



DÖNEM PROJESİ
MAKİNE ÖĞRENMESİ METHODLARI ile
MÜŞTERİ MEMNUNİYETİNİ ANLAMA

GAZİ ÜNİVERSİTESİ
BİLİŞİM ENSTİTÜSÜ

İsmail GÜRLER – 21805004017
DANIŞMAN: Prof. Dr. Cevriye GENCER

Giriş

Makine öğrenimi algoritmaları, geçmiş deneyimlerden öğrenmek ve büyük, yapılandırılmamış ve karmaşık veri kümelerinden faydalı çıkarımlar yapmak ve veri desenlerini tespit etmek için çeşitli istatistiksel, matematiksel ve optimizasyon yöntemleri kullanır. Bu algoritmalar, otomatik metin kategorizasyonu ve çevirisi, ağ saldırı tespiti, spam e-posta tespiti, kredi kartı dolandırıcılığının tespiti, müşteri davranışlarının tespiti ve duygu analizi, üretim sürecinin optimizasyonu ve hastalık teşhisi gibi geniş bir uygulama alanına sahiptir. Bu uygulamaların çoğu makine öğrenimi algoritmalarının denetimli (supervised) varyantları kullanılarak gerçekleştirilmiştir. Denetimli öğrenme varyantlarında, etiketin belli olduğu bir veri kümesi öğrenilerek bir tahmin modeli geliştirilir ve buna göre etiketsiz örneklerin sonucu tahmin edilebilir.

Bu çalışmada, makine öğrenimi algoritmaları kullanılarak veri bilimi tahmin çalışması yapılması hedeflenmiştir. Online bir veri kaynağından python arayüzü olan jupyter lab programı ile havayolları anket verileri çekilip veri kümesi oluşturulmuş; bu ham verinin analizinin yapılabilmek için veri ön işleme yapılmış ve sonrasında denetimli (supervised) makine öğrenmesi algoritmaları ile model oluşturup veriler eğitilmiştir. Veriler eğitilip model oluşturulurken Lojistik regresyon (logistic regression), karar ağaçları (decision tree), K en yakın komşu (K nearest Neighbour), gaussian naive bayes, ve doğrusal ayırmacılık analizi (Linear Discriminant Analysis) algoritmaları kullanılmıştır.

MAKİNE ÖĞRENİMİ

Makine öğrenimi (ML), bilgisayar sistemlerinin belirli bir görevi yerine getirmek için kullandıkları algoritmaların ve istatistiksel modellerin programlandığı bilimsel çalışmadır. Makine öğrenimi algoritmaları günlük hayatta kullandığımız birçok uygulamada kendine yer edinmiştir. Google gibi bir web arama motoru internette arama yapmak için her kullanıldığında, bu kadar iyi çalışmasının nedenlerinden biri, web sayfalarını nasıl sıralayacağını öğrenen bir öğrenme algoritması olmasıdır. Makine öğrenimini kullanmanın temel avantajı, bir algoritmanın verilerle eğitildikten sonra işini kendi yapabilmesidir.

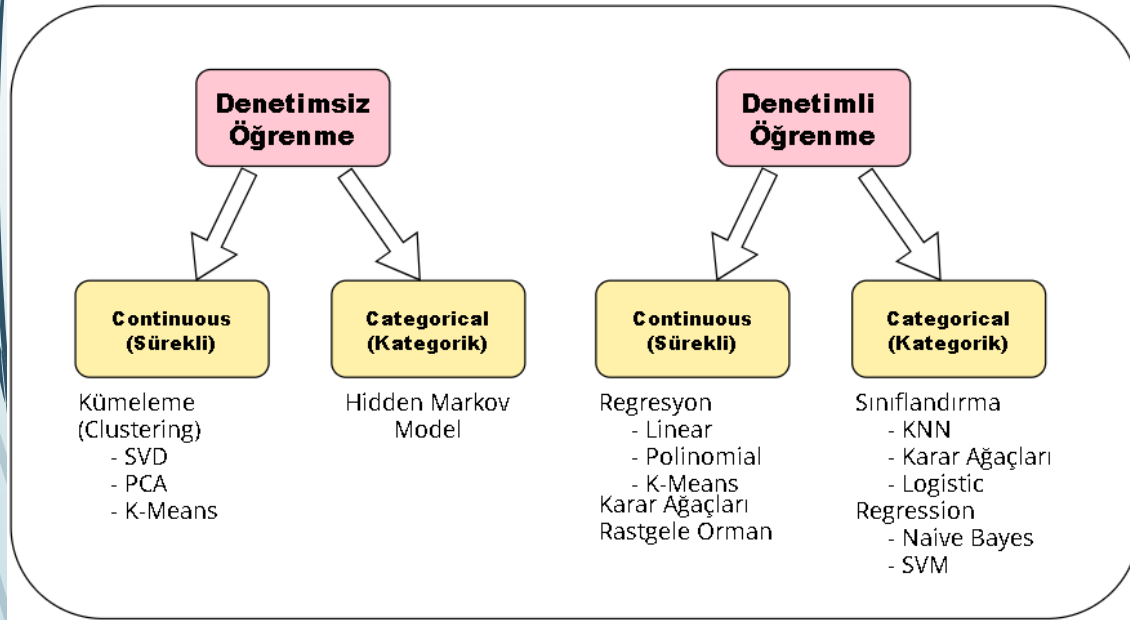
Makine öğrenimi, makinelere verileri nasıl daha verimli kullanacaklarını öğretmek için kullanılır. Halihazırda çok fazla veri üretmemiz ile makine öğreniminin kullanım alanı artmaktadır. Birçok endüstri, ilgili verileri çıkarmak, analiz etmek ve kullanmak için makine öğrenimini uygulamaktadır. Birçok matematikçi ve programcı, büyük veri setlerine sahip olan problemlerin en doğru ve sürekli çözümünü bulmak için çeşitli makine öğrenimi yaklaşımları uygulamaktadır.

