaprak bilge bölünmüş yaklaşımı takip ederek çok daha karmaşık ağaçlar üretir. Ancak, bazen max_depth parametresi ayarlanarak önlenebilecek aşırı sığmaya yol açabilir.
- Büyük Veri Kümeleriyle Uyumluluk: XGBOOST ile karşılaştırıldığında eğitim süresinde önemli bir azalma ile büyük veri kümelerinde eşit derecede iyi performans gösterebilir.[22]

2.17. CatBoost

“Boost”, bu kitaplık gradient boosting kitaplığı dayandığından, gradient boosting makine öğrenme algoritmasından gelir. Gradient boosting, sahtekarlık tespiti, öneri kalemleri, tahmin gibi çeşitli iş zorluklarına yaygın olarak uygulanan güçlü bir makine öğrenme algoritmasıdır ve aynı zamanda iyi performans gösterir. Ayrıca, büyük miktarda veriyi öğrenmesi gereken DL modellerinin aksine, nispeten daha az veriyle çok iyi sonuç döndürebilir. [23]

CatBoost, karar ağaçlarında gradient boosting için bir algoritmadır. Yandex araştırmacıları ve mühendisleri tarafından geliştirilen, şirket içinde görevleri sıralamak, tahmin etmek ve önerilerde bulunmak için yaygın olarak kullanılan MatrixNet algoritmasının halefidir. Evrenseldir ve çok çeşitli alanlarda ve çeşitli problemlere uygulanabilir.

Catboost iki kritik algoritmik ilerleme sunar - ordered boosting uygulanması, klasik algoritmaya permütasyon odaklı bir alternatif ve kategorik özellikleri işlemek için yenilikçi bir algoritma. Her iki teknik de, mevcut gradient boosting algoritmalarının tüm uygulamalarında bulunan özel bir tür hedef sızıntısının neden olduğu tahmin kaymasıyla mücadele etmek için eğitim örneklerinin rastgele permütasyonlarını kullanmaktadır.

Her kategorik özelliği, kategorinin şart koştuğu beklenen hedefin tahmini ile kodladığımız basit ama etkili bir yaklaşımdır. Bu kodlamanın dikkatsizce uygulanmasının (aynı kategorideki eğitim örnekleri üzerinde ortalama y değerinin) hedef sızıntıya neden olduğu ortaya çıkıyor.

Bu tahminle mücadele etmek için CatBoost daha etkili bir strateji kullanıyor. Sipariş prensibine dayanır ve eğitim örneklerini zamanında sırayla alan çevrimiçi öğrenme algoritmalarından ilham alır. Bu ortamda, her örnek için TS değerleri sadece gözlenen geçmişe dayanır. Bu fikri standart bir çevrimdışı ortama uyarlamak için Catboost yapay bir “zaman” sunar - eğitim örneklerinin rastgele permütasyonu σ . Daha sonra, her örnek için, Hedef İstatistiklerini hesaplamak için mevcut tüm "geçmiş" kullanır. Yalnızca bir rasgele permütasyon kullanıldığında, Hedef İstatistik'te sonraki örneklerden daha yüksek varyansa sahip önceki örneklerle sonuçlandığına dikkat edin.

Bu amaçla, CatBoost gradient boosting farklı adımları için farklı permütasyonlar kullanır.[24]

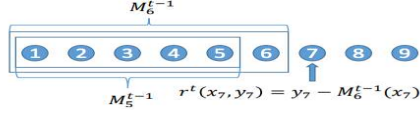


Figure 1: Ordered boosting principle.

Algorithm 1: Ordered boosting

input : $\{(\mathbf{x}_k, y_k)\}_{k=1}^n, I;$
 $\sigma \leftarrow \text{random permutation of } [1, n];$
 $M_i \leftarrow 0$ for $i = 1..n;$
for $t \leftarrow 1$ **to** I **do**
 for $i \leftarrow 1$ **to** n **do**
 $r_i \leftarrow y_i - M_{\sigma(i)-1}(i);$
 for $i \leftarrow 1$ **to** n **do**
 $\Delta M \leftarrow \text{LearnModel}((\mathbf{x}_j, r_j) : \sigma(j) \leq i);$
 $M_i \leftarrow M_i + \Delta M;$
return M_n

Algorithm 2: Building a tree in CatBoost

input : $M, \{y_i\}_{i=1}^n, \alpha, L, \{\sigma_i\}_{i=1}^s, Mode$
 $grad \leftarrow \text{CaculGradient}(L, M, y);$
 $r \leftarrow \text{random}(1, s);$
 $G \leftarrow (grad_r(1), \dots, grad_r(n))$ for *Plain*;
 $G \leftarrow (grad_{r, \sigma_r(1)-1}(i) \text{ for } i = 1 \text{ to } n)$ for *Ordered*;
 $T \leftarrow \text{empty tree};$
foreach *step of top-down procedure* **do**
 foreach *candidate split* c **do**
 $T_c \leftarrow \text{add split } c \text{ to } T;$
 if $Mode == Plain$ **then**
 $\Delta(i) \leftarrow \text{avg}(grad_r(p))$ for
 $p : leaf(p) = leaf(i)$ for all $i;$
 if $Mode == Ordered$ **then**
 $\Delta(i) \leftarrow \text{avg}(grad_{r, \sigma_r(i)-1}(p))$ for
 $p : leaf(p) = leaf(i), \sigma_r(p) < \sigma_r(i) \forall i;$
 $loss(T_c) \leftarrow ||\Delta - G||_2$
 $T \leftarrow \text{argmin}_{T_c}(loss(T_c))$
if $Mode == Plain$ **then**
 $M_{r'}(i) \leftarrow M_{r'}(i) - \alpha \text{avg}(grad_{r'}(p))$ for
 $p : leaf(p) = leaf(i)$ for all $r', i;$
if $Mode == Ordered$ **then**
 $M_{r', j}(i) \leftarrow M_{r', j}(i) - \alpha \text{avg}(grad_{r', j}(p))$ for
 $p : leaf(p) = leaf(i), \sigma_{r'}(p) \leq j$ for all $r', j, i;$
return T, M

Şekil 2.17-1 CatBoost Sözde Kodu.[24]

cat_features içinde dize değerlerine sahip bir sütun sağlanmazsa, CatBoost bir hata atar. Ayrıca, varsayılan int türüne sahip bir sütun varsayılan olarak sayısal olarak kabul edilir, algoritmanın kategorik olarak davranmasını sağlamak için cat_features içinde belirtmek gerekir.

Bir_hot_max_size değerinden daha fazla benzersiz kategoriye sahip kalan kategorik sütunlar için CatBoost, ortalama kodlamaya benzer, ancak overfitting azaltan etkili bir kodlama yöntemi kullanır. Süreç böyle gider –

- Girdi gözlemleri kümesinin rastgele sırada izin vermesi. Birden fazla rastgele permütasyon üretilir.
- Etiket değerini bir kayan noktadan veya kategoriden bir tam sayıya dönüştürme
- Tüm kategorik özellik değerleri, aşağıdaki formül kullanılarak sayısal değerlere dönüştürülür:

Catboost numerik dönüşüm denklemi:

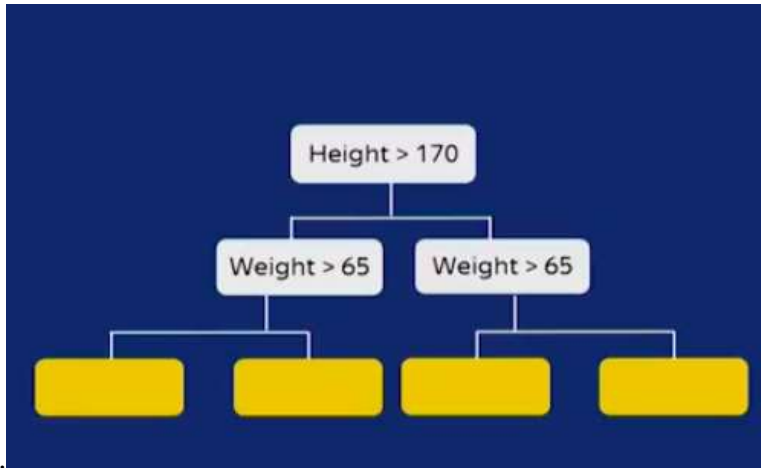
$$avg_target = \frac{countInClass + prior}{totalCount + 1} \quad (2.17.1)$$

Burada, CountInClass, geçerli kategorik özellik değerine sahip nesneler için etiket değerinin kaç kez "1" e eşit olduğu anlamına gelir. Önceki değer, payın ön değeridir. Başlangıç parametreleri ile belirlenir. TotalCount, geçerli olanla eşleşen kategorik özellik değerine sahip toplam nesne sayısıdır (geçerli olana kadar).

Matematiksel olarak, bu aşağıdaki denklem kullanılarak temsil edilebilir: [25]

$$Total\ Count = \frac{\sum_{j=1}^{p-1} [X_{a_j,k}=X_{a_p,k}]Y_{a_j} + a * P}{\sum_{j=1}^{p-1} [X_{a_j,k}=X_{a_p,k}] + a} \quad (2.17.2)$$

CatBoost ve diğer güçlendirici algoritmalar arasındaki temel farklardan biri, CatBoost'un simetrik ağaçları uygulamasıdır. Düşük gecikmeli ortamlar için son derece önemli olan tahmin süresinin azaltılmasına yardımcı olur.



Şekil 2.17-2 CatBoost temel ağaç simetriği.

Diğer gradient boosting algoritmaları için prosedür şöyledir (XGBoost, LightGBM)

- Adım 1: Oldukça önyargılı bir model oluşturmak için tüm veri noktalarını (veya bir örneği) düşünün.
- Adım 2: Her veri noktası için kalıntıları (hataları) hesaplayın.

- Adım 3: Sınıf etiketleriyle aynı veri noktalarına ve ilgili kalıntılara (hatalara) sahip başka bir model eğitin.
- Adım 4: Adım 2 ve Adım 3'ü tekrarlayın (tekrarlar için).

Bu prosedür overfitting eğilimlidir, çünkü her bir veri noktasının kalıntılarını, aynı veri noktaları kümesi üzerinde zaten eğitilmiş olan modeli kullanarak hesaplıyoruz.

CatBoost için prosedür şöyledir:

Verilerin zamanı yoksa, CatBoost her veri noktası için rastgele bir yapay zaman oluşturur.

- Adım 1: O anda tüm diğer veri noktalarında eğitilmiş bir model kullanarak her veri noktası için kalıntıları hesaplayın (Örneğin, x_5 veri noktası için kalıntıyı hesaplamak için x_1 , x_2 , x_3 ve x_4 kullanarak bir model eğitiyoruz) . Bu nedenle, farklı veri noktaları için kalıntıları hesaplamak üzere farklı modeller eğitiyoruz. Sonunda, karşılık gelen modelin bu veri noktasını daha önce hiç görmediği her veri noktası için kalıntıları hesaplıyoruz.
- 2.Adım: Her veri noktasının kalıntılarını sınıf etiketi olarak kullanarak modeli eğitin
- Adım 3: Adım 1 ve Adım 2'yi tekrarlayın (yineleme için) Kategorik Özelliklerin Kullanılması.

CatBoost, kategorik verilerin çok iyi bir vektör temsiline sahiptir. Sıralı güçlendirme kavramlarını alır ve aynısını yanıt kodlamasına uygular. Yanıt kodlamasında, her kategorik özelliği, aynı kategorik özelliğe sahip tüm veri noktalarının hedef değerlerinin ortalamasını kullanarak temsil ederiz. Sınıf etiketiyle veri noktasının özellik değerini temsil ediyoruz. Bu hedef sızıntısına yol açar. CatBoost yalnızca o ana kadar önceki veri noktalarını dikkate alır ve aynı kategorik özelliğe sahip veri noktalarının hedef değerlerinin ortalamasını hesaplar.[26]

2.17.1. CatBoost kütüphanesinin avantajları

Performans: CatBoost son teknoloji ürünü sonuçlar sağlar ve performans cephesinde önde gelen makine öğrenimi algoritmalarıyla rekabet edebilir.

Kategorik özellikleri otomatik olarak işleme: CatBoost'u, kategorileri sayılara dönüştürmek için açık bir ön işlem yapmadan kullanabiliriz. CatBoost, kategorik özelliklerin kombinasyonları ile kategorik ve sayısal özelliklerin kombinasyonları hakkındaki çeşitli istatistikleri kullanarak kategorik değerleri sayılara dönüştürür.

Sağlam: Kapsamlı hiper parametre ayarlama ihtiyacını azaltır ve daha fazla genelleştirilmiş modellere yol açan aşırı uyum olasılığını azaltır. CatBoost'un ayarlamak için birden fazla parametresi olmasına rağmen, ağaç sayısı, öğrenme oranı, düzenlenme, ağaç derinliği, kat boyutu, torbalama sıcaklığı ve diğerleri gibi parametreler içerir.

Kullanımı kolay: CatBoost'u hem Python hem de R için kullanıcı dostu bir API kullanarak komut satırından kullanabilirsiniz.[23]

2.18. Lojistik Regresyon

Bilinen doğrusal regresyon analizinde bağımlı değişken ve bağımsız değişkenler sayısal (ölçümle belirtilen sürekli ya da kesikli sayısal) olarak belirtilir. Örneğin, yaş ile kan basıncı arasında bir ilişki aranacaksa; hem yaş, hem de kan basıncı sayısal olarak belirtilmelidir. Nitelik olarak belirtilemezler.

