Titanik Kazasının Veri Madenciliği Yöntemleri İle İncelenmesi

Investigation of Titanic Accident by Data Mining Methods

Fatih Afşin
Teknoloji Fakültesi Yazılım Mühendisliği / Fırat Üniversitesi
Elazığ,Türkiye

ÖZET

RMS(Royal Mail Ship) Titanik'in batması , tarihin en azılı gemi kazalarından biridir . 15 Nisan 1912'de ilk seferinde Titanik bir buz dağına çarparak battı ve 2224 yolcusundan 1514 yolcu ve mürettebat yaşamını yitirdi. Batması imkansız denilen bu gemide bu denli bir can kaybının olmasının sebeplerinden birisi ise yolcu ve mürettebat için yeterli sayıda cankurtaran olmamasıydı . Efsaneleşmiş bu trajik kaza uluslararası toplumu şoke etti ve gemiler için daha iyi güvenlik düzenlemelerine yol açtı. Hayatta kalmak için bazı şans unsurları bulunsa da,bazı insanların(kadın,çocuk ve üst sınıf gibi) diğer insanlara nazaran hayatta kalma olasılığı daha yüksekti. Günümüzde veri madenciliği yöntemlerinin gelişmesi ile özellikle hangi yolcuların enkazdan kurtulduğunu tahmin etmek için makine öğrenmesi araçlarından 5 tanesini (Desicion Tree , SVM , Random Forest , KNN,Logistic Regression) algoritmalarını uygulayarak elde ettiğimiz başarım incelenmiştir.

Investigation of Titanic Accident by Data Mining Method

ABSTRACT

The RMS(Royal Mail Ship) Titanic sank is one of the shipwecks at least in history . On the first time on April 15,1912, the Titanic slammed into an iceberg and sank and 2224 crew died.One of the reasons why this ship,which is called impossible to sink,caused such a loss of life was that there were not enough lifeguards fort he passengers and crew. This legendary tragic accident shocked the international community and led to better safety arrangements for ships. While there were some chanses for survival,some people(such as women,children,and upper classes) were more likely to survive than others. Today,with the development of data minning methods,our performance has been examined by applying 5 of the machine learning tools(Desicion Tree,SVM, Random Forest,KNN,Logistic Regression) to predict which passengers escaped the wreckage.

1.GİRİŞ

Titanik , White Star Line şirketine ait Oympic sınıfı bir transatlantik yolcu gemisiydi. Harland and Wolff tersanelerinde üretilmiş yapımı tamamlandığında dünyanın en büyük buharlı yolcu gemisiydi . 15 Nisan 1912 gecesi daha ilk seferinde bir uz dağına çarpmış ve yaklaşık iki saat kırk dakika içinde Kuzey Atlantik'in buzlu sularına gömülmüştür.Batışı 1514 kişinin ölüyle sonuçlanmış ve en büyük deniz felaketlerinden biri olarak tarihe geçmiştir.[1] Bu çalışmamızın veri setini bizlere Kaggle sunuyor . Kaggle veri bilimcilerin ,veri setlerinde oluşan problemleri çözmeye çalıştıkları global bir yarışma platformudur. Kaggle'ın bizlere sunmuş olduğu titanik veri setini (VS) tanımakla başlayalım Eğitim setini (train.csv),makine öğrenim modelimizi oluşturmak için kullanacağız . Modelimiz , yolcuların cinsiyet ve sınıfı gibi "özelliklere" dayanacaktır. Survived : hayatta kalma (0=Hayır,1=Evet) , Pclass : bilet sınıfı (1=1., 2=2., 3=3.sınıf) , Sex : Cinsiyet , Sibsp : titanik'teki kardeş/eşsayısı , Parch : Titanik'teki aynı aileye ait ebeveynlerin/çocukların sayısı , Ticket : bilet numarası , Fare : ödenen ücret , Cabin = kabin numarası , Embarked ise yolcuların bindikleri limanlardır . Bu çalışmada kullandığımız VS 'den test ve eğitim setleri oluşturup algoritmalarımız üzerindeki başarımını ölçüp algoritmalarımızı karşılaştıracağız. Çalışmamız da sırasıyla VS üzerinde boş ,eksik (nul ,NaN) değerleri veri ön işleme adımlarından sınıflarına göre medyan değerlerini veya ortalama değerlerini uygulayarak veri setimizde kayıp verilerimizden kurtulup , verinin görselleştirilmesi , Kullanacağımız makine öğrenme algoritmalarının performanslarını arttırmak için Feature enginering (özellik mühendisliği) aşmasını yapıp modellerimizi eğitimini yapıp başarım sonuçlarını karşılaştırdık .