Makine Öğrenimi, veri problemlerini çözmek ve çözerken en iyi sonucu almak için farklı algoritmaları kullanmayı ya da en iyi algoritmayı bulmayı gerektirir. Kullanılan algoritma türü, çözmek istenilen sorunun türüne, değişken sayısına, veri tipine ve buna en uygun model türüne bağlıdır. Genel olarak, 3 ana Makine Öğrenimi Algoritması türü vardır. Bunlar denetimli öğrenme, denetimsiz öğrenme ve pekiştirmeli öğrenmedir.

MACHINE LEARNING

MAKİNE ÖĞRENİMİ VE ALGORİTMALAR

Şekil'de değişken türlerine göre kullanılabilecek denetimli ve denetimsiz öğrenme modelleri aktarılmıştır.



1. Denetimli Öğrenme (Supervised Learning)

Denetimli öğrenme, örnek girdi-çıkı değerlerine dayalı olarak girdiyi çıktıya eşleyen bir işlevi öğrenip model oluşturmaya yönelik makine öğrenimi görevidir. Etiketli bir dizi eğitim verisinden bir metot oluşturur. Denetimli makine öğrenimi algoritmaları, harici yardıma ihtiyaç duyan algoritmalar. Sahip olunan veri kümesi, eğitim ve test veri kümelerine ayrılır. Eğitim veri kümesi, tahmin edilmesi veya sınıflandırılması gereken girdi ve çıktı değişkenine sahiptir. Tüm algoritmalar eğitim veri kümesinden öğrenerek bir model oluşturur ve modelden ürettiği çıktıları test veya doğrulama için test veri kümesine uygular.

2. Denetimsiz Öğrenme (Unsupervised Learning)

Denetimsiz öğrenme metodu, verilerden kurallar üretmek için tanımlanmamış mevcut örüntüleri tanımlamayı sağlamaktadır. Bu teknik, veri kategorilerinin bilinmediği ve verilerin etiketlerin olmadığı durumlarda kullanılmaktadır. Burada eğitim verileri etiketlenmemiştir. Denetimsiz öğrenme, öğrenme için istatistik tabanlı bir yaklaşım olarak kabul edilir ve bu nedenle etiketsiz verilerdeki gizli yapıyı bulma problemini ifade eder.

2. Pekiştirmeli Öğrenme (Reinforcement Learning)

Bu algoritma kullanılarak makine belirli kararlar vermek üzere eğitilir. Makine, ceza ödül sistemini kullanarak kendini sürekli eğittiği bir ortama maruz bırakılır. Bu makine geçmiş deneyimlerden öğrenir ve doğru iş kararları vermek için mümkün olan en iyi bilgiyi yakalamaya çalışır.

Denetimli Öğrenme (Supervised Learning)

Denetimli Öğrenme Algoritmaları: Lojistik Regresyon, Karar Ağacı, Rastgele Orman, KNN, Linear Ayrımcılık Analizi:

- Karar Ağacı: Veri kümelerini ağaç benzeri bir yapıda sınıflandırmak için karar adımlarını, yani testleri ve karşılık gelen sonuçları modeller. Bir karar ağacının düğümleri, ilk veya en üst düğüm olan kök düğümle bu kök düğümünün bir veya daha fazla seviyede dallanarak ayrıldığı yavru düğümlerden oluşmaktadır. Test sonucuna bağlı olarak, sınıflandırma algoritması uygun alt düğüme doğru dallanır ve burada test ve dallanma süreci yaprak düğüme ulaşana kadar tekrar eder. Yaprak düğümler karar sonuçlarına karşılık gelir. KA'ların yorumlanması kolay ve öğrenilmesi hızlı bulunduğu için birçok sınıflandırma alanlarında kullanılmıştır.
- Lojistik Regresyon: Bu algoritma bir regresyon değil sınıflandırmadır. Belirli bir bağımsız değişkenler kümesine dayalı olarak ayrık değerleri (0/1, evet/hayır gibi ikili değerler) tahmin etmek için kullanılır. Olasılığı tahmin ettiğinden, çıktı değeri genellikle 0 ile 1 arasında yer alır. Lojistik regresyon, sınıflar arasındaki sınırların belirlenmesini sağlar. Veri seti daha büyük olduğunda uç noktalara yani doğru karara daha hızlı ulaşır. Lojistik regresyonu bir sınıflandırıcıdan daha fazlası yapan olasılık değerleriyle çalışmasıdır
- Doğrusal ayrımcılık analizi: Özniteliklerin bir doğrusal birleşimini bularak veriyi sınıflara ayırmaya yarayan bir yöntemdir. LDA, makine öğrenimi sınıflandırma uygulamaları için ön işlemede boyutluluğu azaltmaya yönelik yaygın bir tekniktir. LDA, sınıflar arası varyansın sınıf içi varyansa oranını en aza indirerek özellikleri daha düşük boyutlu bir uzaya dönüştürmek ve böylece maksimum sınıf ayrılabilirliğini garanti etmek için geliştirilmiştir. DAA'nın temel amacı, bir uzayı (N boyutlu veri), sınıf ayrımını korurken daha küçük bir K alt uzayına ($K \leq n - 1$) yansıtmaktır.

