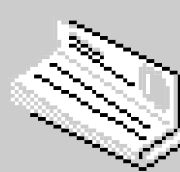
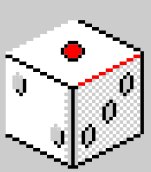
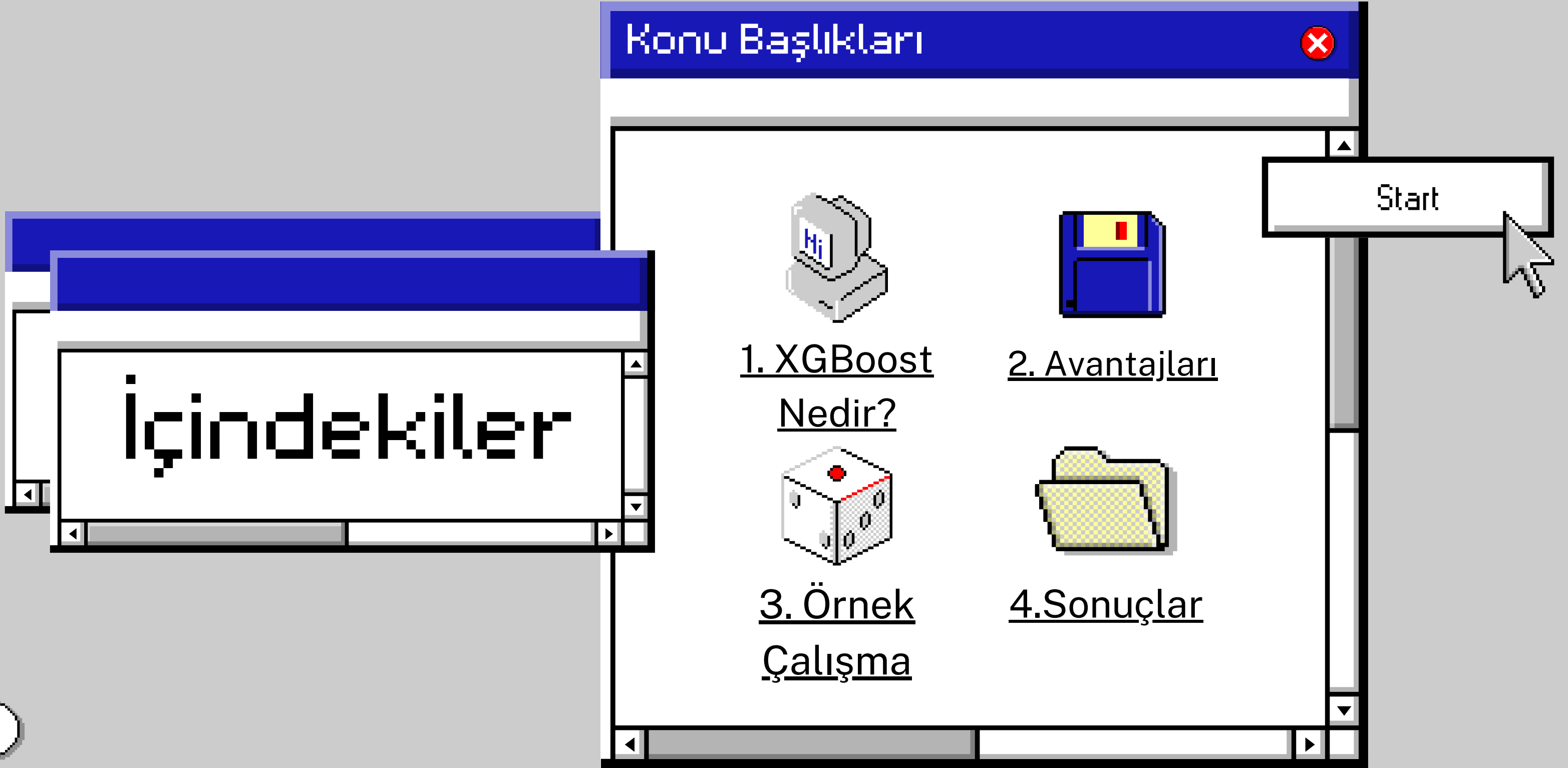


