T.C
SAKARYA ÜNİVERSİTESİ

Bilgisayar ve Bilişim Mühendisliği

BSM617- VERİ MADENCİLİĞİ UYGULAMALARI

Rapor konusu:

MAKINE ÖGRENME ALGORITMALARI KULLANILARAK TITANIK YOLCULARIN HAYATTA KALMA ANALIZI

Yapan:

Y195012015 - Ibrahima BARRY

Özet

Titanik felaketi 100 yıl önce 15 Nisan 1912'de meydana geldi ve yaklaşık 1500 yolcu ve mürettebat öldü. Önemli olay, araştırmacıları ve analistleri hala bazı yolcuların hayatta kalmasına ve diğerlerinin ölümüne neyin yol açmış olabileceğini anlamaya zorluyor. Makine öğrenme yöntemlerinin kullanılması ve eğitim setinde 891 satır ve test setinde 418 satırdan oluşan bir veri setinin kullanılmasıyla araştırma, faktörler arasındaki ilişkiyi (yaş, cinsiyet, yolcu sınıfı, ücret vb.) Belirlemeye çalışmaktadır. yolcuların hayatta kalma şansına. Bu faktörler yolcuların hayatta kalma oranlarını etkileyebilir veya etkilemeyebilir. Bu araştırma makalesinde, yolcuların hayatta kalmasını tahmin etmek için Lojistik Regresyon, Naif Bayes, Karar Ağacı, Rastgele Orman gibi çeşitli makine öğrenme algoritmaları uygulanmıştır. Özellikle, bu araştırma çalışması, algoritmayı bir test veri kümesindeki doğruluk yüzdesi temelinde karşılaştırır.

I. Giriş

Makine öğrenimi alanı, analistlerin geçmiş verilerden ve geçmiş olaylardan içgörüleri ortaya çıkarmasına izin verdi. Titanik felaketi, dünya tarihinin en ünlü gemi enkazlarından biridir. Titanic, bir buzdağına çarptıktan birkaç saat sonra Kuzey Atlantik Okyanusu'nda Batan bir İngiliz gemisidir. Geminin kırılması olayının nedenini destekleyecek gerçekler olsa da, Titanic felaketinde yolcuların hayatta kalma oranıyla ilgili çeşitli spekülasyonlar var. Yıllar geçtikçe, hayatta kalanların yanı sıra ölen yolcuların verileri de toplandı. Veri kümesi, Kaggle.com adlı bir web sitesinde halka açıktır.

Bu veri kümesi, çeşitli makine öğrenme algoritmaları (Random Forest, SVM vb.) Kullanılarak incelenmiş ve analiz edilmiştir. Araştırmanın temel amacı, çeşitli makine öğrenme algoritmaları kullanarak yolcuların hayatta kalması ile yolcuların özellikleri arasındaki ilişkiyi belirlemek için Titanic felaketini analiz etmektir. Özellikle, bu araştırma çalışması, algoritmaları bir test veri setindeki doğruluk yüzdesi temelinde karşılaştırır.

II. Veriset

Makale için kullanılan veri kümesi Kaggle web sitesi tarafından sağlanmaktadır. Veriler, ilgili etiketlerle birlikte bir yolcu örneği olan tren setinde 891 satırdan oluşmaktadır. Her yolcu için yolcunun adı, cinsiyeti, yaşı, yolcu sınıfı, gemideki kardeş veya eş sayısı, gemideki ebeveyn veya çocuk sayısı, kabin, bilet numarası, bilet ücreti ve biniş bilgileri verildi. Veriler bir CSV (Virgülle Ayrılmış Değer) dosyası biçimindedir. Test verileri için, web sitesi aynı CSV formatında 418 yolcu örneği sağladı. Veri setinin bir örnek sıralı yapısı Tablo I'de listelenmiştir. Eğitim setinin nitelikleri ve açıklamaları Tablo II'de verilmiştir.