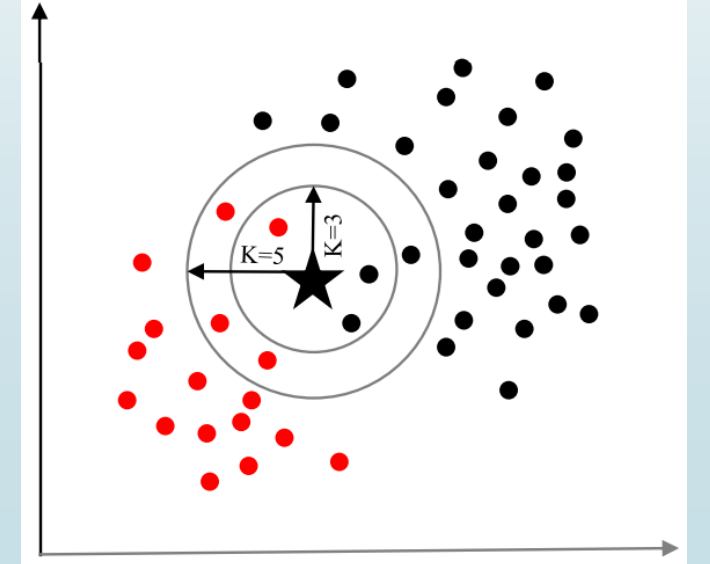
Denetimli Öğrenme (Supervised Learning)

- **Gaussian Naive Baise (NB)**: Bayes'in Olasılık Teoremini kullanarak sınıflandırma problemlerini çözmek için kullanılır. Bayes teoremi, $P(A/B)$ ile temsil edilen B olayının önceki olasılıklarını göz önüne alarak bir olayın (A) olasılığını aşağıdaki gibi hesaplar:

$$P(A/B) = P(B/A) * P(A) / P(B)$$

Naïve Bayes sınıflandırıcıları, özellikler arasında güçlü bağımsızlık varsayımlarına sahip Bayes Teoremi kavramına dayanan basit olasılıksal sınıflandırıcılar olarak kullanılırlar. Yani değişkenler arasında güçlü bir korelasyon olmaması öngörülmektedir.

- **K En Yakın Komşu(KNN) Algoritması**: NB tekniğinin aksine, KNN algoritması olasılık değerlerini dikkate almayı gerektirmez. KNN algoritmasında 'K', 'oy' almak için dikkate alınan en yakın komşuların sayısıdır. Oy alması demek hangi küme elemanına yakınsa o gruba ait olmak için artı bir değer kazanması demektir. 'K' için farklı değerlerin seçilmesi, aynı örnek nesne için farklı sınıflandırma sonuçları üretebilir. Grafikte KNN'nin yeni bir nesneyi sınıflandırmak için nasıl çalıştığının bir örneğini göstermektedir. K=3 için yeni nesne (yıldız) 'siyah' olarak sınıflandırılmıştır; ancak K=5 olduğunda 'kırmızı' olarak sınıflandırılmıştır.



MÜŞTERİ MEMNUNİYETİ UYGULAMASI

Problemin Tanımlanması

Çalışmada, beş farklı makine öğrenimi algoritmasıyla, havayolu firması yolcu memnuniyet anket verileri kullanılarak müşteri memnuniyetinin en iyi oranda tahmin edebilmesi hedeflenmiştir.

Verinin Hazırlanması

Veri kümesi Kaggle platformundan eğitim ve test verisi olarak çekilmiştir. Kaggle, veri bilimcileri ve makine öğrenimi çalışanları için çevrimiçi bir topluluk platformudur. Kaggle, kullanıcıların diğer kullanıcılarla iş birliği yapmasına, veri kümeleri bulmasına ve veri bilimi zorluklarını çözmek için diğer veri bilimcilerle rekabet etmesine olanak tanır.

Kütüphanelerin Çağırılması

Veri setini yüklemek ve ön işleme yapmak, veri setiyle istenildiği gibi oynamak, üzerinde analiz yapmak ve sonuçları görselleştirmek için çeşitli python kütüphaneleri kullanılmaktadır. Aşağıdaki Python kütüphaneleri kullanılmıştır:

- NumPy, Python'da bilimsel hesaplama yapabilen temel paketlerden biridir.
- Pandas, veri işleme ve analiz için kullanılan bir Python kütüphanesidir.
- Matplotlib, Python'daki en çok kullanılan bilimsel görsel ve çizim kütüphanelerinden biridir.
- Seaborn, Matplotlib üzerinde çizim stili ve renk varyantları için uygun seçenekler sunar.
- Scikit-learn (Sklearn) Python'da tutarlı bir arayüz aracılığıyla sınıflandırma, regresyon, kümeleme ve boyutsallık azaltma dahil olmak üzere makine öğrenimi ve istatistiksel modelleme için bir dizi verimli araç sağlayan bir kütüphanedir.