XGBoost Modeli



Veri Sınıflandırma

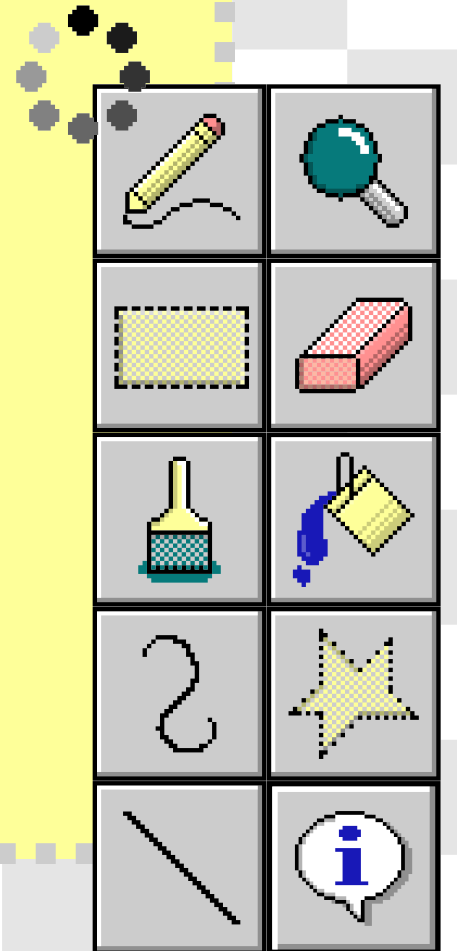






1.XGBoost Nedir?

[Back to Agenda Page](#)



XGBoost Nedir?



Extreme Gradient Boosting (XGBoost) modelinin nasıl çalıştığını anlamamız için önce kolektif öğrenme yöntemleri nelerdir onları hatırlayıp, daha sonra ise XGBoost yöntemini bu yöntemlerden yola çıkarak konu hakkında sağlam bir kavrayış inşa etmeye çalışacağız.

Kolektif Öğrenme Yöntemleri:

- karar ağaçları,
- bagging,
- random forest,
- stacking,
- boosting,
- gradient boosting

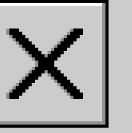
Karar Ağaçları



Temel fikir, giriş verisinin bir kümeleme algoritması yardımıyla tekrar tekrar gruplara bölünmesine dayanır. Grubun tüm elemanları aynı sınıf etiketine sahip olana kadar kümeleme işlemi derinlemesine devam eder. Karar ağaçları verileri alt kümelemek için en iyi ayrımı bulmaya çalışır ve genellikle sınıflandırma ve regresyon ağacı algoritması ile eğitilirler.

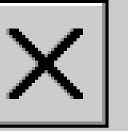
Karar ağaçları, muhtemelen bulabileceğiniz en kolay yorumlanabilir makine öğrenmesi algoritmasıdır. Doğru tekniklerle kullanıldığı zaman çok güçlü bir algoritma haline dönüşebilir.

Bagging



Bagging, modelimizde yüksek varyansı düşürmeye yarar. Verisetimizin her seferinde farklı bir altkümelerini alarak onu eğitir ve en son tahlilde tüm baz modellerin çıktılarının ortalamasını alarak (sınıflandırma yapılıyorsa tüm baz modeller arasında her birini eşit ağırlıkta kabul ettiği bir oylama yaparak) bir kolektif öğrenme modeli oluşturur. Baz modelleri oluştururken verisetimizin (paralel şekilde) farklı altkümelerini seçerek yola çıkarız.

Boosting



Boosting, yönteminde ise yine veri setimizin bir alt kümesini alırız. Yine o alt kümenin üzerinde bir model kurarız. Fakat bundan sonra kuracağımız başka bir baz model (weak learner) ilkinden bağımsız değil, aksine ilk modeli geliştirmek üzere çalışır.

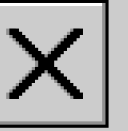
Bagging kullanılırken her bir baz model sonuca eşit etki ederken, boosting ilk etapta hatalı tahmin edilen verilere daha fazla ağırlık vererek bir sonraki baz modelin bu hataları düzeltmesini amaçlar.

Stacking



Stacking ise yukarıdaki diğer iki kolektif öğrenme modelinin aksine verisetinin altkümesinin değil verisetinin tamamı üzerinde kurulmuş farklı modelleri birleştirir. Boosting yönteminde baz modeller -dizi halinde- kendinden önce gelen baz modelleri geliştirmeye uğraşırken, stacking yönteminde her bir baz model kendi içinde en iyi sonuca ulaşmaya çalışır.