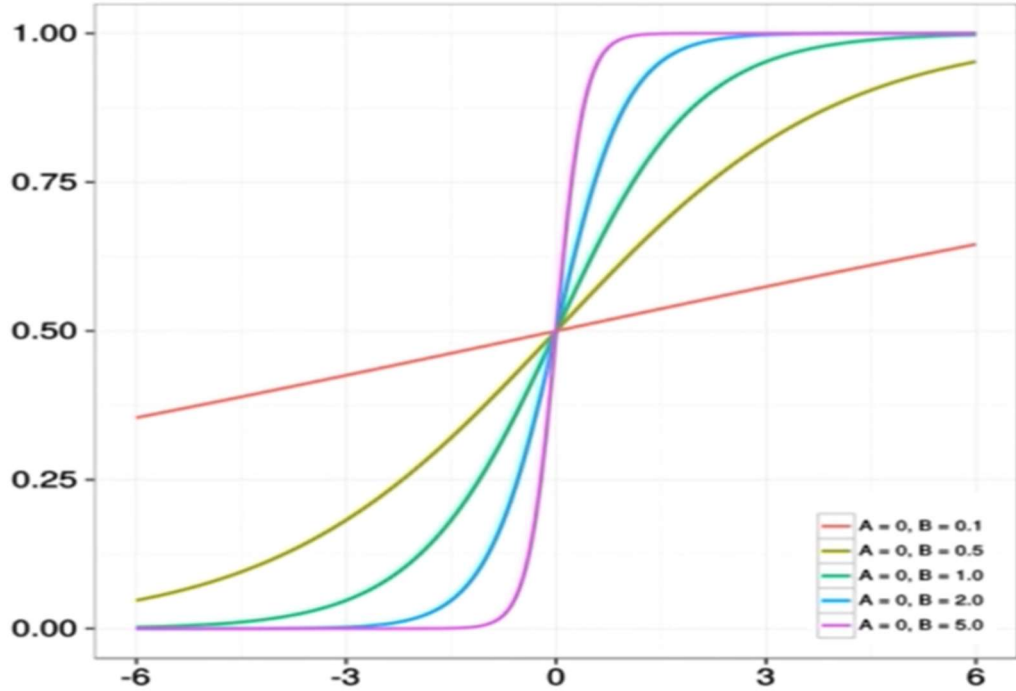
Bağımlı değişken nitelik olarak belirtilirse, bağımsız değişken ya da değişkenlerle arasındaki ilişki lojistik regresyon yöntemiyle aranır.

Lojistik regresyonun uygulandığı durumlar şunlardır: Bağımlı değişkenin kategori sayısına göre uygulanacak yöntem farklıdır. En çok uygulandığı durum bağımlı değişkenin iki kategorili (iyileşti-iyileşmedi gibi) olduğu durumdur.

Lojistik regresyon lineer bir regresyona benzer, ancak eğri olasılık yerine hedef değişkenin olasılıklarının doğal logaritması kullanılarak oluşturulur.

Lojistik regresyon yönteminin hedefi, bağımlı değişkenin sonucunu tahmin edebilecek en sade modeli bulmaktır. Lojistik regresyon analizi sonucunda elde edilen modelin uygun olup olmadığı “model ki-kare” testi ile, Her bir bağımsız değişkenin modelde varlığının anlamlı olup olmadığı ise Wald istatistiği ile test edilir.

- Odds: Odds başarı ya da görülme olasılığının “p”, başarısızlık ya da görülmemeye olasılığına “1p” oranıdır.
- Odds ratio (OR): İki odds’un birbirine oranıdır. İki değişken arasındaki ilişkinin özet bir ölçüsüdür.
- Odds ratio’nun doğal logaritmasıdır. Odds ratio asimettiktir. Doğal logaritması alınarak simetrik hale dönüştürülür. Lojit katsayıları (lojit) doğrusal regresyon analizindeki “b” katsayısının karşılığıdır. Paket programlar “b” katsayısının standart hatasını, anlamlılık için Wald istatistiğini, odds ratio ve odds ratio’nun güven aralığını vermektedir. [27]



Şekil 2.18-1 Logistic Regression kat sayı değişimleri [28]

$$\sigma(t) = \frac{e^t}{e^t + 1} = \frac{1}{1 + e^{-t}}$$

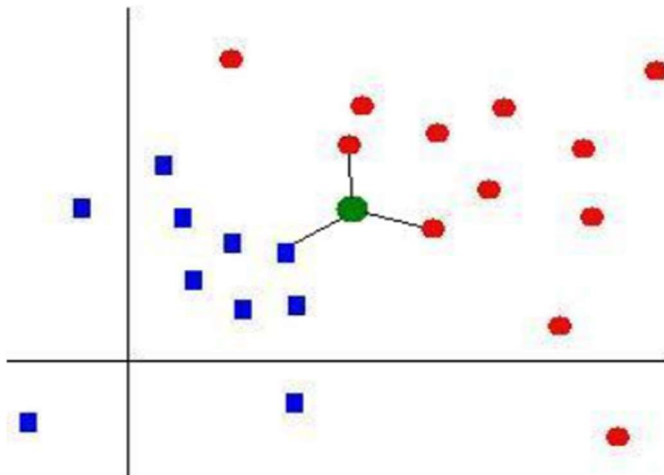
$$t = \beta_0 + \beta_1 x \quad t=A+Bx$$

$$p(x) = \frac{1}{1 + e^{-(\beta_0 + \beta_1 x)}} \quad (2.18.1)$$

$$\beta_0 + \beta_1 x_1 + \beta_2 x_2 + \cdots + \beta_m x_m = \beta_0 + \sum_{i=1}^m \beta_i x_i$$

2.19. KNN

KNN algoritması basitçe sizin belirlediğiniz komşu sayısı kadar en yakın komşusuna bakıp sınıflandırma yapan algoritmadır. En yakın komşu hesaplamasında genellikle Öklid mesafesi kullanılmaktadır. Fakat farklı mesafe hesaplama yöntemleri de kullanılabilir. Komşu sayısı arttırıldıkça algoritmanın doğruluğu artacak diye bir şey yoktur. En doğru komşu sayısı veriye göre değişir.



Şekil 2.19-1 KNN temsili komşular gösterimi [28]

KNN algoritması, uygulaması kolay gözetimli öğrenme algoritmalarındandır. Hem sınıflandırma hem de regresyon problemlerinin çözümünde kullanılıyor olmakla birlikte, endüstride çoğunlukla sınıflandırma problemlerinin çözümünde kullanılmaktadır.

KNN; eski, basit ve gürültülü eğitim verilerine karşı dirençli olması sebebiyle en popüler makine öğrenme algoritmalarından biridir. Fakat bunun yanında dezavantajı da mevcuttur. Örneğin, uzaklık hesabı yaparken bütün durumları sakladığından, büyük veriler için kullanıldığında çok sayıda bellek alanına gereksinim duymaktadır.

Yaygın Olarak Kullanılan Mesafe Fonksiyonları:

Öklid Denklemi:

$$d(x,y) = \sqrt{\sum_{i=1}^k (x_i - y_i)^2} \quad (2.19.1)$$

Minkowski Denklemi:

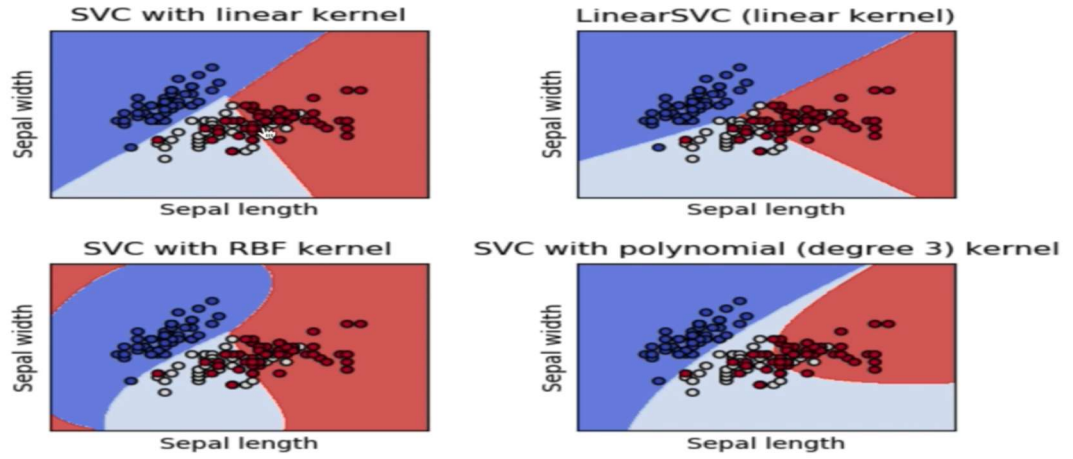
$$d(x,y) = \left(\sum_{i=1}^k [|x_i - y_i|^q] \right)^{1/q} \quad (2.19.2)$$

Basit olmasına rağmen oldukça başarılıdır ve diğer sınıflandırıcı algoritmalara göre eğitim maliyeti en düşük yaklaşımdır.

2.20. SVM

Support Vector Machine sınıflandırma algoritması Support Vector Regressiona benzer bir yapı sürdürür. Sınıflandırma için kullanılan oldukça başarılı ve basit algoritmalardan bir tanesidir. Bir düzlemde bulunan verileri sınıflandırmak için bu gruplar arasına bir sınır çizebilir. Bu sınır ise gruplara en uzak yerden çizilmelidir. SVM bu sınırın nereye çizilmesi gerektiğini belirlemeye çalışır.

Bu sınır çizgisi ise gruplara yakın ve birbirine paralel olarak çizilir. Bu çizgileri birbirine yaklaştırarak asıl sınır çizgisini elde etmeye çalışır.



Şekil 2.20-1 SVM farklı kernel fonksiyonları temsili [28]

Doğrusal bir Support Vector Machine (SVM) için formül:

$$u = \bar{w} * \bar{x} - b \quad (2.20.1)$$

Bu denklemde w , hiper düzlemdeki normal vektördür ve x , giriş vektörüdür. En yakın noktalar $u = \pm 1$ düzlemlerindedir. d mesafesi:

SVM'de d mesafesi:

$$d = \frac{1}{\|w\|_2} \quad (2.20.2)$$

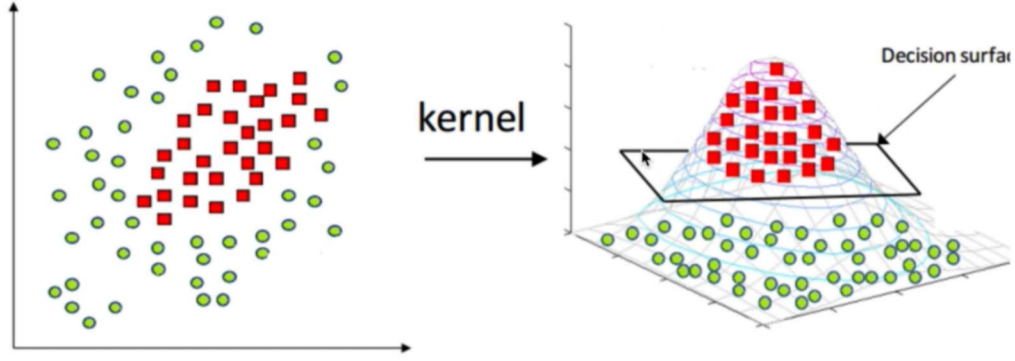
Maksimum d uzaklığı, optimizasyon problemi kullanılarak ifade edilebilir.

SVM Maksimum d optimizasyonu:

$$\min_{\bar{w}, b} \frac{1}{2} \|\bar{w}\|^2 y_i (\bar{w} * \bar{x} - b) \geq 1 \quad (2.20.3)$$

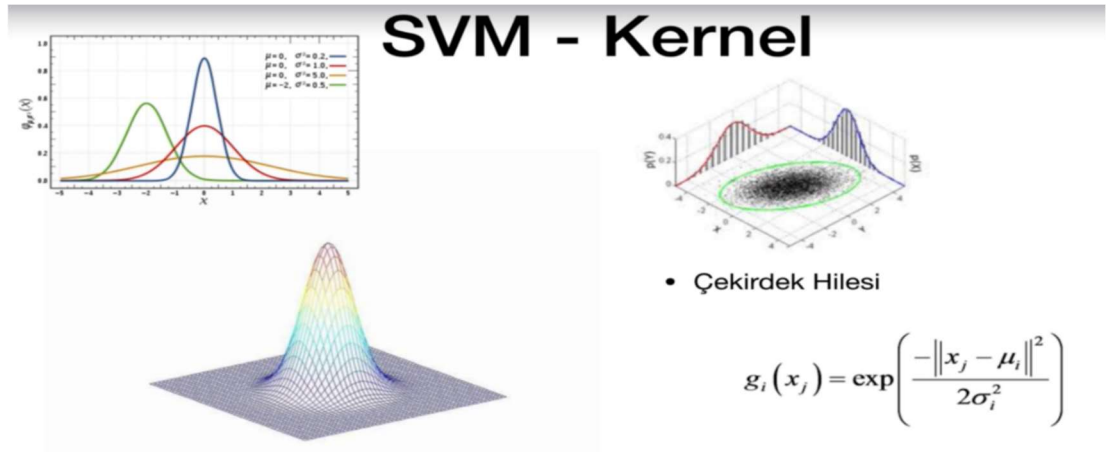
\bar{x}_i ve y_i Eğitilmiş örnekler için doğru çıktısıdır. y pozitif örnekler için $+1$, negatif örnekler için -1 alır.