2.VERİ MADENCİLİĞİ (DATA MINING)

Veri madenciliği , büyük miktardaki verinin içinden geleceği tahmin edilmesinde yardımcı olacak

anlamlı ve yararlı bağlantı ve kuralların veya tahminlerin bilgisayar programlarının aracılığıyla aranması ve analizidir . Ayrıca veri madenciliği , çok büyük miktardaki verilerin içindeki ilişkileri inceleyerek aralarındaki bağlantıyı bulmaya yardımcı olan ve veri tabanı sistemleri içerisinde gizli kalmış bilgilerin çekilmesini sağlayan veri analiz tekniğidir. Bu işlemlerin uygulama alanları oldukça geniştir . Bu alanlar içerisinde veri görselliği , yapay sinir ağları , istatistik , yapay öğrenme vb. gibi disiplinler bulunmaktadır. Makine öğrenmesinde sayısal verilerin tahmin edilmesi için Prediction Algorithms kullanılır . Sayısal olmayan yani kategorik verilerin tahmini için ise Classification(Sınıflandırma) kullanılır.

2.1 Bazı Makine Öğrenmesi Algoritmaları

Bu çalışmamızda kullandığımız 5 makine öğrenmesi algoritmalarını tanıyalım ;

Logistic Regression : Lojik regresyon sınıflandırma işlemi yapmaya yarayan bir regresyon yöntemidir .Kategorik veya sayısal verilerin sınıflandırılmasında kullanılır .Bağımlı değişkenin yani sonucun sadece 2 farklı değer alabilmesi durumda çalışır (Evet/Hayır , Erkek/Kadın, Şişman/Zayıf vs.) . [3]

SVM: (Support Vector Machine) Lojik regresyon ile benzer bir sınıflandırma algoritmasıdır . Her ikiside iki sınıfı ayıran en iyi çizgiyi bulmaya çalışırlar. Algoritma çizlecek doğrunun iki sınıfında elemanlarına en uzak yerden geçecek şekilde ayarlanmasını sağlar . Hiçbir parametre almayan bir sınıflayıcıdır. SVM aynı zamanda doğrusal ve doğrusal olmayan verileri de sınıflandırabilir ancak genellikle verileri doğrusal olarak sınıflandırmaya çalışır. [2]

Desicion Tree: Karar ağacı algoritması , veri madenciliği sınıflandırma algoritmalarından biridir . Önceden tanımlanmış bir hedef değişkene sahiptir. Yapıları itibariyle en tepeden aşağı inen bir strateji sunmaktadır. Yani basit karar verme adımları uygulanarak büyük miktardaki kayıtları , çok küçük kayıt gruplarına bölerek kullanılan bir yapıdır .[4]

Random Forest : Rastgele ormanlar veya rastgele karar ormanları , sınıflandırma , regresyon ve diğer görevler için , eğitim aşamasında çok sayıda karar ağacı oluşturarak problemin tipine göre sınıf veya sayı tahmini yapan bir toplu öğrenme yöntemidir. [7]

KNN : K-En yakın komşu algoritması Thomas Cover tarafından sınıflandırma ve regresyon için önerilen parametrik olmayan bir yöntemdir. Her iki durumda da , girdi özellik alanındaki en yakın örneklerinden oluşur. Çıktı K-NN sınıflandırma veya regresyon için kullanılıp kullanılmadığına bağlıdır. [6]

2.2 Veri Madenciliğinde Karşılaşılan Problemler

Veri madenciliği uygulamalarında karşılaşılabilecek problemler şunlardır.

Artık Veri: Artık veri problemde istenilen sonucu elde etmek için kullanılan örneklem kümesindeki gereksiz niteliklerdir. Bu durum pek çok işlem arasında karşımıza çıkabilir .

Boş Veri: Bir veri setinde boş değer , özellikler arasında herhangi bir niteliğin değeri olabilir . Boş değer ,tanımı gereği kendisi de dahil olmak üzere hiçbir değere eşit olmayan veridir.

Dinamik veri: Kurumsal çevrim içi veri tabanları dinamiktir ve içeriği sürekli olarak değişir. Bu durum, bilgi keşfi metotları için önemli sakıncalar doğurmaktadır.