Gradient Boosting













Gradient Boosting, ensemble (toplu) bir yöntemdir, yani birkaç modelden gelen tahminleri tek modelde birleştirmenin bir yoludur. Bu işlemi, her bir tahminleyiciyi sırayla alarak ve bir öncekinin hatasına göre modelleyerek (daha iyi performans gösteren tahminleyicilere daha fazla ağırlık vererek) yapar:

- Orijinal verileri kullanarak ilk modelini kurar.
- Daha sonra ilk modelin kalıntılarıyla ikinci bir model kurar.
- Model 1 ve 2'nin toplamını kullanarak üçüncü bir model kurar.

Gradient boosting, bir gradient descent algoritması kullanarak kayıp işlevini minimize eder.



2. Avantajları



Neden Bu Kadar İyi Sonuç Veriyor?



Sistem Optimizasyonu:

- Parallelleştirme(Parallelization)

Parallelleşme her ağacın inşası sırasında çok düşük bir seviyede gerçekleşir. Ağacın her bir bağımsız dalı ayrı ayrı eğitilir. Hiper parametre ayarı, ağaç başına birçok dal ve model başına birçok ağaç ve hiper parametre değeri başına birkaç model ve test edilecek birçok hiper parametre değeri gerektirir.

- Ağaç Budama (Tree Pruning)

Bu noktada Gradient Boost algoritmasıyla bir karşılaştırma yapalım. GBM, bölünmede negatif bir kayıpla karşılaştığında bir düğümü bölmeyi durdurur. Bu yüzden açgözlü bir algoritmadır. Diğer taraftan XGBoost, belirtilen max_depth parametresine kadar bölmeler yapar ve ağacı geriye doğru budamaya başlar. Ardından pozitif kazanım olmayan bölmeleri kaldırır.

- Donanım Optimizasyonu

Gradyan istatistiklerini depolamak için her iş parçacığına dahili tamponlar tahsis ederek önbellek farkındalığı sağlayarak gerçekleşir. Belleğe sığmayan büyük veri çerçevelerini işlerken kullanılabilir disk alanını optimize eder.

Neden Bu Kadar İyi Sonuç Veriyor?



Algoritmik Kazanımlar:

- Düzenleme(Regularization)

Bu algoritmanın baskın bir faktörü olarak kabul edilir. Düzenleme, modelin aşırı uyumluluğundan kurtulmak için kullanılan bir tekniktir.

- Çapraz Doğrulama(Cross Validation):

Normalde fonksiyonu Scikit-learn'den içe aktararak çapraz doğrulamaya alışmışızdır, ancak XGBoost içerisine monte edilen çapraz doğrulama fonksiyonu ile otomatik olarak cross validation yapmaktadır.

- Eksik Değer(Missing Value)

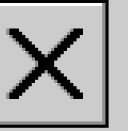
Eksik değerleri kaldırabilecek şekilde tasarlanmıştır. Eksik değerlerdeki eğilimleri bulur ve yakalar.

- Esneklik(Flexibility):

Modelin performansını değerlendirmek için kullanılan işlevdir ve ayrıca kullanıcı tanımlı doğrulama ölçütlerini işleyebilir.

özelliklerinden dolayı diğer algoritmalara nazaran daha iyi sonuçlar vermektedir.

XGBoost'un Hiperparametreleri



n_estimators: Kullandığımız ağaç sayısını temsil eder. Büyük veri setlerinde 100 küçük-orta ölçekte veri setlerinde 1000 tercih edilebilir.

subsample: Verisetindeki örneklerin(sample,observation) her bir karar ağacında hangi yüzde ile kullanılacağını tayin eder. Uygun bir yüzde verideki rastgeleliği (randomness) artırıp varyansı düşürmeye yardımcı olabilir.

gamma: Öğrenme hızı gibi aşırı öğrenmeyi önleyici bir diğer faktör. Değeri verilere göre değişmektedir. Gamma değeri ne kadar yüksekse tutuculuk o kadar yüksek olacaktır. Grid Search ile bu parametreyi incelemekte fayda var.

max_depth: Bir ağacın maksimum derinliği. Bu değeri artırırsanız modelinizi daha karmaşık hale getirirsiniz ve aşırı öğrenmenin önünü açarsınız. Belirli sabit bir değeri yoktur. Verilerin boyutuna göre belirlenmelidir. Cross Validation kullanarak ayarlamalar yapabilirsiniz.

learning_rate: En düşük loss function'ı elde etmek için gradient değerimizi adım adım düşürürken yararlandığımız, attığımız her bir adımı uygun şekilde ölçeklendirmeye yarayan hiperparametre. Genellikle 0.1 ile 0.01 arasında değerler kullanılır.