Bir model oluşturmadan önce, tahmin için sınıflandırıcı oluştururken tüm faktörlerin veya özniteliklerin nelerin faydalı olabileceğini belirlemek için veri araştırması yapılır. Araştırmaya başlamak için, her özellik için genel bir fikir edinmek için birkaç X-Y jenerik grafiği yapılır. Genel grafiklerden bazıları aşağıda gösterilmiştir. Şekil 2'deki yaş grafiği, yolcuların maksimum veya büyük çoğunluğunun 20-40 yaş grubuna ait olduğunu göstermektedir.

Benzer şekilde, Şekil 3'de bir grafik çizilmiştir ve cinsiyet özelliği için bazı hesaplamalar yapılmıştır ve sonuçlar, kadının hayatta kalma oranının, erkeğinkinden % 25.67 daha yüksek olduğunu göstermektedir. Benzer şekilde, özniteliklerin her biri, daha sonra tahmin için kullanılacak olan öznitelikleri veya özellikleri çıkarmak için araştırılır.

	survived	pclass	sex	age	sibsp	parch	fare	embarked	class	who	adult_male	deck	embark_town	alive	alone
0	0	3	male	22.0	1	0	7.2500	S	Third	man	True	NaN	Southampton	no	False
1	1	1	female	38.0	1	0	71.2833	C	First	woman	False	C	Cherbourg	yes	False
2	1	3	female	26.0	0	0	7.9250	S	Third	woman	False	NaN	Southampton	yes	True
3	1	1	female	35.0	1	0	53.1000	S	First	woman	False	C	Southampton	yes	False
4	0	3	male	35.0	0	0	8.0500	S	Third	man	True	NaN	Southampton	no	True
5	0	3	male	NaN	0	0	8.4583	Q	Third	man	True	NaN	Queenstown	no	True
6	0	1	male	54.0	0	0	51.8625	S	First	man	True	E	Southampton	no	True
7	0	3	male	2.0	3	1	21.0750	S	Third	child	False	NaN	Southampton	no	False
8	1	3	female	27.0	0	2	11.1333	S	Third	woman	False	NaN	Southampton	yes	False
9	1	2	female	14.0	1	0	30.0708	C	Second	child	False	NaN	Cherbourg	yes	False

1. Tablo: Kaggle veri seti

Attributes	Açıklama
PassengerID	Yolcuların kimlik numarası.
Pclass/Class	Yolcuların sınıfı (1/First, 2/Second veya 3/Third)
Name	Yolcuların adı
Sex	Yolcuların cinsiyeti (erkek veya kadın)
Age	Yolcuların yaşı
SibSp	Gemideki kardeş veya eş sayısı
Parch	Gemideki ebeveyn veya çocuk sayısı
Ticket	Bilet numarası
Fare	Biletin fiyatı
Cabin	Yolcunun kabin numarası
Embark_town	Gemiye biniş limanı (Cherbourg, Queenstown veya Southampton)
Survived/Alive	Hedef değişken (yok olan için 0/No ve hayatta kalan için 1/Yes değerleri)
alone	Yalnız olup olmadığını belirlenir
who	Yolculuğun kim olduğunu belirlenir (değerleri: erkek, kadın veya çocuk)
adult_male	Yolcunun aynı anda yetişkin ve erkek olup olmadığını bilmeyi sağlar (değerlerı: doğru=true veya yanlış=false)
deck	her yolcu kabininin hangi güvertede olduğu hakkında bilgidir

2. Tablo: antrenman veri kümesindeki özellikleri.

Korelasyon analizi:

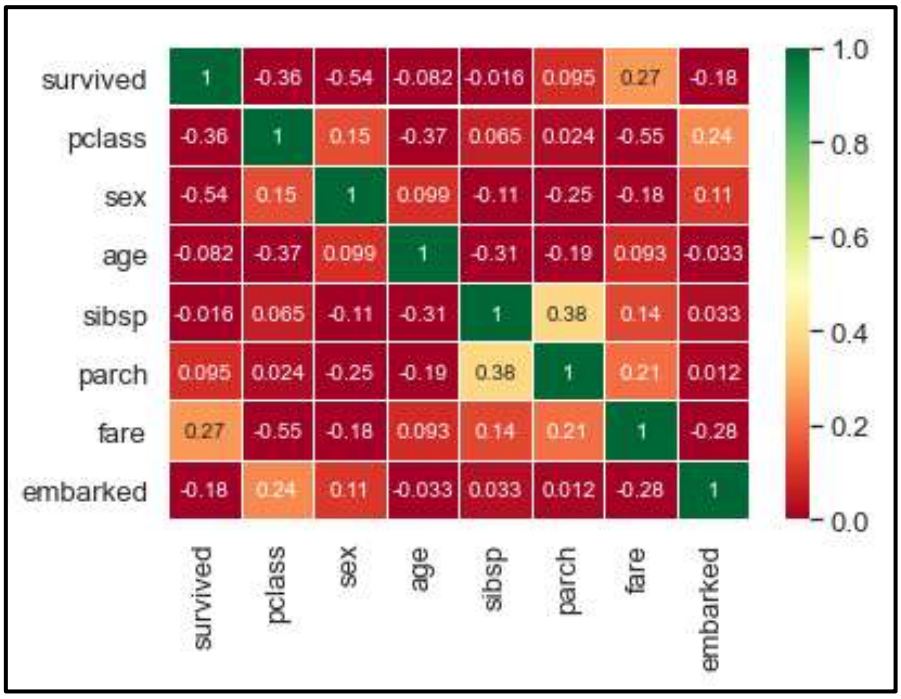
Bir korelasyon matrisi, belirli bir verilerdeki değişken çiftleri arasındaki "korelasyonları" temsil eden tablo verileridir. Farklı değişkenlerimiz arasındaki ilişkileri görselleştirmemizi sağlamaktadır.

Bir korelasyon matrisinin yorumlanması şu şekilde yapılımaktadır:

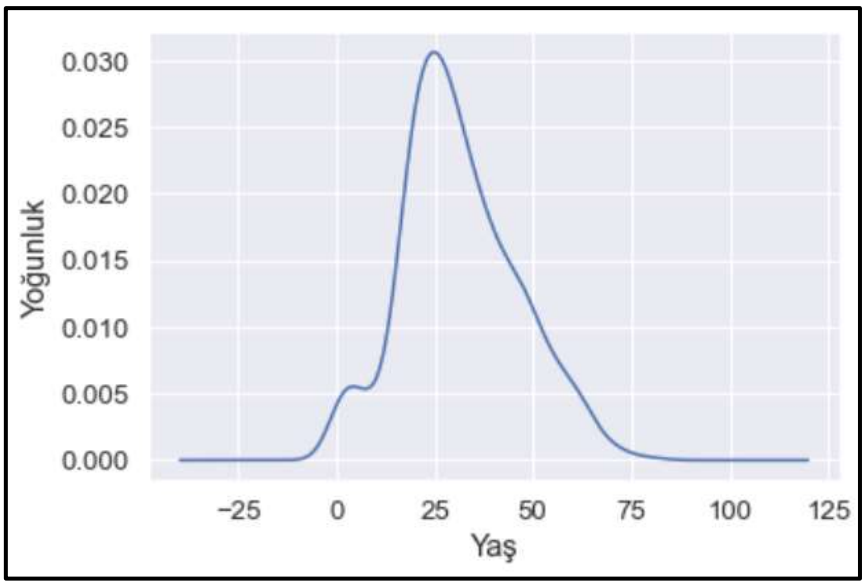
- Pozitif 1 (-1.000), iki değişken arasında mükemmel, pozitif bir korelasyon olduğu anlamına gelir, bir özellik artarken diğeri tam orantılı olarak artar. Negatif bir sayı varsa
- Sıfır 1 (—0.000), iki değişken arasında pozitif veya negatif bir korelasyon olmadığı anlamına gelir. Tamamen rastgele hareket ederler.
- Negatif 1 (–1.000), iki değişken arasında mükemmel, negatif veya ters bir korelasyon olduğu anlamına gelir. Bir özellik yükselirken diğeri aşağı iner ve bunun tersi de geçerlidir.