Standart SVM iki sınıfı birbirinden ayırmak için daha iyidir. Fakat birden fazla sınıf kullanabilmek için bire karşı diğer sınıflar ya da çok sınıflı support vector machine yapıları kullanılabilir.



Şekil 2.20-2 SVM'de kernel trick [28]

Bunun gibi doğrusal olmayan bir veri yapısında SVM uygulayabilmek için boyut artırma kullanılır. Bir merkez noktası belirlendikten sonra o merkez noktası artırılan boyutta en yukarı çıkarılır. İki boyutlu düzlemde de ona yakın olan veriler benzer oranlarla yükseltilirken uzak olanlar daha aşağıda bırakılır. Bu sayede artık ikiye rahatlıkla ayırabileceğimiz iki farklı sınıfımız oluşmuş olur.



Şekil 2.20-3 SVM kernel trick 2 [28]

2.21. Karar Ağacı Sınıflandırma

Genel olarak regresyonla benzer bir algoritma fakat farkları var. Sınıflandırmada bölünme işlemini hesaplamak için birden fazla yöntem vardır. Algoritma seçimi, hedef

değişkenin tipine dayanır. Karar ağaçlarında en sık kullanılan algoritmalar; kategorik değişkenler için Entropi, Gini, Sınıflandırma Hatası; sürekli değişkenler için ise En Küçük Karalere yöntemi şeklindedir.

Entropi, verilerimizle ilgili belirsizliğin bir ölçüsüdür. Sezgisel olarak, bir veri kümesinin yalnızca bir etiketi varsa (örneğin, her yolcu hayatta kaldı), daha düşük bir entropiye sahip olduğunu düşünebiliriz. Dolayısıyla verilerimizi, entropiyi en aza indirecek bir şekilde bölmemiz gerekmektedir. Bölünmeler ne kadar iyi olursa, tahminimiz de o kadar iyi olur.

Entropi:

$$H = - \sum p(x) \log p(x) \quad (2.21.1)$$

Burada, $p(x)$ belirli bir sınıfa ait grubun yüzdesini ve H ise entropiyi belirtmektedir.

Karar ağacımızın entropi değerini en aza indirgeyen bölünmeler yapmasını isteriz. En iyi bölünmeyi belirlemek içinde bilgi kazancını kullanırız. Bilgi kazancı aşağıdaki eşitlik ile hesaplanır:

Karar ağacında bilgi kazancı formülü:

$$Gain(S, D) = H(S) - \sum_{V \in D} \frac{|V|}{|S|} H(V) \quad (2.21.2)$$

Burada, S orijinal veri kümesidir ve D ise kümenin bölünmüş bir parçasıdır. Her V , S 'nin bir alt kümesidir. V 'nin tümü ayırıktır ve S 'yi oluşturmaktadır. Bu durumda bilgi kazancı, bölünmeden önceki orijinal veri setinin entropisi ile her bir özniteliğin entropi değeri arasındaki fark olarak tanımlanmaktadır.

Karar ağacı algoritmasında da sınıflandırılacak veri seti eğitim(train) ve test olmak üzere iki parçaya ayrılmalıdır. Genelde toplam veriden rastgele seçilen 2/3 oranında veri eğitim için kalan 1/3 oranında veri ise test için seçilir. Seçilen eğitim verileri ile ağaç yapısı kurulur. Kurulan ağaç yapısı test verileri üzerinde denenerek ağaç yapısının problem üzerindeki başarı oranı hesaplanır.

En çok kullanılan karar ağacı algoritmaları: ID3,C4.5,CHAID,CART

$$Info(D) = - \sum_{i=1}^m p_i * \log_2(p_i)$$

$$Info_A(D) = \sum_{j=1}^y \frac{|D_j|}{|D|} * I(D_j) \quad (2.21.3)$$

$$Gain(A) = Info(D) - Info_A(D)$$

2.22. Naive Bayes

Naive Bayes bu algoritma koşullu olasılık üzerine çalışıyor. Sisteme verilen önceki verileri işleyip yeni gelenlerin olasılığını ona göre ölçen bir algoritmadır.

Naïve Bayes sınıflandırması koşullu olasılık kullanılarak yapılan bazı hesaplamalar sonucunda sisteme verilen verileri eğiterek yeni gelen ve sınıfı bilinmeyen verileri sınıflandırmaya çalışır. Örnek vermek gerekirse kişi kanser ya da kanser değil gibi bir sınıfa atamaya çalışır.

Naïve Bayes sınıflandırmasında sisteme belirli bir oranda öğretilmiş veri sunulur (Örn: 100 adet). Öğretim için sunulan verilerin mutlaka bir sınıfı/kategorisi bulunmalıdır. Öğretilmiş veriler üzerinde yapılan olasılık işlemleri ile, sisteme sunulan yeni test verileri, daha önce elde edilmiş olasılık değerlerine göre işletilir ve verilen test verisinin hangi kategoride olduğu tespit edilmeye çalışılır. Elbette öğretilmiş veri sayısı ne kadar çok ise, test verisinin gerçek kategorisini tespit etmek o kadar kesin olabilmektedir.

Naïve Bayes sınıflandırma algoritmasında veriler ile olasılıksal bir sonuç çıkarımı yaptığımız için öğretilecek veriler binary veya test veriler olabilir. Sayısal ya da kategorik veriler de olabilir. Veri tipinden veya türünden ziyade önemli nokta veriler arasında kurduğumuz oransal ilişkidir.

Bayes sınıflandırıcısının, gen ifade verilerinde diğer sınıflandırıcılara göre daha başarılı sonuçlar elde ettiğine dair gözlemler bulunmaktadır. Bayes'in olasılık denklemi ise şu şekildedir:

$$P(A/B) = \frac{P\left(\frac{B}{A}\right) * P(A)}{P(B)} \quad (2.22.1)$$

Burada:

$P(A|B)$ = B olayı gerçekleştiğinde A olayının gerçekleşme olasılığı

$P(A)$ = A olayının gerçekleşme olasılığı

$P(B|A)$ = A olayı gerçekleştiğinde B olayının gerçekleşme olasılığı

$P(B)$ = B olayının gerçekleşme olasılığı

2.23. Random Forest

Rasgele orman, adından da anlaşılacağı gibi, bir topluluk olarak çalışan çok sayıda bireysel karar ağacından oluşur. Rastgele ormandaki her bir ağaç bir sınıf tahmini verir ve en çok oy alan sınıf modelimizin tahmini olur.

Rastgele ormanın ardındaki temel kavram, basit ama güçlü bir kavramdır - kalabalıkların bilgeliği. Veri biliminde, rastgele orman modelinin bu kadar iyi çalışmasının nedeni:

Komite olarak faaliyet gösteren göreceli olarak ilişkisiz çok sayıda model (ağaç), tek tek kurucu modellerin herhangi birinden daha iyi performans gösterecektir.

Modeller arasındaki düşük korelasyon anahtardır. Tıpkı düşük korelasyonlu (hisse senetleri ve tahviller gibi) yatırımların parçalarının toplamından daha büyük bir portföy oluşturmak için nasıl bir araya geldiği gibi, ilişkisiz modeller de bireysel tahminlerden daha doğru olan topluluk tahminleri üretebilir. Bu harika etkinin nedeni, ağaçların birbirlerini bireysel hatalarından korumalarıdır (sürekli olarak aynı yönde hata yapmadığı sürece). Bazı ağaçlar yanlış olsa da, diğer birçok ağaç haklı olacaktır,

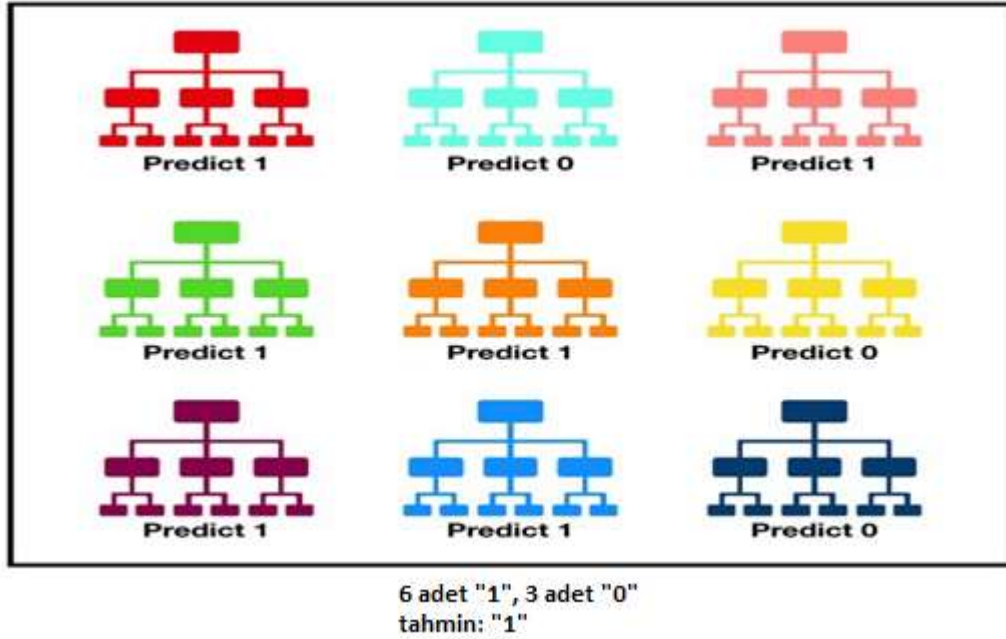
bu nedenle bir grup olarak ağaçlar doğru yönde hareket edebilir. Rastgele ormanın iyi performans göstermesi için önkoşullar:

Özelliklerimizde gerçek bir sinyal olması gerekir, böylece bu özellikleri kullanarak oluşturulan modeller rastgele tahmin etmekten daha iyi sonuç verir.

Her bir ağaç tarafından yapılan tahminlerin (ve dolayısıyla hataların) birbirleriyle düşük korelasyonları olmalıdır.[29]

Rasgele Orman Algoritmasının Çalışması şöyledir: Rastgele Orman algoritmasının çalışmasını aşağıdaki adımların yardımıyla anlayabiliriz -

- Adım 1 - İlk olarak, belirli bir veri kümesinden rastgele örneklerin seçilmesiyle başlayın.
- Adım 2 - Daha sonra, bu algoritma her örnek için bir karar ağacı oluşturacaktır. Sonra her karar ağacından tahmin sonucunu alır.
- Adım 3 - Bu adımda, tahmin edilen her sonuç için oylama yapılacaktır.
- Adım 4 - Son olarak, nihai tahmin sonucu olarak en çok oylanan tahmin sonucunu seçin.[30]



Şekil 2.23-1 Rastgele Orman tahmini temsili [29]

3. UYGULANAN YÖNTEMLER

Tezin bu kısmında bahsi geçecek tüm yazılım, kod ve teknik dökümanlara çalışmaya ait GitHub hesabı üzerinden erişilebilmektedir[32].

3.1. Veri Seçimi

Verileri edindiğimiz platform olan myvariant.info'dan en yüksek sayıda BRCA1 ve BRCA2 varyantı içeren veri tabanları. Platformun Web API 'ına Python dilinden yapılan sorgularla belirlenmiştir. Bu sorgu sonucunda veri tabanı isimleri ve içerdikleri BRCA1, BRCA3 varyantları sayısı aşağıdaki gibidir:

Tablo 3.1.1 Veritabanlarında bulunan varyant sayıları

Veri Tabanı	BRCA1+BRCA2 Varyantları
dbSNP	55834
CADD	50571
dbNSFP	38781

Bu veri tabanlarından dbSNP ve CADD'in varyant sayıları yakın olduğundan öğrenme için oldukça önemli bir parametre olan özellik sayısına göre seçim yapılmasına karar verilmiştir. Bu iki veri tabanının platform üzerinden erişilebilecek özellik sayılarına platformun internet sitesi üzerinden erişilebilmektedir ve şöyledir:

Tablo 3.1.2 dbSNP ve CADD veri tabanlarının özellik sayıları

Veri Tabanı	Özellik Sayısı
dbSNP	58
CADD	135

Bu ölçümler sonucunda kullanılacak veri tabanı olarak CADD seçilmiştir.

3.2. Veri Hazırlama

Seçilen CADD veri tabanına ait BRCA1 ve BRCA2 geni varyantları yine myvariant.info web API sorgularıyla Python programlama dili aracılığıyla indirilmiş ve toplamda 50,571 adet veri JSON yapısında olması sebebiyle daha kolay incelenebilmesi adına lokal mongoDB database sistemi içerisine kayıt edilmiştir.