Veri Setinin Özetlenmesi

8

Shape fonksiyonu ile veri setinde toplamda 103.904 satırın ve 23 tane öz niteliğin olduğu ve yüklenen verinin ilk 5 satırı; çağırılan özniteliklerin yani sütunların hepsi ve öznitelikler açıklamalarıyla detaylı olarak aşağıda aktarılmıştır.

```
[2]: data = pd.read_csv('train.csv')
data.head()
```

```
[2]:
```

	Unnamed: 0	id	Gender	Customer Type	Age	Type of Travel	Class	Flight Distance	Inflight wifi service	Departure/Arrival time convenient	...
0	0	70172	Male	Loyal Customer	13	Personal Travel	Eco Plus	460	3	4	...
1	1	5047	Male	disloyal Customer	25	Business travel	Business	235	3	2	...
2	2	110028	Female	Loyal Customer	26	Business travel	Business	1142	2	2	...
3	3	24026	Female	Loyal Customer	25	Business travel	Business	562	2	5	...
4	4	119299	Male	Loyal Customer	61	Business travel	Business	214	3	3	...

5 rows x 25 columns

```
[4]: data.shape
```

```
[4]: (103904, 25)
```

```
[5]: data.columns
```

```
[5]: Index(['Unnamed: 0', 'id', 'Gender', 'Customer Type', 'Age', 'Type of Travel',
        'Class', 'Flight Distance', 'Inflight wifi service',
        'Departure/Arrival time convenient', 'Ease of Online booking',
        'Gate location', 'Food and drink', 'Online boarding', 'Seat comfort',
        'Inflight entertainment', 'On-board service', 'Leg room service',
        'Baggage handling', 'Checkin service', 'Inflight service',
        'Cleanliness', 'Departure Delay in Minutes', 'Arrival Delay in Minutes',
        'satisfaction'],
        dtype='object')
```

Sütun (Öznitelik)	Özniteliklerin Açıklaması
Gender:	Yolcuların cinsiyeti (Kadın, Erkek)
Customer Type:	Müşteri tipi (Sadık müşteri, sadakatsiz müşteri)
Age:	Yolcuların gerçek yaşı
Type of Travel:	Yolcuların uçuş amacı (Kişisel Seyahat, İş Seyahati)
Class:	Yolcuların uçaktaki seyahat sınıfı (Business, Eco, Eco Plus)
Flight distance:	Bu yolculuğun uçuş mesafesi
Inflight wifi service:	Uçak içi wifi hizmetinden memnuniyet düzeyi (0: Geçerli Değil;1-5)
Departure/Arrival time convenient:	Kalkış/Varış saatinin uygunluğuna ilişkin memnuniyet düzeyi
Ease of Online booking:	Online rezervasyonun kolaylığıyla ilgili memnuniyet düzeyi
Gate location:	Uçağa giriş kapısının konumunun memnuniyet düzeyi
Food and drink:	Yiyecek ve içecekten memnuniyet düzeyi
Online boarding:	Online check-in memnuniyet düzeyi
Seat comfort:	Koltuk konforu memnuniyet seviyesi
Inflight entertainment:	Uçak içi eğlenceden memnuniyet düzeyi
On-board service:	Uçağa binış hizmetinden memnuniyet düzeyi
Leg room service:	Koltuklar arası ayak mesafesinden memnuniyet düzeyi
Baggage handling:	Bagaj işlemlerinden memnuniyet düzeyi
Check-in service:	Check-in hizmetinden memnuniyet düzeyi
Inflight service:	Uçak içi hizmetlerden memnuniyet düzeyi
Cleanliness:	Temizlikten memnuniyet düzeyi
Departure Delay in Minutes:	Kalkışta gecikilen dakika
Arrival Delay in Minutes:	Varışta gecikilen dakika
Satisfaction:	Havayolu memnuniyet düzeyi (Memnuniyet, nötr veya memnuniyetsizlik)