Eksik veri: Veri kümesinin büyüklüğünden ya da doğasından kaynaklanmaktadır.

Eksik veriler olduğunda yapılması gerekenler şunlardır:

- Eksik veri içeren kayıt veya kayıtlar çıkarılabilir.
- Değişkenin ortalaması eksik verilerin yerine kullanılabilir.
- Var olan verilere dayalı olarak en uygun değer kullanılabilir.

Gürültülü ve Kayıp Değerler: Veri girişi veya veri toplanması esnasında oluşan sistem dışı hatalara gürültü denir. Büyük veri tabanlarında pek çok niteliğin değeri yanlış olabilir. Veri toplanması esnasında oluşan hatalara ölçümden kaynaklanan hatalar da dâhil olmaktadır. Bu hataların sonucu olarak birçok niteliğin değeri yanlış olabilir ve bu yanlışlardan dolayı veri madenciliği amacına tam olarak ulaşmayabilir.

Sınırlı Bilgi: Veri tabanları genel olarak basit öğrenme işlerini sağlayan özellik veya nitelikleri sunmak gibi veri madenciliği dışındaki amaçlar için hazırlanmışlardır. Bu yüzden, öğrenme görevini kolaylaştıracak bazı özellikler bulunmayabilir.[8]

3.Veri Madenciliği Aşamaları

Veri Madenciliği Yöntemi ile Titanik Kazasının Araştırılması veri madenciliği uygulaması aşağıda aşamalar halinde verilmiştir.

3.1 Verilerin Temizlenmesi

VS verilerinde bir sapma , anormal bir değer olup olmadığının tespiti için veri kalitesi incelenmiştir.

3.1.1 Artık Verilerin Temizlenmesi

Bu çalışmamızda modelimizi eğitmeden önce Passenger Id ve Cabin adlı feature'leri (Nitelik,özellik) veri setimizde modelimizin eğitimi sırasında hata ve sorun ile karşılaşmamak için silinmiştir

3.1.2 Boş Değerlerin Doldurulması

VS' de boş değer olarak Embarked yani yolcuların hangi limandan bindiğini gösteren nitelik içerisinde saptanmış olup bu durumdan kurtulmak için embarked değeri ile bilete ödenen fiyat arasında bir ilişki bulup boş değer saptanan kayıtlarda ödenen ücrete göre yolcunun embarked değeri girilmiştir.

3.1.3 Eksik Verilerin Tamamlanması

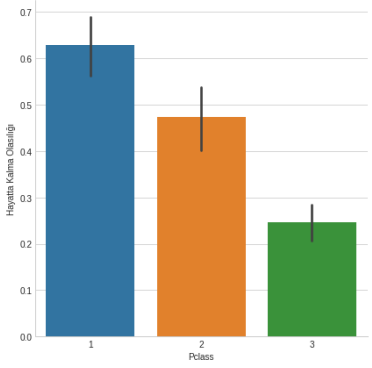
Bu çalışmamızda VS üzerinde bir'den fazla nitelik'te eksik veri saptanmış olup bu verileri VS değerlerini bozmamak için bazı niteliklerde ortalamasını bazı niteliklerde ise medyan ortanca değeri yazılmıştır.

3.2 Verilerin Görselleştirilmesi

VS de bulunan verileri grafik halinde sunulmuştur.

3.2.1 Sınıflara Göre Hayatta Kalma

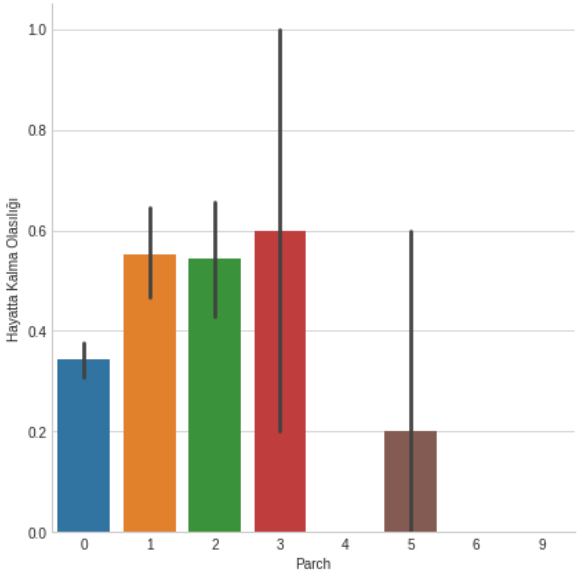
Pclass yolcularımızın seyahat ettiği sınıflardı Şekil-1 'de gördüğümüzde 1.sınıfta hayatta kalma oranı yüksek olduğudur. Burada değerlerimizin ve özelliklerimizin çok olmayışından ötürü modelimiz için yeni bir özellik oluşturulmasına gerek duyulmamıştır .