Uygulama Adımları



- 1- Veri setini düzenleme ve ön işleme yapımı: Bu adım, veri setini ikiye bölmeyi (eğitim ve test verileri için), verileri normalleştirmeyi veya standartlaştırmayı, veri özelliklerini önceden seçmeyi ve düzenlemeyi gibi işlemleri içerir.
- 2- XGBoost modelini oluşturup hiperparametrelerini ayarlama: Bu adım, modelin öğrenme oranını, ağacın derinliğini gibi özellikleri içerir.
- 3- Modeli eğitme: Verisetini kullanarak modelin öğrenme sürecini başlatır. XGBoost, veri setinden öğrenir ve bir dizi zayıf model oluşturur. Her zayıf model, önceki modellerin tahminlerini düzeltir ve yeni bir model oluşturur. Bu işlem, belirli bir döngü sayısı kadar (örneğin, `n_estimators` parametresi tarafından belirtilen sayı kadar) tekrarlanır.
- 4- Model değerlendirme: Eğitim süreci tamamlandıktan sonra, modeli değerlendirebilir ve tahminlerini test verisiyle karşılaştırabilirsiniz. Bu, modelin performansını ölçmenizi ve gerektiğinde hiperparametreleri ayarlamanızı sağlar.

Veriseti



Verisetimiz 120,000'den fazla havayolu yolcusundan alınan müşteri memnuniyeti puanları üzerinedir. Her bir yolcu için uçuşları ve seyahat türü hakkındaki bilgilerin yanı sıra temizlik, konfor, hizmet ve genel deneyim gibi farklı faktörler müşteri memnuniyetini etkilemektedir.

Müşteri memnuniyetini, kategorileştirmek için her bir değer ortalaması üzerinden yola çıkılarak kategorikleştirilmiştir.



Veriseti



Yanda verisetine ait değişkenlerin sütun isimlerini, verisetinin satır ve sütun sayılarını görmekteyiz.

Bunun yanı sıra verisetinde boş değer olup olmadığını kontrol ettik ve boş değer varsa hangi değişkende olduğunu tespit ettik.

Gördüğünüz gibi "Arrival Delay" değişkeninde 393 adet boş değerimiz vardır.

```
-- Dataset Columns --
['Gender', 'Age', 'Customer Type', 'Type of Travel', 'Class', 'Flight Distance', 'Departure Delay', 'Arrival Delay', 'Departure and Arrival Time Convenience', 'Ease of Online Booking', 'Check-in Service', 'Online Boarding', 'Gate Location', 'On-board Service', 'Seat Comfort', 'Leg Room Service', 'Cleanliness', 'Food and Drink', 'In-flight Service', 'In-flight Wifi Service', 'In-flight Entertainment', 'Baggage Handling', 'Satisfaction']

-- Dataset Shape--
(129880, 23)

-- Is There Any NaN Values? --
True

-- NaN Values --
Gender                                0
Age                                  0
Customer Type                        0
Type of Travel                       0
Class                                0
Flight Distance                      0
Departure Delay                      0
Arrival Delay                        393
Departure and Arrival Time Convenience 0
Ease of Online Booking               0
Check-in Service                     0
Online Boarding                      0
Gate Location                        0
On-board Service                     0
Seat Comfort                         0
Leg Room Service                     0
Cleanliness                          0
Food and Drink                       0
In-flight Service                    0
In-flight Wifi Service                0
In-flight Entertainment               0
Baggage Handling                     0
Satisfaction                         0
dtype: int64
```

Veriseti



Yanda verisetimizin nümerik değişkenlerine ait betimleyici istatistik değerlerini ve bazı değişkenlerle ilgili ilk 5 satırın değerlerini görmekteyiz.