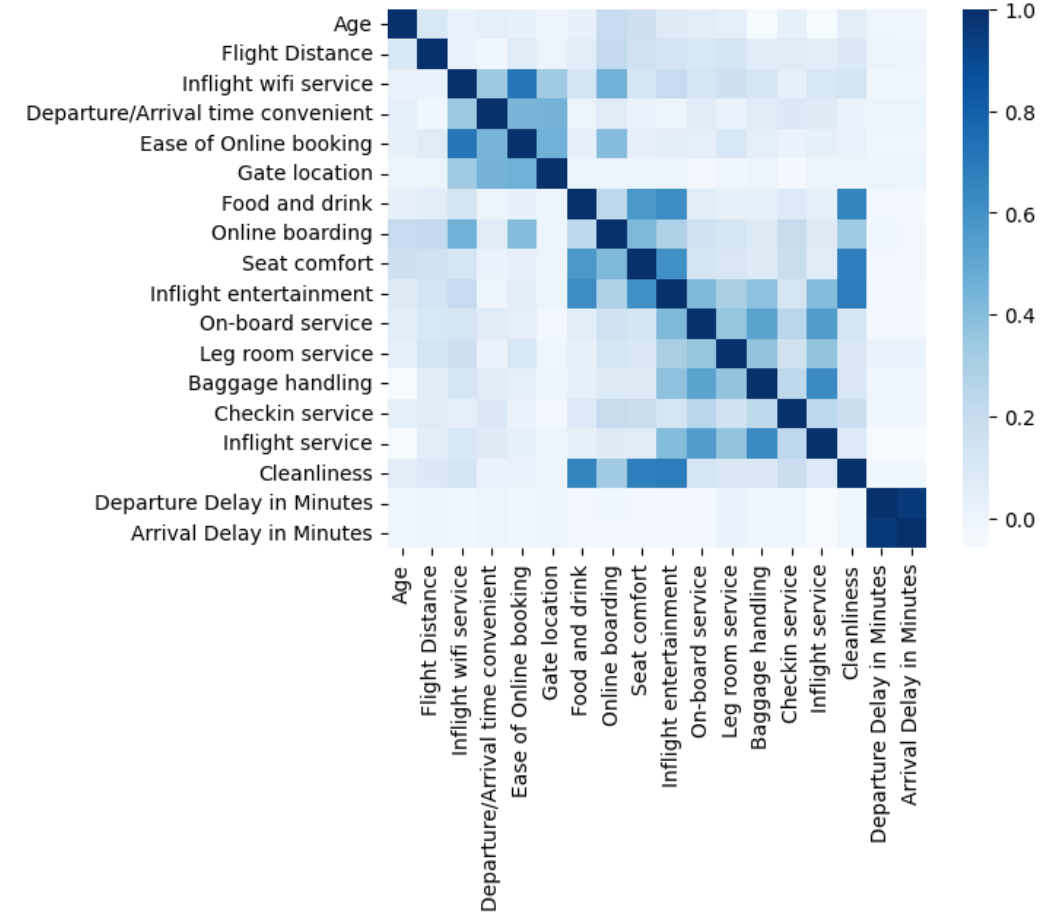
Korelasyon ve Eksik Verilerin Kontrolü

9

IsNull() fonksiyonu ile veri de eksik veri olup olmadığı detaylı olarak kontrol edildiğinde “Arrival Delay in Minutes” sütununda 310 satırın eksik olduğunu bilgisine ulaşılmıştır. “Arrival Delay in Minutes” sütununun silinmesi düşünülebilir, ancak bu sütundaki eksik değerler çok az olduğu için, sadece eksik değerleri ortalama(mean) fonksiyonu ile tahmin edilip veri setine eklenmesi modelin oluşmasını olumsuz olarak etkilemeyecektir.