Şekil-1

3.2.2 Aile Sayısının Hayatta Kalma Oranları

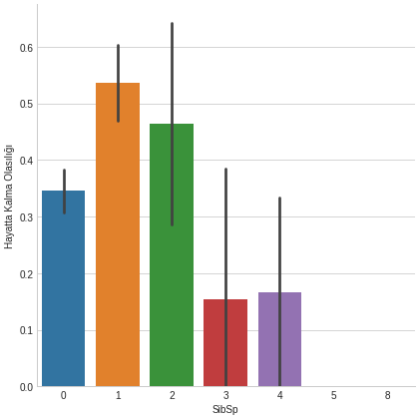
Şekil 2'deki grafikte bar plotların üstünde olan çizgiler şunu gösterir ; örnek olarak 3'ü ele alırsak ortalama olarak 3 bireye sahip ailelerin hayatta kalma olasılığı %60 dır ama ortadaki çizgi bize 3 kişiye sahip ailenin hayatta kalma oranını 0.2-1.0 aralığında olabileceğini de gösteriyor . Burada hayatta kalma olasılığı çok değişken ve büyük bir aralığa sahip .Model geliştirirken bunu göz önüne alıp veri setimizi Parch ile SibSp yi 3 ten aşağısı ve yukarısı olarak birleştirip standart sapmamızın daha tutarlı olmasını sağladık .



Şekil-2

3.2.3 Ebeveyn Çocuk Sayısına Göre Hayatta Kalma

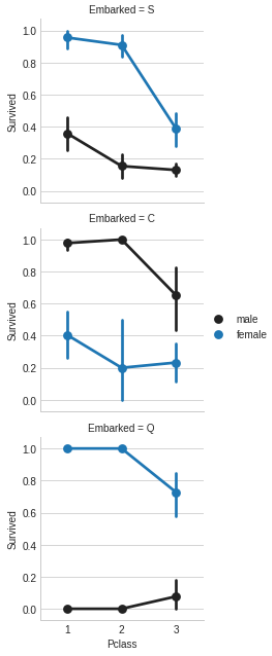
Şekil-3 te gördüğümüz eğer 2 den fazla SibSp değerine yani ebeyn çocuk sayısına sahip bir ailenin hayatta kalma olasılığı sert bir şekilde azalıyor. SibSp==0 veya 1 veya 2 olan grubun hayatta kalma oranları daha fazla. Bu kategoriye göre yeni bir feature(özellik) oluşturduk.



Şekil-3

3.2.4 Yolcuların Bindikleri Liman-Cinsiyet için Hayatta Kalma Oranları

Şekil-4 te görüldüğü gibi kadın yolcular erkek yolculara göre daha fazla hayatta kalmışlardır. Erkeklerin ise bindikleri limana göre C limanında binenlerin hayatta kalma olasılığı daha yüksek olduğu görülmüştür. Bu durum sınıflandırma yaparken doğrudan modelimizde kullanılmıştır.



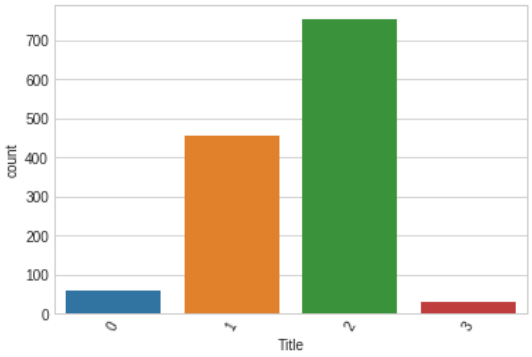
Şekil-4

3.3 Öznitelik Seçimi

Makine öğreniminde öznitelik çıkarımı büyük bir veri kümesini açıklamak için gereken kaynak miktarını azaltmayı içerir [9] .Veri setimizde 3 farklı öznitelik değeri oluşturulmuştur.