Veriler varyanta ait bir ID sistemi ile kimliklendirilmiş olarak gelir, aşağıda bir örneği mevcuttur:


cadd	{ 35 attributes }	Object
1000g	{ af : 0.01, afr : 0.04, eur : 0.01 }	Object
_license	http://bit.ly/2Tluab9	String
alt	A	String
anc	G	String
annotype	["Intergenic", "Transcript"]	Array
bstatistic	111	Int32
chmm	{ 14 attributes }	Object
chrom	17	Int32
consdetail	["upstream", "intron,nc"]	Array

Şekil 3.2-1 Mongo DB'de verinin görünümü

Veri yapısı 50,571x135'tir ancak bu varyantların tamamı öğrenme için gerekli olan klinik statü yani patojenite bilgisini tutan 'rcv.clinival_significance' parametresini içermez. Bu değerler eğitimin gerçekleştirilebilmesi için filtrelendir. Bu işlem sonucunda verinin yeni boyutu silinen kayıtlar sebebiyle (12332 x 120) olarak değişir. Sütun sayısındaki değişim geriye kalan satır değerlerinin hiçbirinin ilgili sütuna ait değer taşınamamasından kaynaklanır.

Her bir veri kaydı yani varyanta ait özellik uzayı değişebilmekte yani her varyant 120 parametrenin farklı kombinasyonlarını bulundurabilmektedir. Ayrıca elde edilen verinin formatı ağaç yapısında olum JSON formatındadır. Yaygın kullanılan kütüphane ve algoritmalar tarafından işlenebilmesi için bu veriler tablo formatına dönüştürülmüştür. Bu esnada bir parametreye ait değer taşımayan varyantlarda ilgili parametrenin sütunu daha sonra veri ön işleme adımlarında doldurulmak üzere boş bırakılmıştır.

Bu işlem, kendi geliştirdiğimiz Python programlama dilinde yazılmış araçlar sayesinde gerçekleştirilmiştir.



```

{
  "_id": "chr1/g.41261008C>T",
  "_score": 2,
  "cadd": {
    "1000g": {
      "af": 0.005,
      "afr": 0.02
    },
    "_license": "http://bit.ly/2Tluab9",
    "alt": "T",
    "anc": "C",
    "annotype": "Transcript",
    "bstatistic": 111,
    "chmm": {
      "bivlink": 0,
      "enh": 0,
      "enhbiv": 0,
      "het": 0
    }
  }
}

```

cadd.alt	cadd.anc	cadd.ann	cadd.bsta	cadd.chro	cadd.cons	cadd.cons
T	C	Transcript	111	17	intron	INTRONIC
A	G	Transcript	111	17	intron	INTRONIC
A	G	CodingTra	111	17	missense	NON_SYN
A	T	CodingTra	111	17	missense	NON_SYN
G	T	CodingTra	115	17	missense	NON_SYN
A	G	Transcript	112	17	intron	INTRONIC
G	G	Transcript	115	17	intron	INTRONIC
A	T	CodingTra	116	17	synonym	SYNONYM
C	T	CodingTra	116	17	missense	NON_SYN
T	C	CodingTra	115	17	missense	NON_SYN
G	A	CodingTra	115	17	missense	NON_SYN
G	A	CodingTra	374	13	missense	NON_SYN
C	A	CodingTra	398	13	missense	NON_SYN

Şekil 3.2-2 Veride yapısal dönüşüm

3.3. Veri Önleme

Veriler tablo formatına dönüştürüldükten sonra .csv formatında kaydedilmiş işlemeye hazır duruma getirilmiştir. Daha sonra tablodaki sütunlar yani varyanta ait anotasyonlar veri bilimi dünyasındaki adıyla parametreler aşağıdaki kriterlere uyması durumunda istenmeyen özellik olarak seçilip veriden çıkarılmıştır.

1. ID yani kimlik bilgisi gibi genelleştirilemeyen özgün bilgiyi temsil ediyor mu?
2. 20% referans değerinden daha düşük doluluk oranına sahip mi?

Parametre eleme işleminden sonra sütun sayısı 120'den 74'e düşerek verinin yeni boyutu (12332 x 74) olarak değişir.

3.3.1. Krite 1'e göre silinen sütunlar

Orijinal verideki isimleri ile “_id, cadd._license, clinvar._license, clinvar.rsid, _score“ olan 5 adet sütun bu kritere göre elenmiştir.

3.3.2. Kriter 2'ye göre silinen sütunlar

Referans değer olarak alınan 20%'nin üzerinde boş değer bulunduran bu **41** sütun elenmiştir.

Tablo 3.3.1 Doluluk Oranı sebebiyle elenen sütunlar

		Boş Değer Sayısı	Boş Değer Yüzdesi			Boş Değer Sayısı	Boş Değer Yüzdesi
Sütun Adı				Sütun Adı			
1 motif.ecount		12329	99.9757	22 1000g.af		11232	91.0801
2 motif.ehipos		12329	99.9757	23 cadd.dst2splice		10706	86.8148
3 motif.ename		12329	99.9757	24 cadd.dst2spltype		10706	86.8148
4 motif.escorchng		12329	99.9757	25 encode.occ		10571	85.7201
5 motif.dist		12240	99.254	26 p_val.comb		10571	85.7201
6 motif.toverlap		12240	99.254	27 p_val.ctcf		10571	85.7201
7 mirsvr.aln		12214	99.0431	28 p_val.dnas		10571	85.7201
8 mirsvr.e		12214	99.0431	29 p_val.faire		10571	85.7201
9 mirsvr.score		12214	99.0431	30 p_val.mycp		10571	85.7201
10 cadd.scoresegdup		12186	98.8161	31 p_val.polii		10571	85.7201
11 1000g.asn		11870	96.2536	32 sig.ctcf		10571	85.7201
12 1000g.eur		11795	95.6455	33 sig.dnase		10571	85.7201
13 esp.af		11758	95.3454	34 sig.faire		10571	85.7201
14 esp.afr		11758	95.3454	35 sig.myc		10571	85.7201
15 esp.eur		11758	95.3454	36 sig.polii		10571	85.7201
16 tf.bs		11700	94.8751	37 sift.cat		4854	39.361
17 tf.bs_peaks		11700	94.8751	38 sift.val		4854	39.361
18 tf.bs_peaks_max		11700	94.8751	39 cadd.grantham		4853	39.3529
19 1000g.amr		11657	94.5264	40 polyphen.cat		4853	39.3529
20 1000g.afr		11516	93.3831	41 polyphen.val		4853	39.3529
21 cadd.intron		11514	93.3669				

Daha sonra %20'nin altında boş değer içeren sütunların boş hücrelerinin doldurulması işlemine geçilir. Geriye kalan veride toplam **912,568** hücreden **15,742** tanesi yani **1.73%**'ü doldurulmak üzere ayrılan boş verilerdir. Buradan itibaren bu verilerin doldurulması işleminden aşağıda bahsedilmektedir.

Imputation olarak da bilinen bu işlem için **scikit-learn** kütüphanesine ait **SimpleImputer** modülü kullanılmış ve aşağıdaki 2 strateji izlenmiştir.

- **Numerik Veriler İçin:** Boş hücreler; ilgili sütunun aritmetik ortalama değeri ile doldurulur. (mean)
- **Metinsel Veriler İçin:** Boş hücreler; ilgili sütunda en sık geçen ifade ile doldurulur. (most_frequent)

Çalışmanın başında da belirtildiği gibi temel olarak 4 tip patojenite sınıfı vardır, Bunlar: Pathogenic, Likely Pathogenic, Likely Benign ve Benign'dır. VUS ise hangi sınıfa ait olduğu bilinmeyen durumu ifade eder. Ancak elde edilen verilerde bunlarında dışında “not_provided”, “other”, risk factor”, “Conflicting interpretations of pathogenicity”, “Benign/Likely Benign”, “Pathogenic/Likely Pathogenic” gibi ACMG temel standartlarına uymayan dolayısıyla bu çalışmanın kapsamının da dışında kalan sınıf verileri bulunmaktadır. Veriden bu sınıflar ve eğitimin gerçekleştirilmesi adına VUS yani verideki adıyla “Uncertain significance” sınıfı filtrelenerek çıkarılmıştır. Bu işlemlerin sonucunda verinin yeni boyutu **(5166 x 74)** olarak değişmiş ve son formunu almıştır.

Farklı sınıflandırıcı metotların farklı tipte verilerle çalışması nedeniyle verinin bu son hali üzerinden 2 farklı versiyon üretilmiş ve algoritmaya göre seçim yapılarak kullanılmıştır. Bunlar:

1. **variants_encoded.csv:** Hedef sınıf olan “rcv.clinical_significance” sınıfı hariç tüm kategorik veriler kodlanarak(**encoding**) numerik değerlere dönüştürülmüş csv formatında kaydedilmiştir. Bu işlem için **scikit-learn** kütüphanesinin **Label Encoder** metodu uygulanmıştır.
2. **variants_not_encoded.csv:** Hiçbir kategorik sütun kodlanmadan csv formatında kaydedilmiştir.

Yazının geri kalanında bu versiyonlardan isimleri ile bahsedilecektir.

Sınıflandırma metotları veri üzerinde çalıştırılırken ihtiyaç duyulduğu durumlarda veri tüm sütunların sonuca başlangıçta aynı oranda etki etmesi ve aynı uzayda temsil edilebilmesi adına 0-1 arasına ölçeklenmiştir.

3.4. Basit Sınıflandırma Metotları Uygulaması

Grid Search: Burada kullanılacak algoritmanın olası tüm parametrelerinden performansı etkileyeceği düşünülen parametreler seçilerek bunlar üzerinde bir dizi deneme yapıp en iyi kombinasyon seçilmeye çalışılır. Verilen her bir parametreye karşılık gelen diziler ilgili parametreye ait denenecek tüm ihtimalleri listeler. Tüm bu listeler çaprazlanarak bütün kombinasyonlar denenir, en iyi çıktıyı veren parametre seti seçilir.

Elde edilen parametre seti: Buradaki parametreler ilgili algoritmanın kullanılan kütüphane ile gelen varsayılan parametrelerini ve bunların yanında **grid search** ile belirlenerek varsayılan değerinin dışında değer atanan parametreleri içerir. Algoritmanın çalıştırıldığı nihai parametre setidir.

Algoritma Çıktısı: Bu başlık altında yer alan tablo. İlgili algoritmanın veriler üzerindeki sınıflandırma performansını ortaya koyar. Burada okunması gereken parametreler ve açıklamaları şöyledir:

- **Ortalama Accuracy:** Verilerden yüzde kaçının doğru olarak sınıflandırıldığını ifade eder.
- **f-1 score:** Satırın en solunda ismi verilen sınıfa ait sınıflandırma performansını ifade eder.
- **Support:** Üzerinde sınıflandırma modelinin uygulandığı veride ilgili sınıfa ait kaç adet veri bulunduğunu ifade eder. Beklendiği üzere yüksek sayıda örnek içeren sınıflar modeller tarafından çok daha iyi öğrenilmiş ve yüksek doğruluklar sunmuştur.

Tüm basit sınıflandırma metotları için **scikit-learn(0.22.1)** kütüphanesinin ilgili modülleri kullanılmış, implemantasyonları tarafımızca, **Python(3.7)** programlama dilinde gerçekleştirilmiştir.

Aşağıdaki maddeler tüm basit sınıflandırma metotları için geçerlidir.

- **variants_encoded.csv** kullanılmıştır
- Ölçekleme işlemi yapılmıştır.
- Orijinal yüzde dağılımlı 5 parçalı çapraz doğrulama uygulanmıştır. (**StratifiedKFold**)

3.4.1. Logistic Regression

Algoritmanın parametre optimizasyonu için **grid_searchCV** algoritması 5 parçalı çapraz doğrulama ile çalıştırılmış, optimize edilmiş parametreler ve bu parametreler için çıktılar aşağıda yer almaktadır. Çapraz doğrulamada her bir parça için ayrı bir çıktı elde edileceğinden bu parçaların ortalama sonuçları hesaplanarak gösterilmektedir.

Deneme 1: En iyi olmayan parametre setini ve çıktılarını ifade eder.

- **grid_search'de çalıştırılan parametre seti:**

```
{'solver': ['newton-cg', 'saga', 'lbfgs'], 'C':[0.01,0.09,0.5,1,5,10], 'class_weight':  
['balanced', None], 'max_iter': [50,100,250,500]}
```

- **Elde edilen parametre seti:**

```
{'C': 0.09, 'class_weight': None, 'dual': False, 'fit_intercept': True, 'intercept_scaling': 1,  
'l1_ratio': None, 'max_iter': 50, 'multi_class': 'auto', 'n_jobs': None, 'penalty': 'l2',  
'random_state': 0, 'solver': 'lbfgs', 'tol': 0.0001, 'verbose': 0, 'warm_start': False}
```

- **Algoritma Çıktısı:**

```
Ortalama Accuracy :
91.88912876993452
Ortalama Classification Report :
              precision    recall  f1-score   support

   Benign           0.87       0.86       0.87       1005
  Likely benign      0.94       0.95       0.95       2876
Likely pathogenic    0.12       0.03       0.05         90
   Pathogenic       0.91       0.96       0.94       1195

 accuracy                   0.92       5166
 macro avg           0.71       0.70       0.70       5166
 weighted avg       0.91       0.92       0.91       5166
```

Şekil 3.4-1 Logistic Regression Classification Report Çıktısı

Final: En iyi parametre setini ifade eder. Bu çalışmanın çıktıları kısmında algoritma çıktısı yer almaktadır

- **Elde edilen parametre seti:**

```
{'C': 1.0, 'class_weight': None, 'dual': False, 'fit_intercept': True, 'intercept_scaling': 1,
'l1_ratio': None, 'max_iter': 100, 'multi_class': 'auto', 'n_jobs': None, 'penalty': 'l2',
'random_state': 0, 'solver': 'lbfgs', 'tol': 0.0001, 'verbose': 0, 'warm_start': False}
```

3.4.2. KNN

Aşağıda seçilen parametreler ve bu parametreler için çıktıları yer almaktadır. Çapraz doğrulamada her bir parça için ayrı bir çıktı elde edileceğinden bu parçaların ortalama sonuçları hesaplanarak gösterilmektedir.