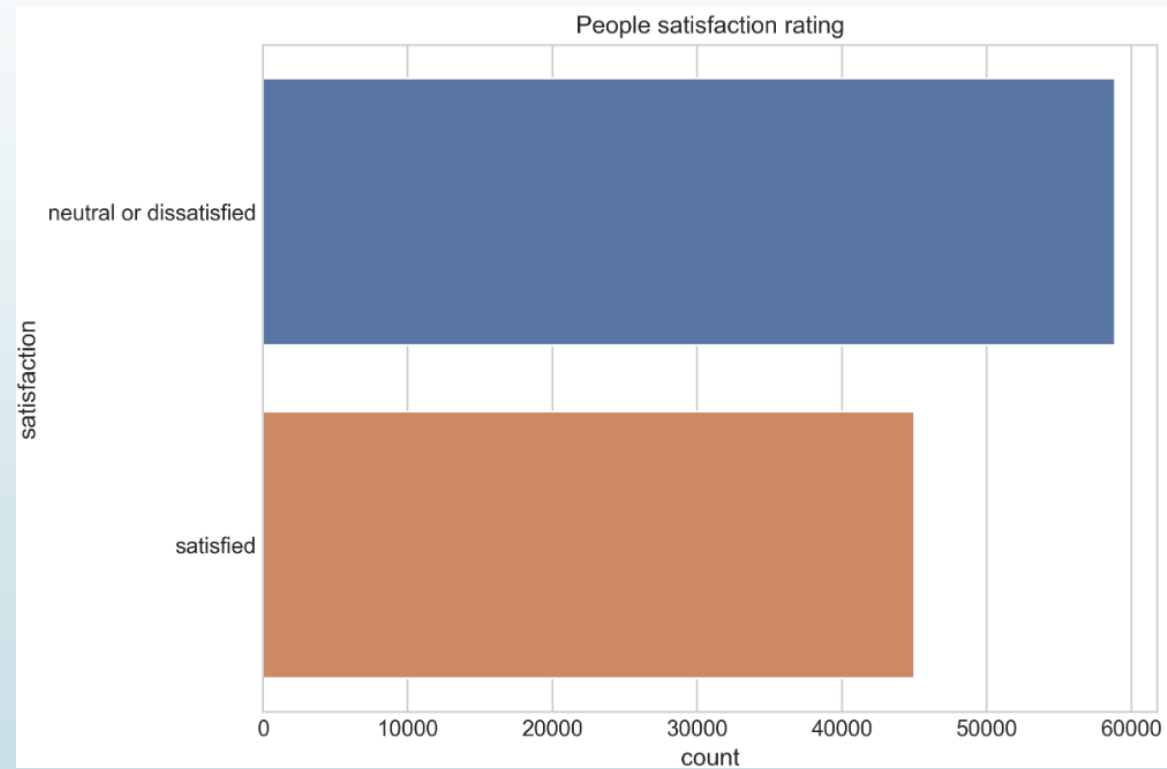
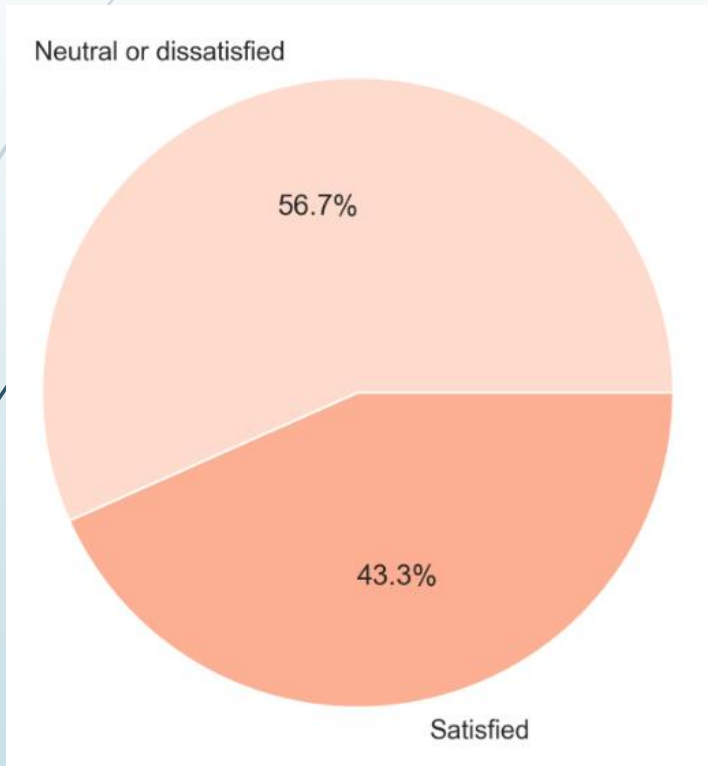
- Model için sütunlar arasında da korelasyon olup olmadığını da model için kontrol edilmesi gerekmektedir. Yüksek yoğunluklu korelasyon modellerin hatalı oluşturulmasını ve yanlış sonuç üretmesini sağlayabilir. Şekil 3.10’da ısı haritası ile hangi öz niteliklerin birbiriyle korelasyonunun olup olmadığı görülebilmektedir. Mavinin yoğun tonları korelasyonun yüksek olduğunu göstermektedir.
- “Arrival Delay in Minutes” ve “Departure Delay in Minutes” sütunlarının birbirlerini çok etkilediği ve korelasyonun yüksek olduğu da görülmektedir. Beklenildiği gibi geç kalkış yapan uçak geç varış yapacaktır. Dolayısıyla, “Arrival Delay in Minutes” sütununu silerek, aynı verileri tekrarlamadan doğru sonuca ulaşılabilmektedir.

```
[24]: corr_mat = data.corr()  
sns.heatmap(corr_mat, square = True, cmap = 'Blues')  
pass
```



Veri setinin görselleştirilmesi

Veri görselleştirme, bilgi ve verilerin görsel veya grafiksel temsilidir. Veri görselleştirme araçları; çizelgeler, grafikler ve haritalar gibi görsel unsurları kullanarak verilerdeki eğilimleri, aykırı değerleri ve kalıpları görmek ve anlamak için erişilebilir bir yol sağlar. Ayrıca, çalışanların veya işletme sahiplerinin verileri teknik olmayan kitlelere kafa karışıklığı olmadan sunmaları için mükemmel bir yol sağlar.



Modellerin Oluşturulması

Modelin oluşturulması için veri setinin “X” ve “y” olmak üzere ayrılması gerekmektedir. Şekil’de bulunan X değişkeninin tuttuğu değerler “satisfaction(memnuniyet)” özniteliği hariç diğer tüm öznitelik verileridir. Y değişkeni ise hedef sütunu yani memnuniyet değerlerini tutacak olan veri kümesidir. Aslında burada yapılan özniteliklerin hedef öznitelik ve diğerleri şeklinde ayrılması ve değişkenlere atanmasıdır.

```
# Split-out validation dataset
X = data.drop("satisfaction", axis = 1)
y = data["satisfaction"]
X_train, X_validation, Y_train, Y_validation = train_test_split(X, y, train_size=0.80,
                                                                random_state=1, shuffle=True)
```

Daha sonrasında X ve y veri setleri de eğitim verisi ve validasyon verisi olarak ayrılması gerekmektedir. Yine aynı Şekil’de görüldüğü gibi bu ayırma işlemi %80 olarak tanımlanmıştır. Yani verilerin yüzde 80’i eğitim verileri ve %20’si ise validasyon yani doğrulama ya da test verisi olacaktır.