Bu çalışmada korelasyon analizini, kullanacak parametrelerini tespit etmek için, kullanılmaktadır. Şekil 1'de survived, pclass, sex, age, sibsp, parch, fare ve embarked parametreler arasında korelasyon

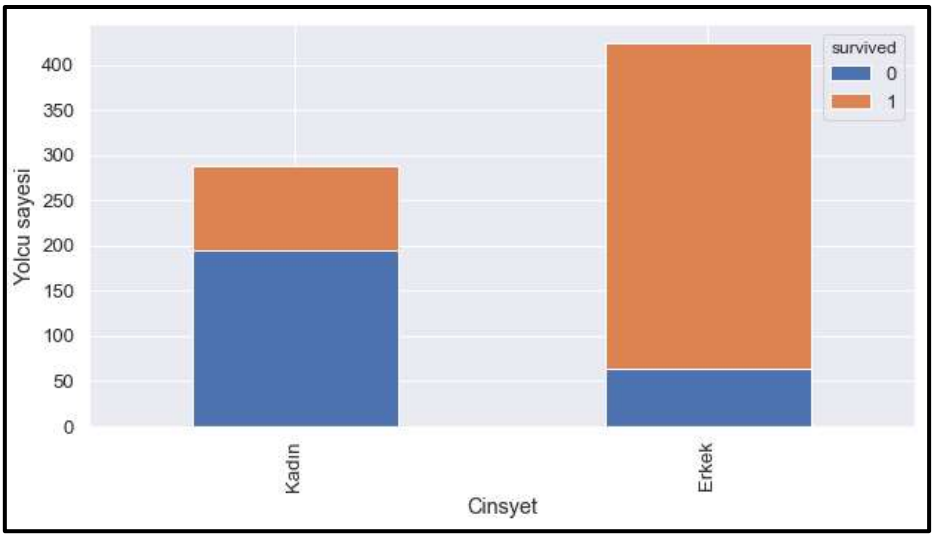
olduğunu görünmektedir. Bunlar, makine öğrenmenin farklı algoritmalarını analiz etmek için kullanacağımız farklı parametrelerdir.



1. Şekil: Korelasyon matris grafiği.



2. Şekil: Yaş grafiği.



3. Şekil : Erkekler ve kadınlar arasında hayatta kalan ve ölen

III. Kullanılmış algoritmalar:

Tahmin modelleri, Lojistik Regresyon, Karar ağacı ve Rangom Forest gibi yedi makine öğrenimi algoritması kullanılarak oluşturulur. Bu algoritmaların her biri, doğruluk yüzdesi temelinde birbiriyle karşılaştırılacaktır. Şekil 4'te makine öğrenme algoritmalar ve çözdükleri problemler görünmektedir.

Veri Ön İşleme:

Tahmin için mevcut veri kümesinde bazı veri değerleri eksik veya bilinmiyor. Bu eksik veriler, genel tahmin modelinin doğruluğunun azalmasına neden oluyordu ve ayrıca saf eğitim verilerinin boyutunu düşürdü ve bu da doğruluğu düşürdü. Veri ön işleme, ham verilerin anlaşılabilir bir biçime dönüştürülmesini içeren bir tekniktir.

Gerçek dünya verileri genellikle eksiktir, tutarsızdır ve/veya belirli davranışlarda veya eğilimlerde eksiktir ve muhtemelen birçok hata içerir. Veri ön işleme, bu tür sorunları çözmek için kanıtlanmış bir yöntemdir. Veri ön işleme, ham verileri daha sonraki işlemler için hazırlar. Eksik değerler, o sütunun ortalaması ile değiştirilir. Böylece yolcuların kolayca tahmin edilebilen eksik ve bilinmeyen verileri bu adımda doldurulur.