-- Dataset First 5 Observation --

	Gender	Age	Customer Type	Type of Travel	Class	Flight Distance	Departure Delay	Arrival Delay	Departure and Arrival Time Convenience	Ease of Online Booking	...	On-board Service	Seat Comfort	Leg Room Service	Cleanliness	Food and Drink	In-flight Service
0	Male	48	First-time	Business	Business	821	2	5.0	3	3	...	3	5	2	5	5	5
1	Female	35	Returning	Business	Business	821	26	39.0	2	2	...	5	4	5	5	3	5
2	Male	41	Returning	Business	Business	853	0	0.0	4	4	...	3	5	3	5	5	3
3	Male	50	Returning	Business	Business	1905	0	0.0	2	2	...	5	5	5	4	4	5
4	Female	49	Returning	Business	Business	3470	0	1.0	3	3	...	3	4	4	5	4	3

-- Dataset Describe --

	Age	Flight Distance	Departure Delay	Arrival Delay	\
count	129880.000000	129880.000000	129880.000000	129487.000000	
mean	39.427957	1190.316392	14.713713	15.091129	
std	15.119360	997.452477	38.071126	38.465650	
min	7.000000	31.000000	0.000000	0.000000	
25%	27.000000	414.000000	0.000000	0.000000	
50%	40.000000	844.000000	0.000000	0.000000	
75%	51.000000	1744.000000	12.000000	13.000000	
max	85.000000	4983.000000	1592.000000	1584.000000	

	Departure and Arrival Time Convenience	Ease of Online Booking	\
count	129880.000000	129880.000000	
mean	3.057599	2.756876	
std	1.526741	1.401740	
min	0.000000	0.000000	
25%	2.000000	2.000000	
50%	3.000000	3.000000	
75%	4.000000	4.000000	
max	5.000000	5.000000	