3.3.1 Yolcu isimlerinin Başlıklara Bölünmesi

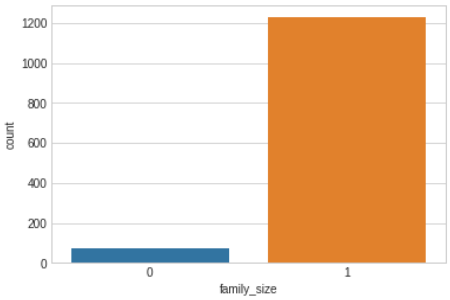
Şekil-5 teki gibi Name niteliğini "Master = 0","Miss,Ms,Mile,MRS =1","Mr =2","Mrs =3" olarak 4 başlığa indirgeyip title adlı yeni bir öznitelik oluşturulmuştur.



Şekil-5

3.3.2 Aile Niteliğinin İndirgenmesi

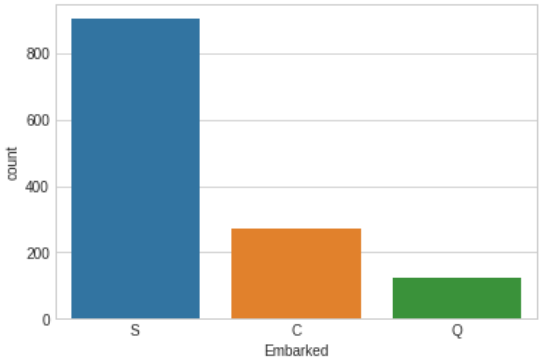
Aile özniteliği incelendiğinde aile sayısının 5 ten büyük olduğu durumlarda hayatta kalma oranlarının sert bir şekilde düşmesinden ötürü bu niteliği aile sayısı 5 ten küçük ve büyük olmak üzere Şekil-6 'da olduğu gibi family_size adında yeni bir özniteliğe dönüştürülmüştür.



Şekil-6

3.3.3 Liman Niteliğinin İndirgenmesi

Yolcuların bindikleri limanlara göre Şekil-7 'de olduğu gibi Embarked adı altında "S","C","Q" formatına dönüştürülmüştür.



Şekil -7

4.Model'in Oluşturulması

Öncelikle modelimizden iki farklı test veri setini oluşturup bu veri setlerimizi makine öğrenmesi algoritmalarımızda teker teker başarım oranları test edilmiştir.

4.1 Test Veri Setinin Oluşturulması

Modelimizi oluşturmak için hayatta kalma niteliğini baz alarak rastgele bir şekilde veri setimizi %33 oranında test veri setlerine böldük .

```
X_train 590
X_test 291
y_train 590
y_test 291
test 418
```

Şekil-8

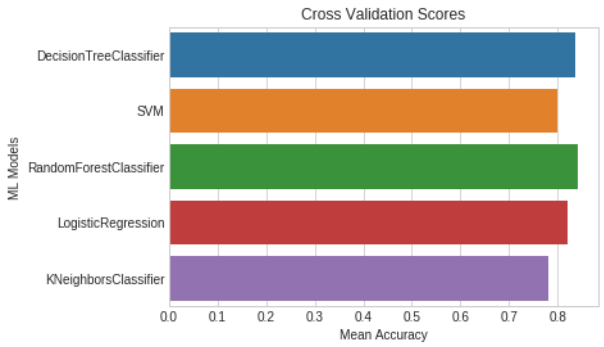
4.2 Makine Öğrenmesi Algoritmalarının Uygulanması