Eksik değerlerle başa çıkmak için veri temizliği yapılır. Gözlem yapılırken veri setinin tamamlanmadığı tespit edildi. Bir veya daha fazla alanın boş olarak işaretlendiği çeşitli satırlar vardır (özellikle yaş ve kabin). Kabin sütunundaki verilerin çoğu eksik olduğu için kabin sütunu analizden çıkarıldı. Ancak yaş sütunu çok önemli bir özelliktir, bu nedenle yaş sütunu analiz için tutulur ve NaN(Sayi olmayan) değerleri çıkarılır.. Tahmin modeline daha iyi uyması için cinsiyet sütunu 0 ve 1 (erkek için 0 ve kadın için 1) olarak değiştirildi.

A- Naive Bayes:

Nave Bayes, bir tahmin modeli oluşturmak için Bayes teoremini uygulayan bir sınıflandırma algoritmasıdır. Naive Bayes, bir tahmin modeli oluşturmak için Bayes teoremini uygulayan bir sınıflandırma algoritmasıdır. Özelliklerle ilgili bazı naif varsayımlara dayanmaktadır. Varsayım, tüm özelliklerin birbirinden bağımsız olmasıdır. Diğer bir deyişle, bir sınıfa ait bir özelliğin değer olasılığı diğer tüm özelliklerden bağımsızdır. Bir özelliğin belirli bir değeri için her bir sınıf değerinin olasılığına koşullu olasılık denir. Bir sınıf değerinin olasılığı, tüm koşullu olasılıkların çarpılmasıyla elde edilir. En

yüksek olasılığa sahip sınıf, belirli bir örneğin atanmış sınıfıdır. Naive Bayes algoritmasının farklı türleri vardır.

B- Logistic Regression:

Lojistik Regresyon, bağımlı değişken ikili (binary) olduğunda yapılacak uygun regresyon analizidir. Tüm regresyon analizleri gibi, lojistik regresyon da öngörücü bir analizdir.

Lojistik regresyon, verileri tanımlamak ve bir bağımlı ikili değişken ile bir veya daha fazla nominal, sıra, aralık veya oran düzeyinde bağımsız değişken arasındaki ilişkiyi açıklamak için kullanılır.

Bağımlı değişkenin değerini tahmin etmek için bağımlı ve bağımsız değişken arasındaki regresyon çizgisini kullanma yöntemini kullanır.

C- Karar Ağacı (Decision Tree):

Daha sonra, araştırma analizi Karar ağacı algoritması uygulanarak gerçekleştirilir. Karar ağacı öğrenme, sınıf etiketli eğitim dizilerinden bir karar ağacı oluşturma yöntemidir. Bir karar ağacı, akış şemasına benzer bir yapı olarak düşünülebilir; burada her dahili (yaprak olmayan) düğüm, bir öznitelik üzerinde bir testi belirtir, her dal bir testin sonucunu temsil eder ve her bir yaprak (veya uç) düğüm bir sınıf etiketi.

D- Random Forest:

Sınıflandırma modelinin doğruluğunu daha da artırmak ve hayatta kalmak için en önemli özellikleri belirlemek için rastgele orman algoritması uygulanır. Random forest algoritması, eğitim sırasında çok sayıda karar ağacı oluşturan ve tek tek ağaçların modu olan sınıfı çıkaran bir sınıflandırma algoritmasıdır.Model, temizlenmiş tren veri setinin tüm değişkenleri ile oluşturulmuştur. Pclass, Cinsiyet, Yaş, Aile, Çocuklar, SibSp, Anne, Parch ve Saygıdeğer. Sınıflandırma sürecinde tüm bu farklı değişkenlerin önemini anlamak için, modelimizi oluştururken bir argüman önemi kullanılır. Cinsiyet ve Pclass'ın sınıflandırma modelinde en önemli rolü oynadığı, Anne, Parch ve Saygın değişkenlerin ise en az önemli değişkenler olduğu açıktır. Bu, lojistik regresyon algoritması kullanan analizimizle uyumludur. Random forestalgoritmasının doğruluğu test verileri üzerinde kontrol edilmiştir. Test durumları üzerinde Random forest analizini gerçekleştirdikten sonra, model Tablo VI'da gösterildiği gibi bir karışıklık matrisi oluşturmuştur.