Final: Bu çalışmanın çıktıları kısmında da buradaki algoritma çıktısı yer almaktadır

- **Tüm parametre seti:**

```
{'algorithm': 'auto', 'leaf_size': 30, 'metric': 'minkowski', 'metric_params': None, 'n_jobs':
None, 'n_neighbors': 5, 'p': 2, 'weights': 'uniform'}
```


- **Algoritma Çıktısı:**

```
Ortalama Accuracy :
71.58358314874144
Ortalama Classification Report :
              precision    recall  f1-score   support

   Benign           0.83       0.72       0.77       1005
  Likely benign     0.70       0.90       0.79       2876
Likely pathogenic   0.23       0.07       0.10         90
   Pathogenic       0.65       0.33       0.44       1195

 accuracy                   0.72       5166
 macro avg           0.60       0.50       0.52       5166
 weighted avg        0.71       0.72       0.69       5166
```

Şekil 3.4-2 KNN Classification Report Çıktısı

3.4.3. SVM

Aşağıda seçilen parametreler ve bu parametreler için çıktılar yer almaktadır. Çapraz doğrulamada her bir parça için ayrı bir çıktı elde edileceğinden bu parçaların ortalama sonuçları hesaplanarak gösterilmektedir.

Final: Bu çalışmanın çıktılar kısmında da buradaki algoritma çıktısı yer almaktadır

- **Tüm parametre seti:**

```
{'C': 1.0, 'break_ties': False, 'cache_size': 200, 'class_weight': None, 'coef0': 0.0,
'decision_function_shape': 'ovr', 'degree': 3, 'gamma': 'scale', 'kernel': 'poly', 'max_iter': -1,
'probability': False, 'random_state': None, 'shrinking': True, 'tol': 0.001, 'verbose': False}
```

- **Algoritma Çıktısı:**

```

Ortalama Accuracy :
91.52147413872198
Ortalama Classification Report :

```

	precision	recall	f1-score	support
Benign	0.90	0.80	0.85	1005
Likely benign	0.92	0.96	0.94	2876
Likely pathogenic	0.12	0.02	0.04	90
Pathogenic	0.92	0.96	0.94	1195
accuracy			0.92	5166
macro avg	0.72	0.69	0.69	5166
weighted avg	0.90	0.92	0.91	5166

Şekil 3.4-3 SVM Classification Report Çıktısı

3.4.4. Gaussian Naive Bayes

Aşağıda seçilen parametreler ve bu parametreler için çıktılar yer almaktadır. Çapraz doğrulamada her bir parça için ayrı bir çıktı elde edileceğinden bu parçaların ortalama sonuçları hesaplanarak gösterilmektedir.

Final: Bu çalışmanın çıktılar kısmında da buradaki algoritma çıktısı yer almaktadır

- **Tüm parametre seti:**

```
{'priors': None, 'var_smoothing': 1e-09}
```

- **Algoritma Çıktısı:**

```

Ortalama Accuracy :
73.65836486843263
Ortalama Classification Report :

```

	precision	recall	f1-score	support
Benign	0.78	0.81	0.79	1005
Likely benign	0.95	0.69	0.80	2876
Likely pathogenic	0.06	0.63	0.10	90
Pathogenic	0.95	0.79	0.86	1195
accuracy			0.74	5166
macro avg	0.68	0.73	0.64	5166
weighted avg	0.90	0.74	0.80	5166

Şekil 3.4-4 Naive Bayes Classification Report Çıktısı

3.4.5. Decision Tree Classifier

Aşağıda seçilen parametreler ve bu parametreler için çıktılar yer almaktadır. Çapraz doğrulamada her bir parça için ayrı bir çıktı elde edileceğinden bu parçaların ortalama sonuçları hesaplanarak gösterilmektedir.

Final: Bu çalışmanın çıktıları kısmında da buradaki algoritma çıktısı yer almaktadır

- **Tüm parametre seti:**

```

{'ccp_alpha': 0.0, 'class_weight': None, 'criterion': 'entropy', 'max_depth': None,
'max_features': None, 'max_leaf_nodes': None, 'min_impurity_decrease': 0.0,
'min_impurity_split': None, 'min_samples_leaf': 1, 'min_samples_split': 2,
'min_weight_fraction_leaf': 0.0, 'presort': 'deprecated', 'random_state': None, 'splitter':
'best'}

```

- **Algoritma Çıktısı:**

```
Ortalama Accuracy :
90.10830223513793
Ortalama Classification Report :
              precision    recall  f1-score   support

   Benign           0.83      0.84      0.84       1005
  Likely benign     0.94      0.93      0.93       2876
Likely pathogenic   0.19      0.19      0.19         90
   Pathogenic       0.93      0.93      0.93       1195

 accuracy          0.90      0.90      0.90       5166
 macro avg          0.72      0.72      0.72       5166
 weighted avg       0.90      0.90      0.90       5166
```

Şekil 3.4-5 Decision Tree Classifier Classification Report Çıktısı

3.4.6. Random Forest Classifier

Algoritmanın parametre optimizasyonu için **grid_searchCV** algoritması 5 parçalı çapraz doğrulama ile birden fazla parametre seti için çalıştırılmış, optimize edilmiş parametreler ve bu parametreler için çıktılar aşağıda yer almaktadır. Çapraz doğrulamada her bir parça için ayrı bir çıktı elde edileceğinden bu parçaların ortalama sonuçları hesaplanarak gösterilmektedir.

Deneme 1: En iyi olmayan parametre setini ve çıktılarını ifade eder.

- **grid_search'de çalıştırılan parametre seti:**

```
{'bootstrap': [True, False], 'max_depth': [10, 30, 50, 70, 90, 100, None], 'max_features':
['auto', 'sqrt'], 'min_samples_leaf': [1, 2, 4], 'min_samples_split': [2, 5, 10],
'n_estimators': [200, 400, 1000, 1400, 1800]}
```

- **Elde edilen parametre seti:**

```
{'bootstrap': [True, False], 'max_depth': [10, 30, 50, 70, 90, 100, None], 'max_features':
['auto', 'sqrt'], 'min_samples_leaf': [1, 2, 4], 'min_samples_split': [2, 5, 10],
'n_estimators': [200, 400, 1000, 1400, 1800]}
```

- **Algoritma Çıktısı:**

```

Ortalama Accuracy :
91.83115786398932
Ortalama Classification Report :

```

	precision	recall	f1-score	support
Benign	0.89	0.82	0.86	1005
Likely benign	0.93	0.96	0.94	2876
Likely pathogenic	0.34	0.12	0.18	90
Pathogenic	0.92	0.96	0.94	1195
accuracy			0.92	5166
macro avg	0.77	0.72	0.73	5166
weighted avg	0.91	0.92	0.91	5166

Şekil 3.4-6 Random Forest Classifier Classification Report Çıktısı

Final: En iyi parametre setini ifade eder. Bu çalışmanın çıktıları kısmında algoritma çıktısı yer almaktadır.

- **Elde edilen parametre seti:**

```
{'bootstrap': True, 'ccp_alpha': 0.0, 'class_weight': None, 'criterion': 'entropy',
'max_depth': None, 'max_features': 'auto', 'max_leaf_nodes': None, 'max_samples':
None, 'min_impurity_decrease': 0.0, 'min_impurity_split': None, 'min_samples_leaf':
1, 'min_samples_split': 2, 'min_weight_fraction_leaf': 0.0, 'n_estimators': 10, 'n_jobs':
None, 'oob_score': False, 'random_state': None, 'verbose': 0, 'warm_start': False}
```

3.5. Boosted Tree Sınıflandırma Metotları Uygulaması

Orijinal yüzde dağılımlı 5 parçalı çapraz doğrulama uygulanmıştır. (**StratifiedKfold**)

XGBoost, LightGBM, CatBoost metotları için ayrıntılı, grid_search, feature importance parametreleri ve çıktıları eklenecek.

3.5.1. XGBoost:

Algoritmanın parametre optimizasyonu için **grid_searchCV** algoritması 5 parçalı çapraz doğrulama ile birden fazla parametre seti için çalıştırılmış, optimize edilmiş parametreler ve bu parametreler için çıktılar aşağıda yer almaktadır. Çapraz doğrulamada her bir parça için ayrı bir çıktı elde edileceğinden bu parçaların ortalama sonuçları hesaplanarak gösterilmektedir.

Deneme 1: En iyi olmayan parametre setini ve çıktılarını ifade eder.

- **grid_search'de çalıştırılan parametre seti:**

```
param_grid = {"max_depth": [10,30,50], "min_child_weight" : [1,3,6],  
              "n_estimators": [200], "learning_rate": [0.05, 0.1,0.16]}
```

- **Elde edilen parametre seti:**

```
{'base_score': 0.5, 'booster': 'gbtree', 'colsample_bylevel': 1, 'colsample_bynode':  
1, 'colsample_bytree': 1, 'gamma': 0, 'learning_rate': 0.1, 'max_delta_step': 0,  
'max_depth': 10, 'min_child_weight': 1, 'missing': None, 'n_estimators': 200,  
'n_jobs': 1, 'nthread': None, 'objective': 'multi:softprob', 'random_state': 0,  
'reg_alpha': 0, 'reg_lambda': 1, 'scale_pos_weight': 1, 'seed': None, 'silent': None,  
'subsample': 1, 'verbosity': 1}
```

- **Algoritma Çıktısı:**

```
Ortalama Accuracy :  
91.56023375606907  
Ortalama Classification Report :  
              precision    recall  f1-score   support  
  
   Benign           0.84      0.87      0.85       968  
  Likely benign     0.95      0.94      0.94      2919  
Likely pathogenic   0.12      0.25      0.16        44  
   Pathogenic       0.96      0.93      0.94      1235  
  
   accuracy                   0.92      5166  
  macro avg           0.72      0.75      0.73      5166  
weighted avg           0.92      0.92      0.92      5166
```

Şekil 3.5-1 XGBoost Classification Report Çıktısı

Final: En iyi parametre setini ifade eder. Bu çalışmanın çıktıları kısmında algoritma çıktısı yer almaktadır.

- **grid_search’de çalıştırılan parametre seti:**

```
param_grid = {"max_depth": [10,30,50], "min_child_weight" : [1,3,6],  
"n_estimators": [200], "learning_rate": [0.05, 0.1,0.16] }
```

- **Elde edilen parametre seti:**

```
{'base_score': 0.5, 'booster': 'gbtree', 'colsample_bylevel': 1, 'colsample_bynode':  
1, 'colsample_bytree': 1, 'gamma': 0, 'learning_rate': 0.05, 'max_delta_step': 0,  
'max_depth': 10, 'min_child_weight': 6, 'missing': None, 'n_estimators': 200,  
'n_jobs': 1, 'nthread': None, 'objective': 'multi:softprob', 'random_state': 0,  
'reg_alpha': 0, 'reg_lambda': 1, 'scale_pos_weight': 1, 'seed': None, 'silent': None,  
'subsample': 1, 'verbosity': 1}
```

3.5.2. LightGBM

Algoritmanın parametre optimizasyonu için **grid_searchCV** algoritması 5 parçalı çapraz doğrulama ile birden fazla parametre seti için çalıştırılmış, optimize edilmiş parametreler ve bu parametreler için çıktılar aşağıda yer almaktadır. Çapraz doğrulamada her bir parça için ayrı bir çıktı elde edileceğinden bu parçaların ortalama sonuçları hesaplanarak gösterilmektedir.

Deneme 1: En iyi olmayan parametre setini ve çıktılarını ifade eder.

- **grid_search’de çalıştırılan parametre seti:**

```
param_dist = { 'n_estimators': [400, 700, 1000], 'colsample_bytree': [0.7, 0.8],  
'max_depth': [15,20,25], 'num_leaves': [50, 100, 200], 'reg_alpha': [1.1, 1.2, 1.3],  
'reg_lambda': [1.1, 1.2, 1.3], 'min_split_gain': [0.3, 0.4], 'subsample': [0.7, 0.8,  
0.9], 'subsample_freq': [20] }
```

- **Elde edilen parametre seti:**

```
{'boosting_type': 'gbdt', 'objective': 'multiclass', 'metric': 'multi_logloss',  
'num_class': 4, 'learning_rate': 0.1, 'max_depth': 15, 'num_leaves': 100,  
'colsample_bytree': 0.7, 'reg_alpha': 1.3, 'reg_lambda': 1.2, 'min_split_gain': 0.4,  
'subsample': 0.9, 'subsample_freq': 20}
```

- **Algoritma Çıktısı:**

```
Ortalama Accuracy :
92.39251695967315
Ortalama Classification Report :
              precision    recall  f1-score   support

   Benign           0.89      0.85      0.87      1005
  Likely beging      0.94      0.96      0.95      2876
Likely pathogenic    0.33      0.17      0.22         90
   Pathegenic       0.93      0.97      0.95      1195

 accuracy                   0.92      5166
 macro avg           0.77      0.73      0.75      5166
 weighted avg        0.92      0.92      0.92      5166
```

Şekil 3.5-2 LightGBM Classification Report Çıktısı

Final: En iyi parametre setini ifade eder. Bu çalışmanın çıktıları kısmında algoritma çıktısı yer almaktadır.