Aşağıda bulunan kod dizisi ile önceden Scikit-Learn kütüphanesinden çağrılan algoritmalar daha sonra kullanılmak üzere bir değişkene liste olarak atanmışlardır.

```
# Spot Check Algorithms
models = []
models.append(('LR', LogisticRegression(solver='liblinear', multi_class='ovr')))
models.append(('LDA', LinearDiscriminantAnalysis()))
models.append(('KNN', KNeighborsClassifier()))
models.append(('CART', DecisionTreeClassifier()))
models.append(('NB', GaussianNB()))
```

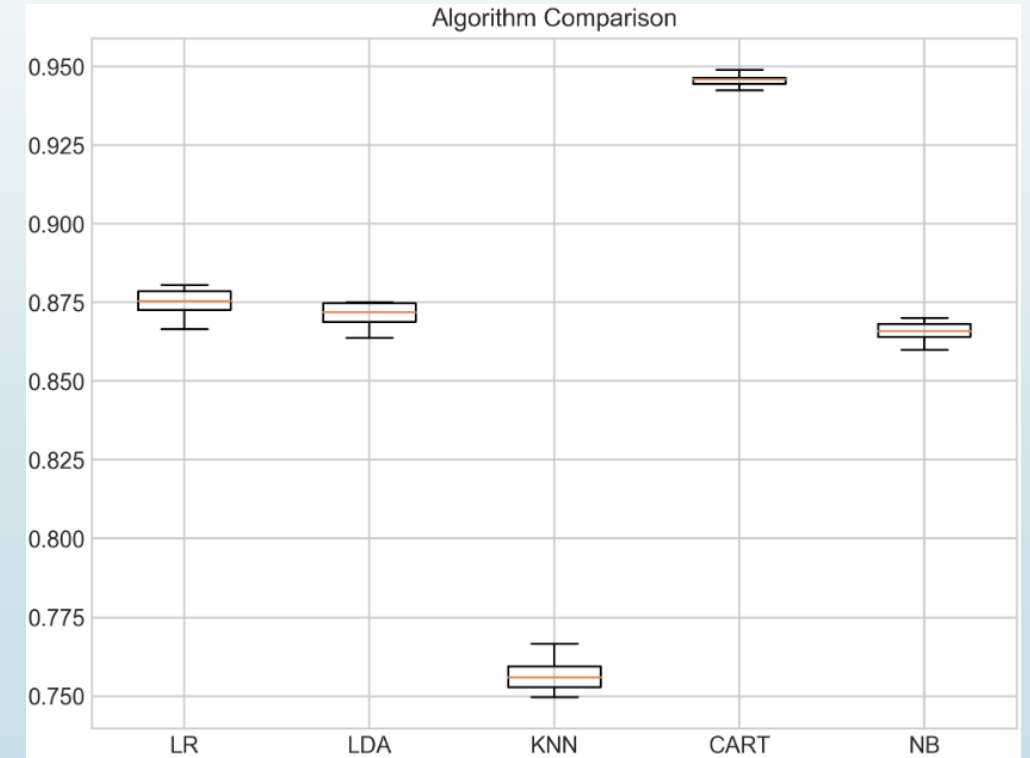
Modellerin Çalıştırılması

Listeye alınan algoritmalar aşağıdaki fonksiyon ile çalıştırılmaktadır. Her biri tek tek daha öncesinde %80 olarak ayrılan eğitim verisiyle eğitilmektedir. Ve sonrasında farklı algoritmalar ile oluşan her bir modelin tahminleri, accuracy (doğruluk) adlı fonksiyon ve %20 olarak ayrılan test verisiyle kontrol edilmektedir. Bulunan sonuç skorları ile en iyi sonucu veren algoritmanın CART (DecisionTreeClassifier) yani karar ağacı olduğu anlaşılmaktadır.

```
# evaluate each model in turn
results = []
names = []
for name, model in models:
    kfold = StratifiedKFold(n_splits=10, random_state=1, shuffle=True)
    cv_results = cross_val_score(model, X_train, Y_train, cv=kfold, scoring='accuracy')
    results.append(cv_results)
    names.append(name)
    print('%s: %f (%f)' % (name, cv_results.mean(), cv_results.std()))
```

```
LR: 0.874956 (0.004616)
LDA: 0.871010 (0.003905)
KNN: 0.756866 (0.005169)
CART: 0.945527 (0.001787)
NB: 0.865693 (0.003199)
```

Çıkan sonuçları daha iyi anlamak için kutu diyagramı ile görselleştirilmesi yapılmıştır.



En İyi Modelin Değerlendirilmesi

En iyi algoritma seçildikten sonra tekrar karar ağacı algoritması ile model oluşturulup bu model bir değişkene atanmıştır. Daha sonra bu model kullanılarak X validasyon verilerinden tahminler üretilmiştir. X validasyon, veri setinden %20 olarak ayırdığımız test verisidir ve bu veri setinde “satisfaction” (memnuniyet) hedef sütunu bulunmamaktadır. Oluşturulan model ile X validasyon veri setinden bir memnuniyet sonucunun tahminlerini oluşturması sağlanmış ve çıkan tahminler y_tahmin değişkenine atanmıştır. Ve sonrasında X validasyon veri setinden üretilen y_tahmin verisi ile gerçek memnuniyet değerlerini saklayan y validasyon değerleri karşılaştırılmıştır. Modelin tahminleri ile gerçek değerlerin %94 oranında uyduğu sonucu ortaya çıkmıştır

```
model = DecisionTreeClassifier()
model.fit(X_train, Y_train)
```

```
▼ DecisionTreeClassifier
DecisionTreeClassifier()
```

```
y_tahmin = model.predict(X_validation)
print(y_tahmin)
```

```
['satisfied' 'neutral or dissatisfied' 'neutral or dissatisfied' ...
 'satisfied' 'neutral or dissatisfied' 'satisfied']
```

```
print(np.mean(y_tahmin==Y_validation))

0.9437466916895241
```