E - K Nearest Neighbor:

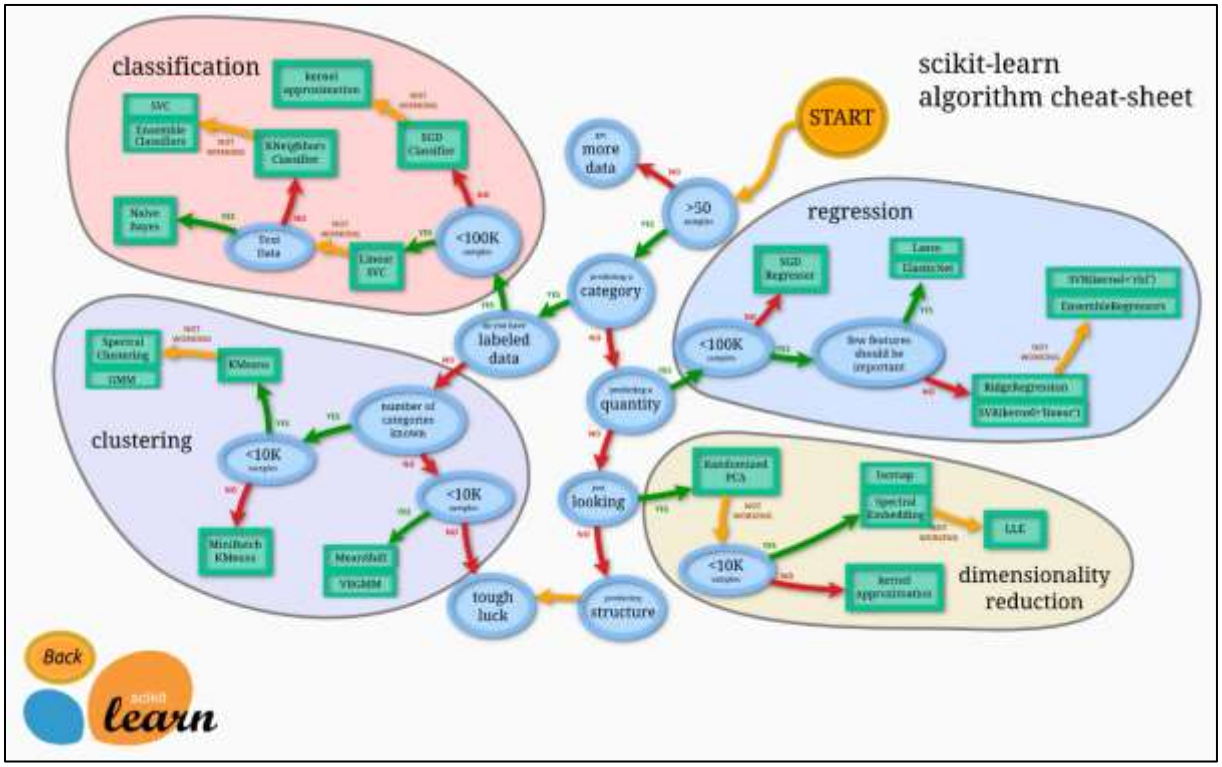
Yapay zekada, daha doğrusu makine öğreniminde, en yakın komşu yöntemi, denetimli bir öğrenme yöntemidir. Kısaltılmış biçimde k-NN veya KNN, İngilizce ise olarak k-nearest neighbors.