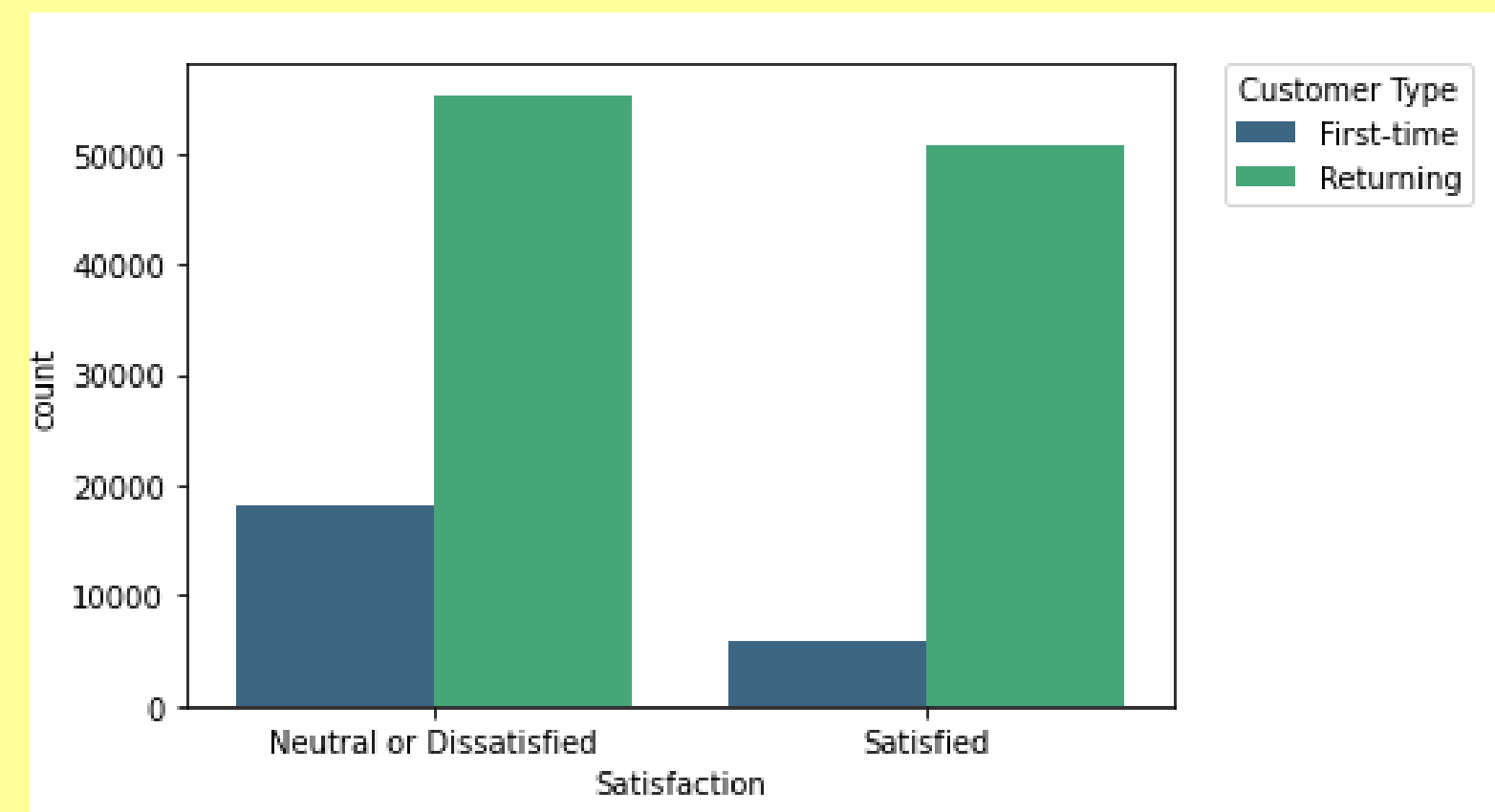
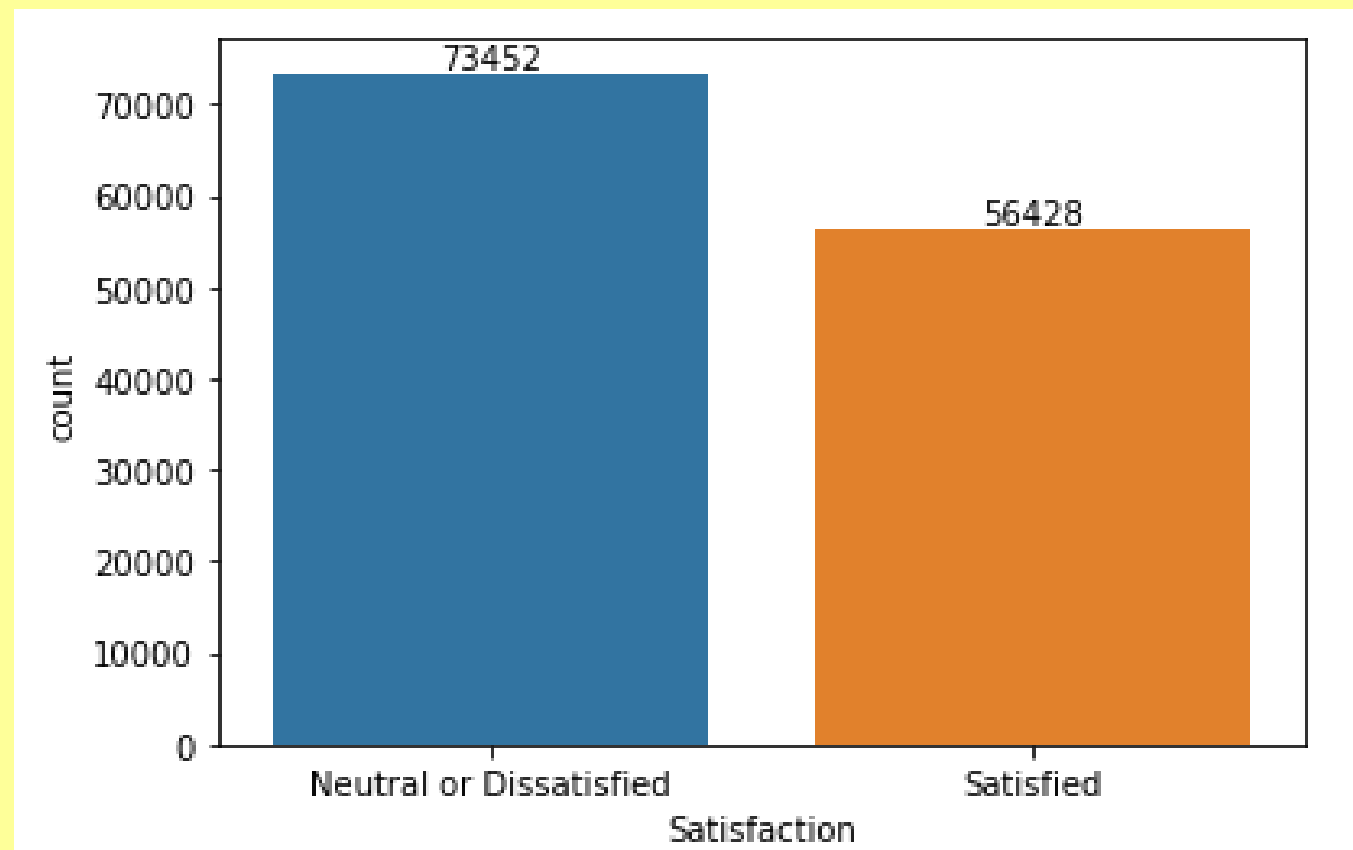
	Check-in Service	Online Boarding	Gate Location	On-board Service	\
count	129880.000000	129880.000000	129880.000000	129880.000000	
mean	3.306267	3.252633	2.976925	3.383023	
std	1.266185	1.350719	1.278520	1.287099	
min	0.000000	0.000000	0.000000	0.000000	
25%	3.000000	2.000000	2.000000	2.000000	
50%	3.000000	3.000000	3.000000	4.000000	
75%	4.000000	4.000000	4.000000	4.000000	
max	5.000000	5.000000	5.000000	5.000000	