- **grid_search’de çalıştırılan parametre seti:**

```
param_dist = {'n_estimators': [400, 700, 1000], 'colsample_bytree': [0.7, 0.8],
'max_depth': [15,20,25], 'num_leaves': [50, 100, 200], 'reg_alpha': [1.1, 1.2, 1.3],
'reg_lambda': [1.1, 1.2, 1.3], 'min_split_gain': [0.3, 0.4], 'subsample': [0.7, 0.8, 0.9],
'subsample_freq': [20] }
```

- **Elde edilen parametre seti:**

```
{'boosting_type': 'gbdt', 'objective': 'multiclass', 'metric': 'multi_logloss', 'num_class':
4, 'class_weight': 'None', 'colsample_bytree': 0.7, 'importance_type': 'split',
'learning_rate': 0.1, 'max_depth': 15, 'min_child_samples': 20, 'min_child_weight':
0.001, 'min_split_gain': 0.4, 'n_jobs': -1, 'num_leaves': 50, 'reg_alpha': 1.2,
'reg_lambda': 1.1, 'silent': 'False', 'subsample': 0.8, 'subsample_for_bin': 200000,
'subsample_freq': 20}
```


3.5.3. CatBoost

Algoritmanın parametre optimizasyonu için **grid_searchCV** algoritması 5 parçalı çapraz doğrulama ile birden fazla parametre seti için çalıştırılmış, optimize edilmiş parametreler ve bu parametreler için çıktılar aşağıda yer almaktadır. Çapraz doğrulamada her bir parça için ayrı bir çıktı elde edileceğinden bu parçaların ortalama sonuçları hesaplanarak gösterilmektedir.

- **grid_search'de çalıştırılan parametre seti:**

```
params = {'depth': [4, 7, 10], 'learning_rate': [0.03, 0.1, 0.15], 'l2_leaf_reg': [1,4,9],  
'iterations': [300] }
```

- **Elde edilen parametre seti:**

```
{'eval_metric'="AUC", 'depth'=10, 'iterations'= 500, 'l2_leaf_reg'= 9,  
'learning_rate'=0.15}
```

- **Algoritma Çıktısı:**

```
average accuracy: 0.9202467508393235  
average classification report:  
              precision    recall  f1-score   support  
  
    Benign           0.88      0.85      0.86       1005  
  Likely beging       0.94      0.95      0.95       2876  
Likely pathogenic     0.25      0.08      0.12         90  
    Pathegenic        0.92      0.97      0.95       1195  
  
   accuracy                   0.92       5166  
  macro avg           0.75      0.71      0.72       5166  
weighted avg           0.91      0.92      0.92       5166
```

Şekil 3.5-3 CatBoost Classification Report Çıktısı

Final: En iyi parametre setini ifade eder. Bu çalışmanın çıktıları kısmında algoritma çıktısı yer almaktadır.

- **grid_search’de çalıştırılan parametre seti:**

```
param_dist = {'depth': [4, 7, 10], 'learning_rate': [0.01, 0.1, 0.2], 'l2_leaf_reg': [1, 4, 9],  
'iterations': [200], 'bagging_temperature': [0.3, 1, 10], 'grow_policy': ['SymmetricTree',  
'Depthwise', 'Lossguide'] }
```

- **Elde edilen parametre seti:**

```
{'bagging_temperature': 0.3, 'depth': 7, 'grow_policy': 'Lossguide', 'iterations': 200,  
'l2_leaf_reg': 1, 'learning_rate': 0.1}
```

4. ÇIKTILAR

Bu bölümde kullanılan algoritmalara ait optimum parametre seti için elde edilen nihai çıktıları yer verilmiştir.

Daha önce de belirtildiği gibi! Algoritma çıktılarında “**Likely Pathogenic**” sınıfının diğer sınıflara oranla **düşük olan skorların sebebi**, bu sınıfa ait veri sayısının diğerlerine kıyasla oldukça **az oluşudur**. Makine öğrenmesi modelleri eğitildiği verinin sayısı arttıkça başarılarını artırmaktadır.

variants_encoded_only_VUS.csv: Eğitim verisinin dışında tutulan ve “**rev.clinical_significance**” sütununa ait değerlerin tamamının “Uncertain Significance” yani bilinmeyen klinik satüsü olan **6761** satır verinin kategorikten numerik dönüşümü **yapılarak** saklandığı dosyadır. Eğitim için kullanılan verilerle aynı 74 sütunu da içerir.

variants_not_encoded_only_VUS.csv: Eğitim verisinin dışında tutulan ve “**rev.clinical_significance**” sütununa ait değerlerin tamamının “Uncertain Significance” yani bilinmeyen klinik satüsü olan **6761** satır verinin kategorikten numerik dönüşümü **yapılmadan** saklandığı dosyadır. Eğitim için kullanılan verilerle aynı 74 sütunu da içerir.

Feature Importance: Bu bölümde Feature Importance başlığı altında göreceğiniz grafikler: İlgili algoritmaların modellerinin kurulmasında algoritma tarafından sınıflandırma işlemi için en ayırt edici, bir başka deyişle en önemli görülen özellikleri ifade ederek bunların önem derecelerini katsayılandırır. Bu bölümde yer alan grafiklerde ilgili algoritmanın en değerli gördüğü **ilk 10** özelliğe ve skorlarına yer verilmiştir.

VUS Çıktısı: Bu bölümde VUS Çıktısı başlığı altında göreceğiniz grafikler aşağıdaki adımlarla oluşturulur:

- İlgili algoritmaya ait model belirtilen parametrelerle kurulur

- Önceden belirtilen ve eğitim için ayrılmış verilerle Makine Öğrenmesi modeli eğitilir.
- Eğitilen modelin performansı **Algoritma Çıktısı** başlığı altında sunulur.
- Elde edilen model **variants_encoded_only_VUS.csv** dosyası içerisinde bulunan ve patojenitesi bilinmeyen veriler üzerinde patojenite tahminlerinde bulunur.
- Algoritmanın tahminlerinin sonuçları alınarak, tahminlerin patojenite sınıflarına göre yüzdelik dağılımları grafik haline getirilerek sunulur.

Bu bölümde yer alan çıktılarına ait tüm dosyalara, bu çalışma kapsamında patojenitesi tahmin edilen **VUS** varyantlarına dair tahminlere ve daha fazlasına çalışmanın GitHub Deposu[32] üzerinden erişebilirsiniz.

4.1. XGBoost:

- **Algoritma Çıktısı:**

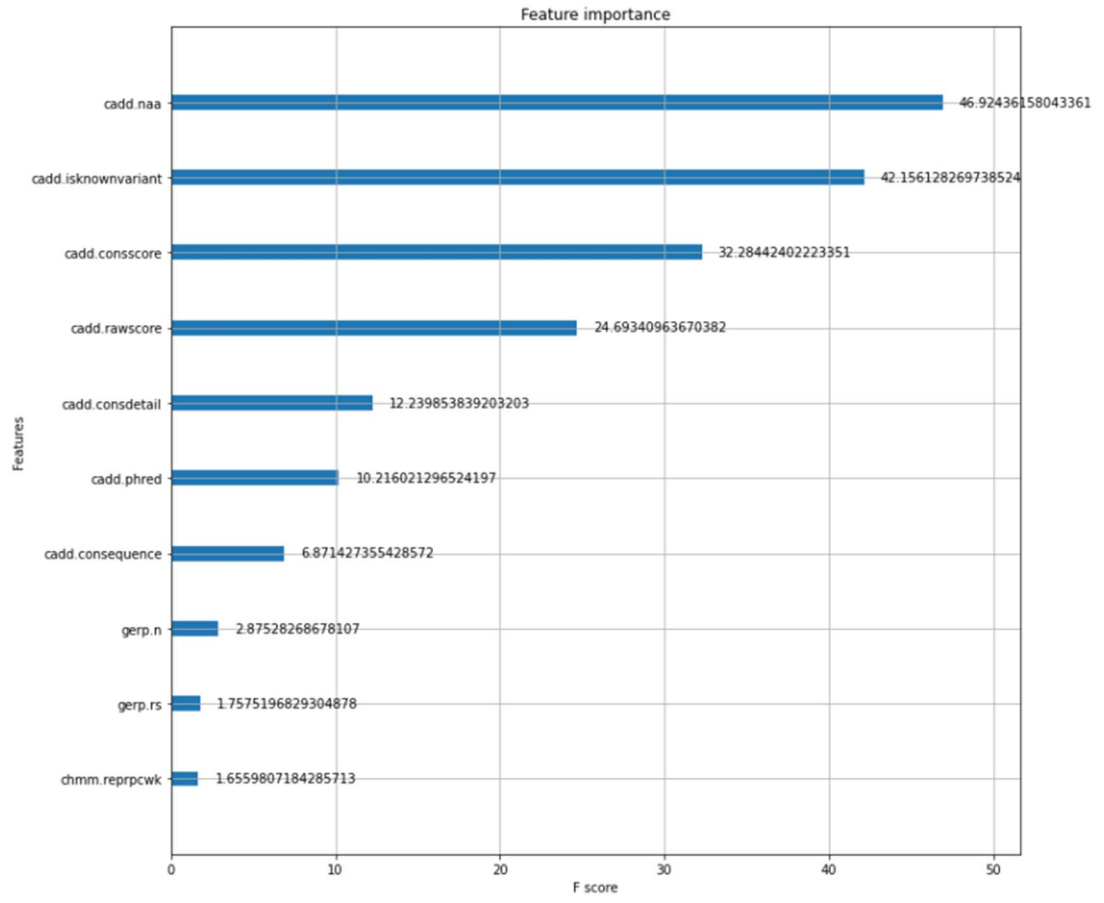
```
Ortalama Accuracy :
91.77307461132717
Ortalama Classification Report :
              precision    recall  f1-score   support

   Benign              0.84      0.88      0.86       962
  Likely benign         0.95      0.94      0.94      2921
Likely pathogenic       0.18      0.31      0.23        51
   Pathogenic          0.96      0.93      0.94      1232

   accuracy              0.92              5166
  macro avg              0.73      0.76      0.74      5166
 weighted avg              0.92      0.92      0.92      5166
```

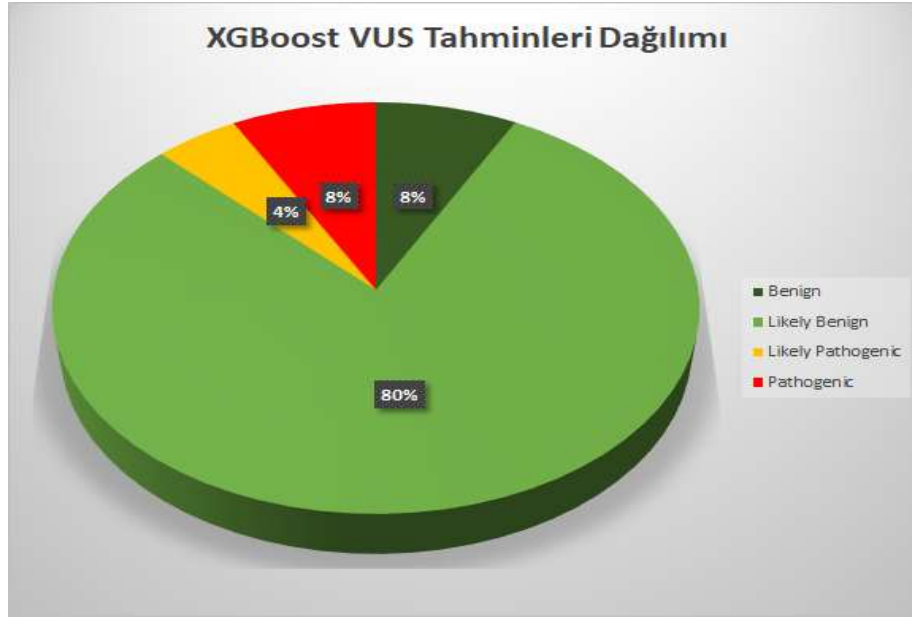
Şekil 4.1-1 XGBoost Classification Report Çıktısı - Final

Feature Importance:



Şekil 4.1-2 XGBoost Feature Importance - Final

- **Vus Çıktısı:**



Şekil 4.1-3 XGBoost'un VUS varyantlar üzerine tahmininin sınıflara göre dağılımı

4.2. LightGBM

- **Algoritma Çıktısı:**

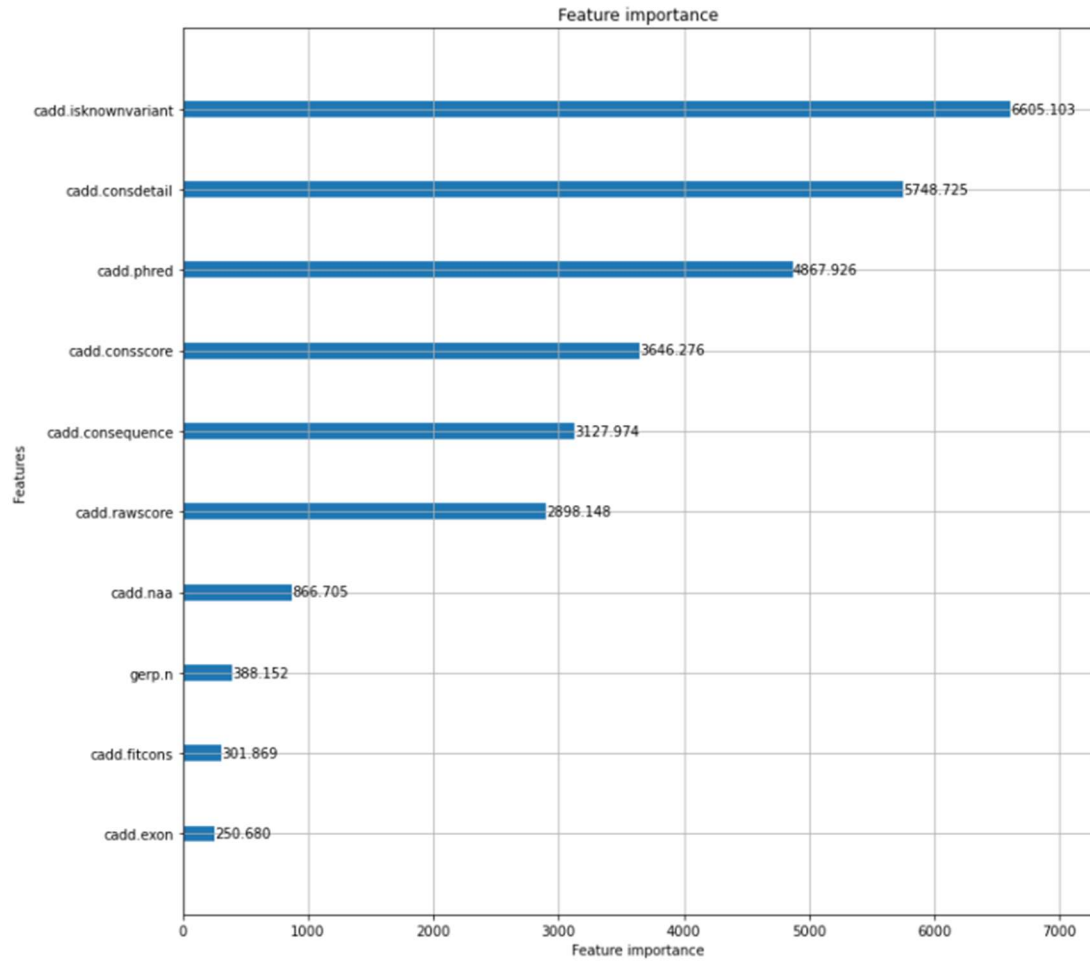
Ortalama Accuracy :
92.33432136029406

Ortalama Classification Report :

	precision	recall	f1-score	support
Benign	0.88	0.84	0.86	1005
Likely beging	0.94	0.96	0.95	2876
Likely pathogenic	0.35	0.18	0.24	90
Pathegenic	0.94	0.97	0.95	1195
accuracy			0.92	5166
macro avg	0.78	0.74	0.75	5166
weighted avg	0.92	0.92	0.92	5166

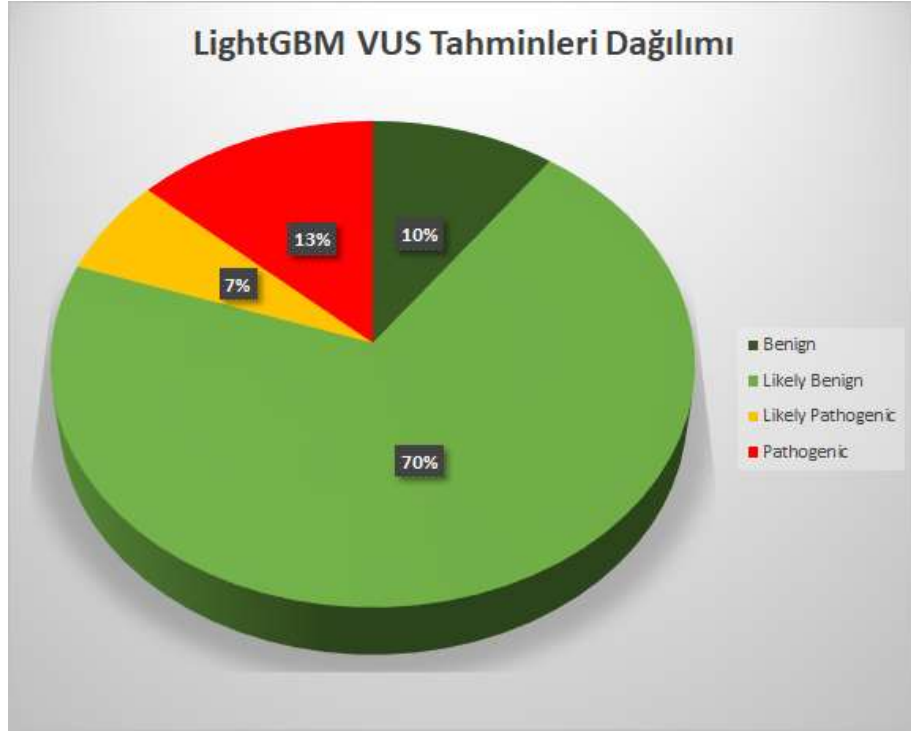
Şekil 4.2-1 LightGBM Classification Report Çıktısı - Final

- **Feature Importance:**



Şekil 4.2-2 LightGBM Feature Importance - Final

- **Vus Çıktısı:**



Şekil 4.2-3 LightGBM'in VUS varyantlar üzerine tahmininin sınıflara göre dağılımı

4.3. CatBoost:

- **Algoritma Çıktısı:**

```

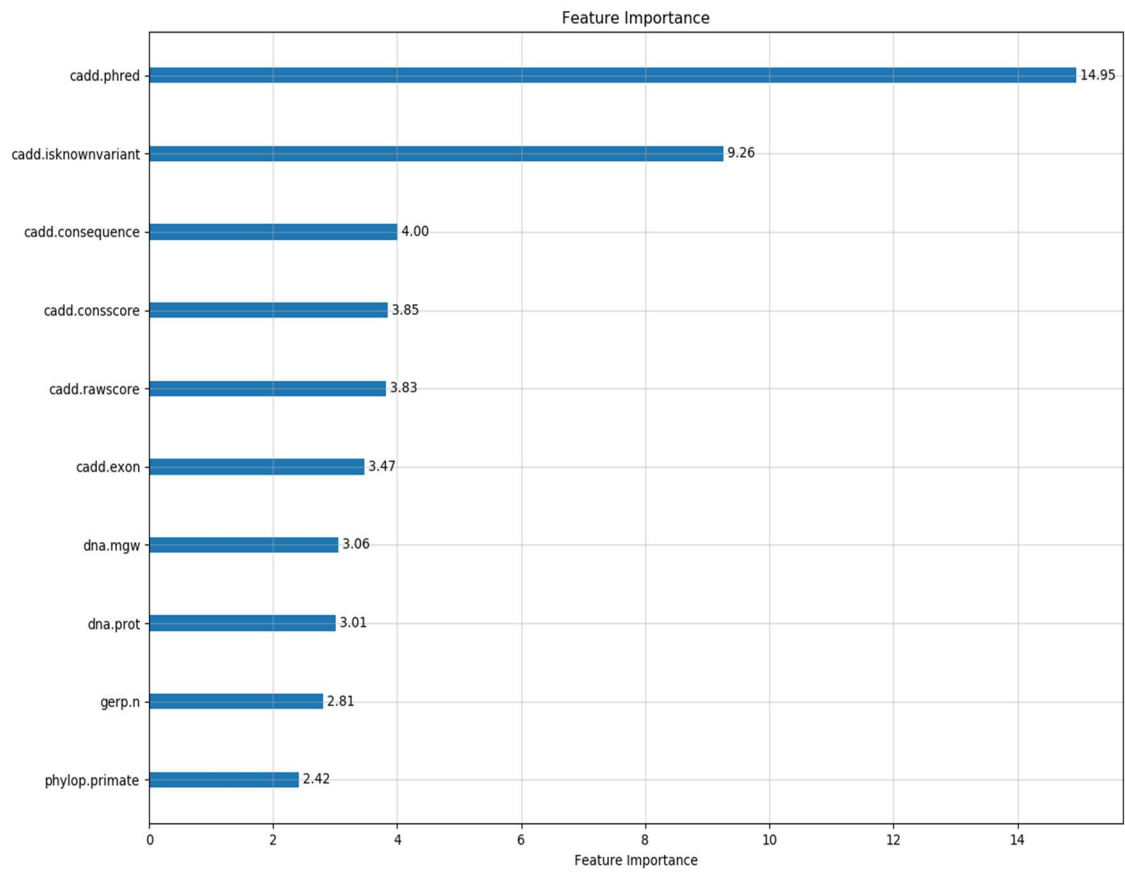
average accuracy: 0.9225693319676965
average classification report:
              precision    recall  f1-score   support

   Benign           0.89      0.85      0.87       1005
  Likely beging      0.94      0.96      0.95       2876
Likely pathogenic     0.29      0.09      0.14         90
   Pathegenic        0.92      0.97      0.95       1195

   accuracy              0.92       5166
  macro avg           0.76      0.72      0.72       5166
 weighted avg           0.91      0.92      0.92       5166
  
```

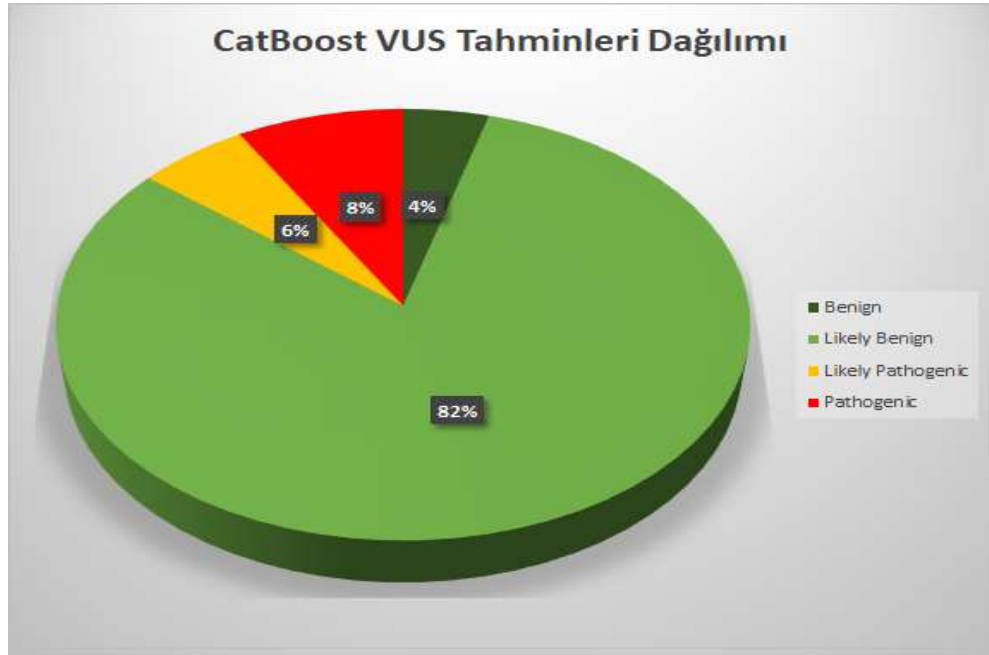
Şekil 4.3-1 CatBoost Classification Report Çıktısı - Final

- **Feature Importance:**



Şekil 4.3-2 CatBoost Feature Importance – Final

- **Vus Çıktısı:**



Şekil 4.3-3 CatBoost'un VUS varyantlar üzerine tahmininin sınıflara göre dağılımı

4.4. Random Forest Classifier:

Ortalama Accuracy :
91.40536380675616

Ortalama Classification Report :

	precision	recall	f1-score	support
Benign	0.85	0.84	0.85	1005
Likely benign	0.94	0.94	0.94	2876
Likely pathogenic	0.38	0.23	0.29	90
Pathogenic	0.94	0.95	0.95	1195
accuracy			0.91	5166
macro avg	0.77	0.74	0.75	5166
weighted avg	0.91	0.91	0.91	5166

Şekil 4.4-1 Random Forest Classification Report Çıktısı - Final

4.5. Logistic Regression:

```
Ortalama Accuracy :
91.9859341910381
Ortalama Classification Report :
              precision    recall  f1-score   support

   Benign           0.88      0.86      0.87       1005
  Likely benign      0.94      0.95      0.95       2876
Likely pathogenic    0.30      0.12      0.17         90
   Pathogenic       0.92      0.96      0.94       1195

 accuracy              0.92       5166
 macro avg            0.76      0.72      0.73       5166
 weighted avg         0.91      0.92      0.92       5166
```

Şekil 4.5-1 Logistic Regression Classification Report Çıktısı - Final

4.6. VUS Varyantların Patojenite Tahminleri

Bu bölümde çalışmanın başında da belirtildiği gibi, başarılı bulunan Makine Öğrenmesi modellerinin klinik statüsü yani patojenitesi bilinmeyen varyantlar için patojenite tahminlerine yer verilmiştir. Bu varyantlar bu çalışmanın **Çıktılar** kısmında da bahsedildiği gibi **variants_encoded_only_VUS.csv** ve **variants_not_encoded_only_VUS.csv** dosyalarında tutulan 6170 adet varyanttan oluşmaktadır. Seçilen en başarılı 5 algoritmaya (XGBoost, LighGBM, CatBoost, Random Forest Classifier, Logistic Regression) ait çıktılar , bir araya getirilerek tablolanmış ve bu tablo üzerinde her bir varyant için 5 algoritmanın oylaması ile ortak karara varılan **majority voting** algoritması uygulanmıştır. Elde edilen patojenite sınıfı bilgisi ilgili varyant için nihai sonucu oluşturmuştur. Sonuçların boyutları gereği tamamına tez içerisinde yer verilememiş, aşağıda çıktıya dair bir örnek sunulmuştur. 6170 adet varyant için tüm tahminlerimize çalışmaya ait **GitHub deposu [32]** üzerinden erişilebilmektedir.