Bu bağlamda, N "giriş-çıkış" çiftinden oluşan bir eğitim veri tabanımız var. Yeni bir x girişi ile ilişkili çıkışı tahmin etmek için, k en yakın komşu yöntemi, tanımlanacak bir mesafeye göre girişi yeni giriş x'e en yakın olan k eğitim örneklerini hesaba katmaktan (aynı şekilde) oluşur. Bu algoritma mesafeye dayalı olduğundan, normalleştirme doğruluğunu artırabilir

F - Support Vector Machine (SVM):

Genel olarak, Destek Vektör Makineleri bir sınıflandırma yaklaşımı olarak kabul edilir, ancak hem sınıflandırma hem de regresyon problemlerinde kullanılabilir. Birden çok sürekli ve kategorik değişkeni kolayca işleyebilir. SVM, farklı sınıfları ayırmak için çok boyutlu uzayda bir hiper düzlem oluşturur. SVM, bir hatayı en aza indirmek için kullanılan yinelemeli bir şekilde optimum hiper düzlem oluşturur.

SVM'nin temel fikri, veri kümesini sınıflara en iyi şekilde bölen bir maksimum marjinal hiper düzlem (maximum marginal hyperplane veya MMH) bulmaktır.



3. Şekil: Makine Öğrenme algoritmalar ve kullanılan alanları

Performans Analizi:

Bu araştırmada kullanılan dört tekniği karşılaştırmak için iki ölçü kullanılmıştır. İlk ölçü doğruluk ve ikinci ölçü yanlış keşif oranıdır. Her iki metrik de Confusıon Matrix'i kullanılarak hesaplanır. confusion matrix'inin yapısı tablo 3'te gösterilmektedir. Doğruluk, bir modelin ne kadar iyi öngördüğünün ölçüsüdür. Doğruluk ne kadar yüksekse o kadar iyidir. Doğruluk, TN + TP / Toplam test seti satırı sayısı * 100 formülü kullanılarak hesaplanır.
Yanlış keşif oranı, birden fazla karşılaştırma yapılırken sıfır hipotez testinde tip I (yanlış pozitif) hataların oranını kavramsallaştırmanın bir yöntemidir. Araştırma belgesinde kullanılan problem için, yanlış keşif oranı önemli bir ölçüdür, çünkü sistem bir yolcunun hayatta kalacağını öngörürse ancak gerçekte hayatta kalamazsa tehlikeli olur. Yanlış keşif oranı, FP / FP + TP * 100 formülü kullanılarak hesaplanır. Bu nedenle yanlış keşif oranı ne kadar düşük olursa o kadar iyidir. Her bir algoritma için doğruluk ve yanlış keşif oranı Tablo 5'te listelenmiştir.

.

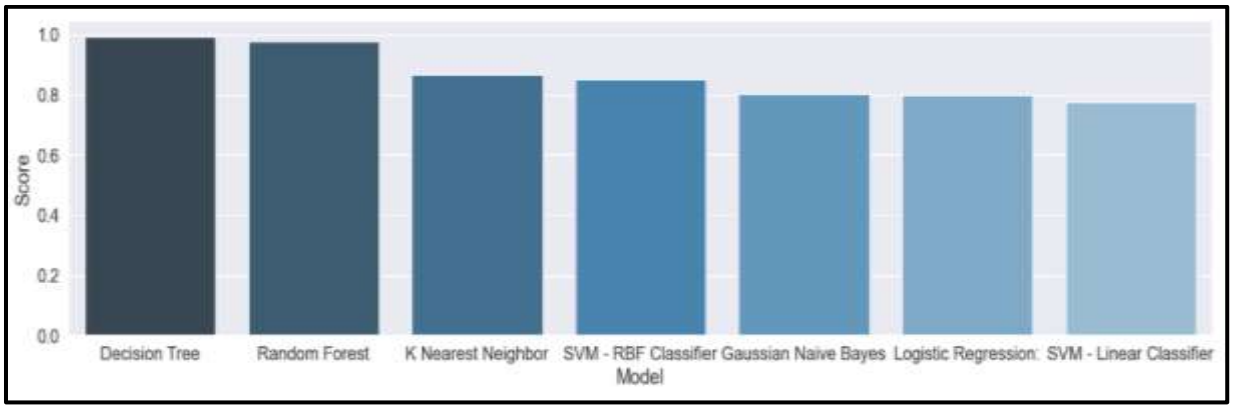
	Actual	**Actual**
Predicted	Survived: NO	Survived: YES
Survived: NO	True Negative (TN)	False Negative (FN)
Survived: YES	False Positive (FP)	True Positive (TP)