Accuracy score ile Confusion matrix ve Classification Report fonksiyonları da modelin durumu hakkında bilgi verilmek üzere kullanılmıştır.

```
model = DecisionTreeClassifier()
model.fit(X_train, Y_train)
predictions = model.predict(X_validation)
print('Accuracy Score(Doğruluk Skoru):')
print(accuracy_score(Y_validation, predictions))
print('*****')
print('Confusion Matrix(Hata Matrisi):')
print(confusion_matrix(Y_validation, predictions))
print('*****')
print('classification_report(Sınıflandırma Raporu):')
print(classification_report(Y_validation, predictions))
```

```
Accuracy Score(Doğruluk Skoru):
0.9443241422453202
```

```
*****
```

```
Confusion Matrix(Hata Matrisi):
```

```
[[11254  621]
 [ 536 8370]]
```

```
*****
```

```
classification_report(Sınıflandırma Raporu):
```

	precision	recall	f1-score	support
neutral or dissatisfied	0.95	0.95	0.95	11875
satisfied	0.93	0.94	0.94	8906
accuracy			0.94	20781
macro avg	0.94	0.94	0.94	20781
weighted avg	0.94	0.94	0.94	20781

En İyi Modelin Değerlendirilmesi

- Accuracy score (doğruluk skoru) değeri %94 olarak çıkmıştır.
- Hata matrisi, veri setindeki var olan durum ile sınıflandırma modelinin doğru ve yanlış tahminlerinin sayısını tablo olarak göstermektedir. Hata matrisi, kaç kişi memnunken model memnun yani doğru değeri verdiğini ya da kaç kişi memnun değil yani yanlış değeri verdiğini detaylı olarak göstermektedir. Hata matrisinin ilk satırı incelendiğinde toplamda (11254 + 621) 11875 kişinin memnun olmadığı ya da nötr olduğu görülmektedir. Ayrıca model 11875 kişiden 11254'ünün memnun olmadığını ve kalan 621 kişinin memnun olduğunu göstermektedir. Buradan çıkarılacak sonuç, model memnun olmayan 621 kişi için doğru değeri bulamamış ve memnun olarak göstermiştir.
- F1-skoru, bir sınıflandırıcının Kesinlik (Precision) ve Duyarlılık (Recall) değerlerini harmonik ortalamalarını alarak tek bir metrikte birleştirmektedir. İki sınıflandırıcının performansını karşılaştırmak için kullanılmaktadır. Classification report (sınıflandırma raporu) fonksiyonun çıktısı olarak hem memnun değil ya da nötr hem de memnun olma ikili karar durumları için f1 skorları üretilmiştir. Model memnun değil ya da nötr kişiler için %95 oranında doğru karar üretebilirken memnun olan kişiler için %94 oranında doğru karar verebilmiştir. Model memnun olmayan ya da nötr kişiler için memnun olan kişilere göre daha doğru tahmin modeli oluşturmuştur.

	Tahminlenen (Predicted)	
Gerçekleşen (Actual)	True Positives (TP)	False Negatives (FN)
	False Positives (FP)	True Negatives (TN)

Confusion Matrix

$$F_1 = 2 * \frac{precision * recall}{precision + recall}$$

Sonuç ve Öneriler

Bu çalışmada, makine öğrenimi algoritmalarını ve bu algoritmalar ile verinin doğasını daha derinlemesine incelenmiştir. Sınıflandırma modelleri için birçok algoritma kullanılmaktadır ama her veri seti için en iyi sonucu veren ve en iyi tahmin üretebilen algoritmanın da bulunması gerektiği vurgulanmıştır.

Bu çalışma da denetimli ve denetimsiz makine öğrenimi konularından bahsedilmiştir. Detaylı olarak denetimli makine öğrenmesi algoritmalarından sınıflandırma algoritmaları incelenmiştir. Bununla ilgili bir havayolu şirketi anketinin veri seti kullanılarak 5 farklı makine öğrenmesi algoritmaları ile model oluşturulmuş ve modellerin sonuçları analiz edilmiştir. En iyi sonucu veren algoritma ile eğitim veri seti kullanılarak tahminler üretilmiş; bu tahminleri test veri setiyle kıyaslanarak doğruluk skoru, hata matrisi ve f1 skorları bulunmuş ve bu skorlar ile modelin hangi hedef öznelik için daha doğru çalışıp çalışmadığı belirlenmiştir.

Doğru bir model kurulumu için veri setinin ön işleme yapılarak verinin hazırlığının yapılması gerekliliği vurgulanmıştır. Veri setinin ve özneliklerin daha iyi anlaşılması için veri setiyle çeşitli görselleştirmeler yapılması tavsiye edilmiştir.

Bu çalışma ile makine öğreniminin farklı bir alanda kullanımı ve kullanılmak istenen veri setiyle en iyi algoritma ve tahmin modelini bulmayı ve bu model ile en iyi oranda tahmin üretilbileceği gösterilmiştir.

**DİNLEDİĞİNİZ İÇİN TEŞEKKÜR
EDERİM...**