_id	LightGBM	XGBoost	CatBoost	RFC	LRegression	Prediction
chr17:g.41234524T >A	L. benign	L. benign	L. benign	L. benign	L. benign	L. benign
chr17:g.41215945T >G	Pathogen ic	Pathogen ic	L. pathogen ic	Pathogen ic	L. pathogen ic	Pathogen ic
chr17:g.41209112T >C	L. benign	L. benign	L. benign	L. benign	L. benign	L. benign
chr17:g.41223028C >T	L. benign	L. benign	L. benign	L. benign	L. benign	L. benign
chr17:g.41215923 A>G	Pathogen ic	Pathogen ic	L. pathogen ic	Pathogen ic	Pathogen ic	Pathogen ic
chr13:g.32910774 A>G	L. benign	L. benign	L. benign	L. benign	L. benign	L. benign
chr13:g.32906768 A>C	L. benign	L. benign	L. benign	L. benign	L. benign	L. benign
chr13:g.32906484T >G	L. benign	L. benign	L. benign	L. benign	L. benign	L. benign
chr13:g.32907158 A>G	L. benign	L. benign	L. benign	L. benign	L. benign	L. benign
chr17:g.41243748 A>G	L. benign	L. benign	L. benign	L. benign	L. benign	L. benign

5. SONUÇLAR VE ÖNERİLER

Bu çalışmada BRCA1 ve BRCA2'e ait genetik [1] varyantların patojenitesini tahmin etmek amacıyla Makine Öğrenmesi ile Sınıflandırma yöntemlerini kullandık. Verileri CADD veri tabanından myvariant.info web API ile elde ettik. Python dilinde scikit-learn kütüphanesini kullanarak 3 ana adımda ön işlemlerini gerçekleştirdik. Bu ön işlemler sonucu uygulamada kullanmak üzere 4 farklı dosya ürettik. Daha sonra Boosted Tree algoritmalarından XGBoost, LighGBM ve CatBoost; Yaygın kullanılan diğer sınıflandırma algoritmalarından Lojistik Regresyon, KNN, SVM, Decision Tree Classifier Naive Bayes, ve Random Forest Classifier gibi farklı algoritmalar için parametre optimizasyonları yapıp, oluşturduğumuz veriler üzerinden Makine Öğrenmesi Modelleri eğittik. Tüm veri üzerinden çarpaz doğrulama ile Accuracy metriği üzerinde 93%'e kadar başarılı olduk, yine sınıflara ait F-1 metriği üzerinde Likely Pathogenic sınıfı için 34%'e kadar, kalan 3 sınıf için 95%'e kadar başarılı olduk. Likely Pathogenic sınıfında elde edilen düşük başarının sebebi bu sınıfa ait verilerin diğer sınıflara oranlar çok daha az örnek içermesi ve modellerin yeterince öğrenme iterasyonunu gerçekleştirememesi oldu, Likely Pathogenic sınıfı yalnızca 90 adet veriyle kendisinden bir sonraki en az veri içeren Benign sınıfının sadece 8.9%'u kadar veri içerir. Burada verideki aşırı dengesizliği gidermek için downsampling-upsampling adı verilen dengelem metotları verilerin örnek sayılarını dengelenebilir ve doğruluğa etkisi incelenebilir.

Bahsedilen 3 Boosted Tree algoritması için Feature Importance grafikleri çizdirerek her algoritma için sınıflandırma yaparken en anlamlı gördüğü 10 parametreye ve bunların önem katsayılarına vurgu yaptık. Buradan elde ettiğimiz sonuçlarda 6 adet özelliğin her 3 algoritmanın da en değerli 10 özellik listesinde ortak olarak yer aldığını tespit ettik bunlar: cadd.isknownvariant, cadd.consscore, cadd.rawscore, cadd.phred, cadd.consequence, gerp.n'dir.

Bahsedilen 3 Boosted Tree algoritması için Feature Importance grafikleri çizdirerek her algoritma için sınıflandırma yaparken en anlamlı gördüğü 10 parametreye ve bunların önem katsayılarına vurgu yaptık. Buradan elde ettiğimiz sonuçlarda 6 adet özelliğin her 3 algoritmanın da en değerli 10 özellik listesinde ortak olarak yer aldığını tespit ettik bunlar: cadd.isknownvariant, cadd.consscore, cadd.rawscore, cadd.phred, cadd.consequence, gerp.n'dir.

Bu bilgiler ışığında, bizce 6/10 kıymetli bir eşleşme oranı olup bahsedilen özellikler üzerinden geriye doğru gidilerek. Bu özelliklerin sınıflandırma ayrımında neden bu denli kıymetli olduğunun ve patojeniteye etkisinin daha detaylı olarak hem biyoenformatik hem de genetik alanından akademisyenlerce ortak olarak incelenmesinde fayda vardır.

Eğitilen ve performansı sunulan modellerin çalışmanın başında da belirtildiği üzere model tahminlerinin yapılması için hazırlanıp ayrılan klinik statüsü VUS yani patojenitesi bilinmeyen 6761 satır veri oluşturulmuştur. Bu 6761 VUS varyantına ait patojenite sınıfı tahminlerimiz gerçekleştirilmiş ve çıktılar kısmında bu tahminlerin sınıflara göre yüzdelik dağılımları sunulmuştur.

Daha sonra sınıflandırma performanslarını göz önünde bulundurarak seçtiğimiz 5 algoritmanın modellerine ait VUS varyantlar üzerindeki patojenite tahminlerini bir araya getirilerek, oylama sistemi olarak da adlandırılan majority voting algoritması ile her bir varyant için 5 algoritmanın oylarıyla belirlenen, bu çalışmanın kapsamı içerisindeki nihai patojenite tahminlerimiz elde edilmiştir. Bu sonuçların bir örneği tezde yer alıp boyut kısıtları sebebiyle devamı dijital ortamda muhafaza edilmiştir. Çalışmanın tüm kaynak, dosya ve dokümanlarının da bulunduğu GitHub deposu üzerinden erişilebilmektedir.

KAYNAKÇA

- [1] Adam Felman(2019), UNDERSTANDING BREAST CANCER, <https://www.medicalnewstoday.com/articles/37136>, SON ERİŞİM TARİHİ: 20/06/2020
- [2] <http://memeder.org>, SON ERİŞİM TARİHİ: 20/06/2020
- [3] <https://www.genome.gov>, , SON ERİŞİM TARİHİ: 20/06/2020
- [4] https://hsgm.saglik.gov.tr/depo/birimler/kanser-db/istatistik/2014-rapor._uzuuun.pdf, , SON ERİŞİM TARİHİ: 20/06/2020
- [5] <https://www.ncbi.nlm.nih.gov/grc/human>, , SON ERİŞİM TARİHİ: 20/06/2020
- [6] <https://ghr.nlm.nih.gov/primer/basics/dna>, , SON ERİŞİM TARİHİ: 20/06/2020
- [7] <https://www.nationalbreastcancer.org/what-is-brca>, , SON ERİŞİM TARİHİ: 20/06/2020
- [8] <https://genos.co/resources/variant.html>, , SON ERİŞİM TARİHİ: 20/06/2020
- [9] <https://cadd.gs.washington.edu/info>, SON ERİŞİM TARİHİ: 20/06/2020
- [10] <https://myvariant.info/about>, SON ERİŞİM TARİHİ: 20/06/2020
- [11] <https://expertsystem.com/machine-learning-definition/>, SON ERİŞİM TARİHİ: 20/06/2020
- [12] <https://www.ibm.com/topics/machine-learning>, SON ERİŞİM TARİHİ: 20/06/2020
- [13] <https://www.edureka.co/blog/classification-in-machine-learning/>, SON ERİŞİM TARİHİ: 20/06/2020
- [14] <https://machinelearningmastery.com/gradient-boosting-with-scikit-learn-xgboost-lightgbm-and-catboost/>, SON ERİŞİM TARİHİ: 20/06/2020
- [15] <https://medium.com/mlreview/gradient-boosting-from-scratch-1e317ae4587d>, SON ERİŞİM TARİHİ: 20/06/2020
- [16] <https://xgboost.readthedocs.io/en/latest/>, SON ERİŞİM TARİHİ: 20/06/2020
- [17] Shubham Malik(2020), XGBoost: A Deep Dive Into Boosting

- [18] Chen ve Guestrin(2016), XGBoost: A Scalable Tree Boosting System, <https://www.kdd.org/kdd2016/papers/files/rfp0697-chenAemb.pdf>, SON ERİŞİM TARİHİ: 20/06/2020
- [19] Labram(2019), Fitting data with XGBoost, <https://www.actuaries.org.uk/news-and-insights/news/article-fitting-data-xgboost>, SON ERİŞİM TARİHİ: 20/06/2020
- [20] Guolin Ke, Qi Meng(2017), LightGBM: A Highly Efficient Gradient Boosting Decision Tree, <https://papers.nips.cc/paper/6907-lightgbm-a-highly-efficient-gradient-boosting-decision-tree.pdf>, SON ERİŞİM TARİHİ: 20/06/2020
- [21] <https://medium.com/@pushkarmandot/https-medium-com-pushkarmandot-what-is-lightgbm-how-to-implement-it-how-to-fine-tune-the-parameters-60347819b7fc>, SON ERİŞİM TARİHİ: 20/06/2020
- [22] PRANJAL KHANDELWAL(2017), Which algorithm takes the crown: Light GBM vs XGBOOST?, <https://www.analyticsvidhya.com/blog/2017/06/which-algorithm-takes-the-crown-light-gbm-vs-xgboost/>, SON ERİŞİM TARİHİ: 20/06/2020
- [23] <https://www.analyticsvidhya.com/blog/2017/08/catboost-automated-categorical-data/>, SON ERİŞİM TARİHİ: 20/06/2020
- [24] SUNIL RAY(2017), CatBoost: A machine learning library to handle categorical (CAT) data automatically, <https://towardsdatascience.com/https-medium-com-talperetz24-mastering-the-new-generation-of-gradient-boosting-db04062a7ea2>, SON ERİŞİM TARİHİ: 20/06/2020
- [25] Alwira Swalin(2018), CatBoost vs. Light GBM vs. XGBoost, <https://towardsdatascience.com/catboost-vs-light-gbm-vs-xgboost-5f93620723db>, SON ERİŞİM TARİHİ: 20/06/2020
- [26] <https://medium.com/@hanishsidhu/whats-so-special-about-catboost-335d64d754ae>, SON ERİŞİM TARİHİ: 20/06/2020
- [27] http://78.189.53.61/-/bs/ess/k_sumbuloglu.pdf, SON ERİŞİM TARİHİ: 20/06/2020
- [28] <https://www.bilkav.com/makine-ogrenmesi-egitimi/>, SON ERİŞİM TARİHİ: 20/06/2020
- [29] Tony Yiu(2019), Understanding Random Forest, <https://towardsdatascience.com/understanding-random-forest-58381e0602d2>, SON ERİŞİM TARİHİ: 20/06/2020

[30] https://www.tutorialspoint.com/machine_learning_with_python/machine_learning_with_python_classification_algorithms_random_forest.htm, SON ERİŞİM TARİHİ: 20/06/2020

[31] RENTZSCH P, WITTEN D, COOPER GM, SHENDURE J, KIRCHER M.(2018),CADD: PREDICTING THE DELETERIOUSNESS OF VARIANTS THROUGHOUT THE HUMAN GENOME. NUCLEIC ACIDS RES. OCT 29. DOI: [10.1093/nar/gky1016](https://doi.org/10.1093/nar/gky1016) . PUBMED PMID: 30371827, SON ERİŞİM TARİHİ: 20/06/2020

[32] <https://github.com/EmreKARAgH/AnalyseBRCAGeneVariants>, Proje Dosyaları ve kodları içeren GitHub deposu. SON ERİŞİM TARİHİ: 20/06/2020

ÖZGEÇMİŞ

Onur Kaplan

26 Şubat 1998 tarihinde Zonguldak'ta doğdu. İlköğretim ve lise eğitimini Karabük'te tamamladı. 2016 yılında Safranbolu Anadolu Öğretmen Lisesi'nden mezun oldu. 2016 yılında eğitime başladığı Kocaeli Üniversitesi Bilgisayar Mühendisliği'nde öğrenimine devam etmektedir.

onurkaplan1907@gmail.com

Muhammed Emre Kara

27 Mayıs 1998 tarihinde Adıyaman'da doğdu. İlköğretim ve lise eğitimini Adıyaman'da tamamladı. 2016 yılında Adıyaman Anadolu Öğretmen Lisesi'nden mezun oldu. 2016 yılında eğitime başladığı Kocaeli Üniversitesi Bilgisayar Mühendisliği'nde öğrenimine devam etmektedir.

mailemrek@gmail.com