3. Tablo: Confusion Matrix'in özellikleri

Algoritmalar	**Doğruluk**
Logistic Regression	% 79.78
K Nearest Neighbor	% 86.64

SVM - Linear Classifier	% 77.68
SVM - RBF Classifier	% 85.06
Gaussian Naive Bayes	% 80.31
Decision Tree	% 99.29
Random Forest	% 97.53

4. Tablo: Algoritmaların öğrenme doğruluk karşılaştırması



5. Şekil: Algoritmaların eğitim doğruluk karşılaştırması

Algoritmalar	**Doğruluk**
Logistic Regression	% 81.11
K Nearest Neighbor	% 78.32
SVM - Linear Classifier	% 79.02
SVM - RBF Classifier	% 79.72
Gaussian Naive Bayes	% 74.82
Decision Tree	% 77.62
Random Forest	% 80.41

5. Tablo: Algoritmaların test doğruluk karşılaştırması

IV. Sonuç:

Hayatta kalıp kalmayacağımızı tahmin etmek için kullanılacak model, *Random Forest Classifier* olacak.

Bu modeli seçilmiş çünkü eğitim ve test verilerinde ikinci en iyiyi yaptı ve test verilerinde % 80.41 ve eğitim verilerinde % 97.53 doğruluğa sahip.

Dolayısıyla, Random Forest'i ve 29 yaşında, üçüncü sınıfta olan (pclass = 3), büyük olasılıkla gemide kardeşleri veya eşleri olmayan (sibsp = 0), çocukları veya ebeveynleri olmayan (parch = 0), minimum ücreti ödemeye çalışacak (ücret = 0) ve Queenstown'da binecek (biniş = 1), bir erkek (cinsiyet = 1) kullanarak hayatta kalamayacağının fark edebiliriz.

Kaynak:

ALPAYDIN, Ethem. *Introduction to machine learning*. MIT press, 2020.

Analyzing Titanic disaster using machine learning algorithms-Computing, Communication and Automation (ICCCA), 2017 International Conference on 21 December 2017, IEEE.

BONACCORSO, Giuseppe. *Machine learning algorithms*. Packt Publishing Ltd, 2017.

BURRELL, Jenna. How the machine 'thinks': Understanding opacity in machine learning algorithms. *Big Data & Society*, 2016, vol. 3, no 1, p. 2053951715622512.

Choosing the right estimator. https://scikit-learn.org/stable/tutorial/machine_learning_map/index.html

Eric Lam, Chongxuan Tang, "Titanic Machine Learning From Disaster", LamTang-TitanicMachineLearningFromDisaster, 2012.

Kaggle.com, 'Titanic Data Science Solutions',[Online]. Available: https://www.kaggle.com/startupsci/titanic-data-science-solutions.

KSHIRSAGAR, Vaishnav et PHALKE, Nahush. Titanic Survival Analysis using Logistic Regression. 2019.

Prediction of Survivors in Titanic Dataset: A Comparative Study using Machine Learning Algorithms, Tryambak Chatterlee, IJERMT-2017.

RASCHKA, Sebastian et MIRJALILI, Vahid. Python Machine Learning: Machine Learning and Deep Learning with Python. *Scikit-Learn, and TensorFlow. Second edition ed*, 2017.

Vyas, Kunal, Zeshi Zheng, and Lin Li, "Titanic-Machine Learning From Disaster", Machine Learning Final Project, UMass Lowell, pp. 1-7, 2015.

ZHANG, Xian-Da. Machine learning. In : *A Matrix Algebra Approach to Artificial Intelligence*. Springer, Singapore, 2020.