Sırası ile Desicion Tree , SVC, Random Forest, Logistic Regression,K-NN algoritmaları uygulandı . Bu modellerin içerisinde bulunan parametrelerin en iyi parametresini (Hyper Paramater) Grid Search yöntemine göre arayıp , bulduğumuz parametrelerin en iyi değerlerini karşılaştırırken Cross Validation yöntemini uyguladık.

```
Fitting 10 folds for each of 250 candidates, totalling 2500 fits

[Parallel(n_jobs=-1)]: Using backend LokyBackend with 4 concurrent workers.
[Parallel(n_jobs=-1)]: Done  88 tasks      | elapsed:    1.6s
[Parallel(n_jobs=-1)]: Done 2500 out of 2500 | elapsed:    6.8s finished
[Parallel(n_jobs=-1)]: Using backend LokyBackend with 4 concurrent workers.

0.8355932283389831
Fitting 10 folds for each of 28 candidates, totalling 280 fits

[Parallel(n_jobs=-1)]: Done 128 tasks      | elapsed:    1.7s
[Parallel(n_jobs=-1)]: Done 280 out of 280 | elapsed:    4.3s finished
[Parallel(n_jobs=-1)]: Using backend LokyBackend with 4 concurrent workers.

0.7983050847457627
Fitting 10 folds for each of 54 candidates, totalling 540 fits

[Parallel(n_jobs=-1)]: Done  42 tasks      | elapsed:    5.4s
[Parallel(n_jobs=-1)]: Done 192 tasks      | elapsed:   20.6s
[Parallel(n_jobs=-1)]: Done 442 tasks      | elapsed:   50.5s
[Parallel(n_jobs=-1)]: Done 540 out of 540 | elapsed:  1.0min finished

0.840677966101695
Fitting 10 folds for each of 14 candidates, totalling 140 fits

[Parallel(n_jobs=-1)]: Using backend LokyBackend with 4 concurrent workers.
[Parallel(n_jobs=-1)]: Done 123 tasks      | elapsed:    0.8s
[Parallel(n_jobs=-1)]: Done 140 out of 140 | elapsed:    1.0s finished
/opt/conda/lib/python3.7/site-packages/sklearn/linear_model/_logistic.py:940: ConvergenceWar
ning: lbfgs failed to converge (status=1):
STOP: TOTAL NO. of ITERATIONS REACHED LIMIT.

Increase the number of iterations (max_iter) or scale the data as shown in:
    https://scikit-learn.org/stable/modules/preprocessing.html
Please also refer to the documentation for alternative solver options:
    https://scikit-learn.org/stable/modules/linear_model.html#logistic-regression
  extra_warning_msg=_LOGISTIC_SOLVER_CONVERGENCE_MSG)
[Parallel(n_jobs=-1)]: Using backend LokyBackend with 4 concurrent workers.

0.8203389830508474
Fitting 10 folds for each of 40 candidates, totalling 400 fits

[Parallel(n_jobs=-1)]: Done 280 tasks      | elapsed:    1.2s

0.7796610169491525

[Parallel(n_jobs=-1)]: Done 400 out of 400 | elapsed:    1.6s finished
```

Şekil-9

Kullandığımız modellerdeki başarım oranlarının Cross Validation skorlarını tablo üzerinde gösterimi Şekil-10 verilmiştir.



Şekil-10

4.Sonuç

Titanik veri setimiz üzerinde görselleştirme ,veri ön işleme ve yeni öznitelikleri oluşturulmasından sonra 5 farklı makine öğrenimi modeli eğitmeye başladık . Bu modellerin içerisinde bulunan parametrelerin en iyi parametresini (Hyper Paramater) Grid Search yöntemine göre arayıp , bulduğumuz parametrelerin en iyi değerlerini karşılaştırırken Cross Validation yöntemini uyguladık ve sonuç olarak 0.840677966101695 yani %84 başarım oranında Random Forest algoritması ile başarım oranı en yüksek algoritmamızın Random Forest olduğunu gördük.

Veri görselleştirme kısmında özellikle cinsiyeti kadın olan ve üst sınıf insanların hayatta kalma oranlarının yüksek olduğunu ve üst sınıfta olan insanların aynı limandan bindiklerini görmüş olduk. Model başarımını yükseltebilmek için daha kapsamlı öznitelikler çıkartılıp , gürültülü özellikleri tanımlayıp bu verileri kaldırarak başarım oranı iyileştirilebilir.

KAYNAKLAR (REFERENCES)

[1]Wiki,RMS Titanic,Wikipedia

[2]Kadir Ulgen ,(2016)Makine Öğrenmesinde Karar Ağaçları, Medium

[3]Ekrem Hatipoğlu , (2018)logisctic regression ,part-8,Medium

[4] Ekrem Hatipoğlu, (2018)machine-learning, classification ,supportvector ,Medium

[5]Wiki , Rastgele Orman, Wkipedia.Org

[6]Wiki , K-nearest_neighbors_algorithm ,Wikipedia.Org

[7]Kadir Alan ,(2020) veri madenciliği yöntemleri , Medium

[8]Levent Sabah,Hüseyin Bayraktar ,(2020) Veri madenciliği birliktelik kuralları yöntemi kullanılarak binaların risk durumlarının belirlenmesi,Dergi Park

[9]Wiki,öznitelik çıkarımı,Wikipedia