	Seat Comfort	Leg Room Service	Cleanliness	Food and Drink	\
count	129880.000000	129880.000000	129880.000000	129880.000000	
mean	3.441361	3.350878	3.286326	3.204774	
std	1.319289	1.316252	1.313682	1.329933	
min	0.000000	0.000000	0.000000	0.000000	
25%	2.000000	2.000000	2.000000	2.000000	
50%	4.000000	4.000000	3.000000	3.000000	
75%	5.000000	4.000000	4.000000	4.000000	
max	5.000000	5.000000	5.000000	5.000000	

Görselleştirme



Veri görselleştirme kısmında bağımlı değişkenimize (müşteri memnuniyeti, satisfaction) ait birkaç grafik çizdirdik. Aşağıda memnun olup olmama ve bu memnuniyetin ilk uçuşta olup olmadığını gösteren grafikler bulunmaktadır.

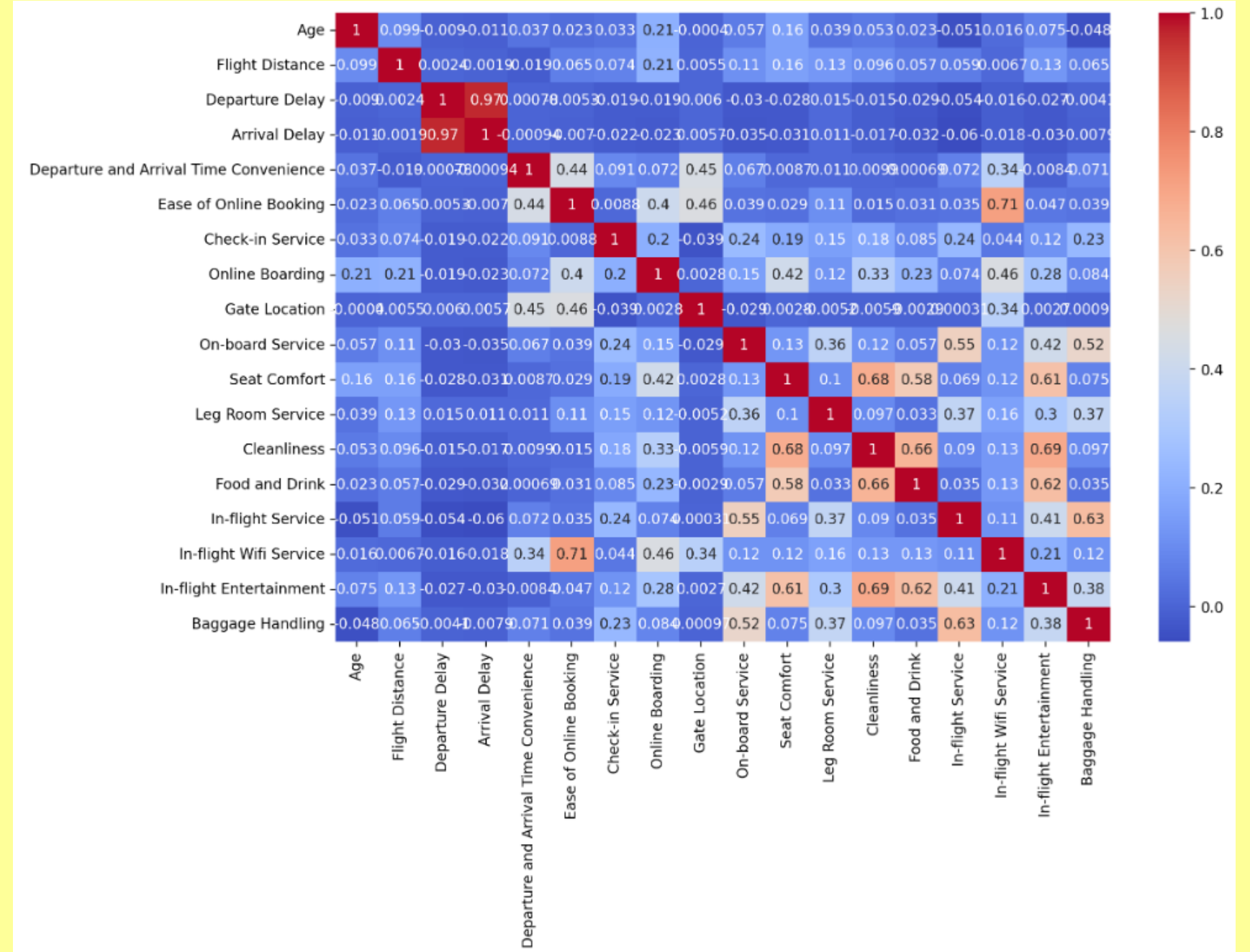


Görselleştirme



Bu grafikte ise değişkenler arası korelasyon değerlerini görmekteyiz.

En yüksek korelasyon değeri, "Arrival Delay" ve "Departure Delay" arasındadır. Korelasyon değeri ise 0.97'dir.



Boş Veri Doldurma



"Arrival Delay" değişkeninde boş değişkenimizin olduğunu önceki slaytlarda göstermiştik. Boş değerleri değişkenin ortalaması ile doldurup tekrar boş veri olup olmadığını kontrol ettik.

Ardından tüm kategorik değerleri get_dummies() fonksiyonunu kullanarak her bir kategoriye ayrı değişken olarak aldık.

```
In [14]: df['Arrival Delay'] = df['Arrival Delay'].fillna(value = df['Arrival Delay'].mean())

In [15]: df.isnull().sum().sum()

Out[15]: 0

In [16]: df = pd.get_dummies(df, drop_first=True)

In [17]: df.head()

Out[17]:
```

	Age	Flight Distance	Departure Delay	Arrival Delay	Departure and Arrival Time Convenience	Ease of Online Booking	Check-in Service	Online Boarding	Gate Location	On-board Service	...	In-flight Service	In-flight Wifi Service	In-flight Entertainment	Baggage Handling	Gender_Male
0	48	821	2	5.0	3	3	4	3	3	3	...	5	3	5	5	1
1	35	821	26	39.0	2	2	3	5	2	5	...	5	2	5	5	0
2	41	853	0	0.0	4	4	4	5	4	3	...	3	4	3	3	1
3	50	1905	0	0.0	2	2	3	4	2	5	...	5	2	5	5	1
4	49	3470	0	1.0	3	3	3	5	3	3	...	3	3	3	3	0

Modelin Kurulması



Modelimiz için bağımlı değişken olarak müşteri memnuniyetini ele aldığımız için "Satisfaction_Satisfied" değişkenini y değişkeni olarak tanımlıyoruz.

```
X = df.drop('Satisfaction_Satisfied',axis=1)
y = df['Satisfaction_Satisfied']
```

```
X_train, X_test, y_train, y_test = train_test_split(X,y,test_size=0.30, random_state=101)
```

```
scaler = StandardScaler()
scld_X_train = scaler.fit_transform(X_train)
scld_X_test = scaler.transform(X_test)
```

```
parameters = {
    'n_estimators': [100, 500],
    'subsample': [0.8, 1.0],
    'gamma' : [0,1,5],
    'max_depth': [3, 4, 5],
    'learning_rate': [0.1, 0.3]}
```

```
xgboost_model = XGBClassifier()
xgboost_cv = GridSearchCV(xgboost_model, parameters, cv = 3, n_jobs = -1, verbose = 2)
```

```
xgboost_cv.fit(scld_X_train,y_train)
```

Fitting 3 folds for each of 72 candidates, totalling 216 fits

4. Sonuçlar



Recall - Gerçekteki sınıfın ne kadarı doğru tahmin edildi. 1 sınıfına bakılacak olursa yaklaşık %94'sı doğru tahmin edilmiş.

Precision - Tahmin edilen sınıfın ne kadarı doğru. Yine 1 sınıfa bakılırsa yaklaşık %94'ü doğru diyebiliriz.

F1 score ise recall ve precision değerlerinin harmonik ortalaması.

```
best = xgboost_cv.best_params_  
best
```

```
{'gamma': 5,  
 'learning_rate': 0.1,  
 'max_depth': 5,  
 'n_estimators': 500,  
 'subsample': 0.8}
```

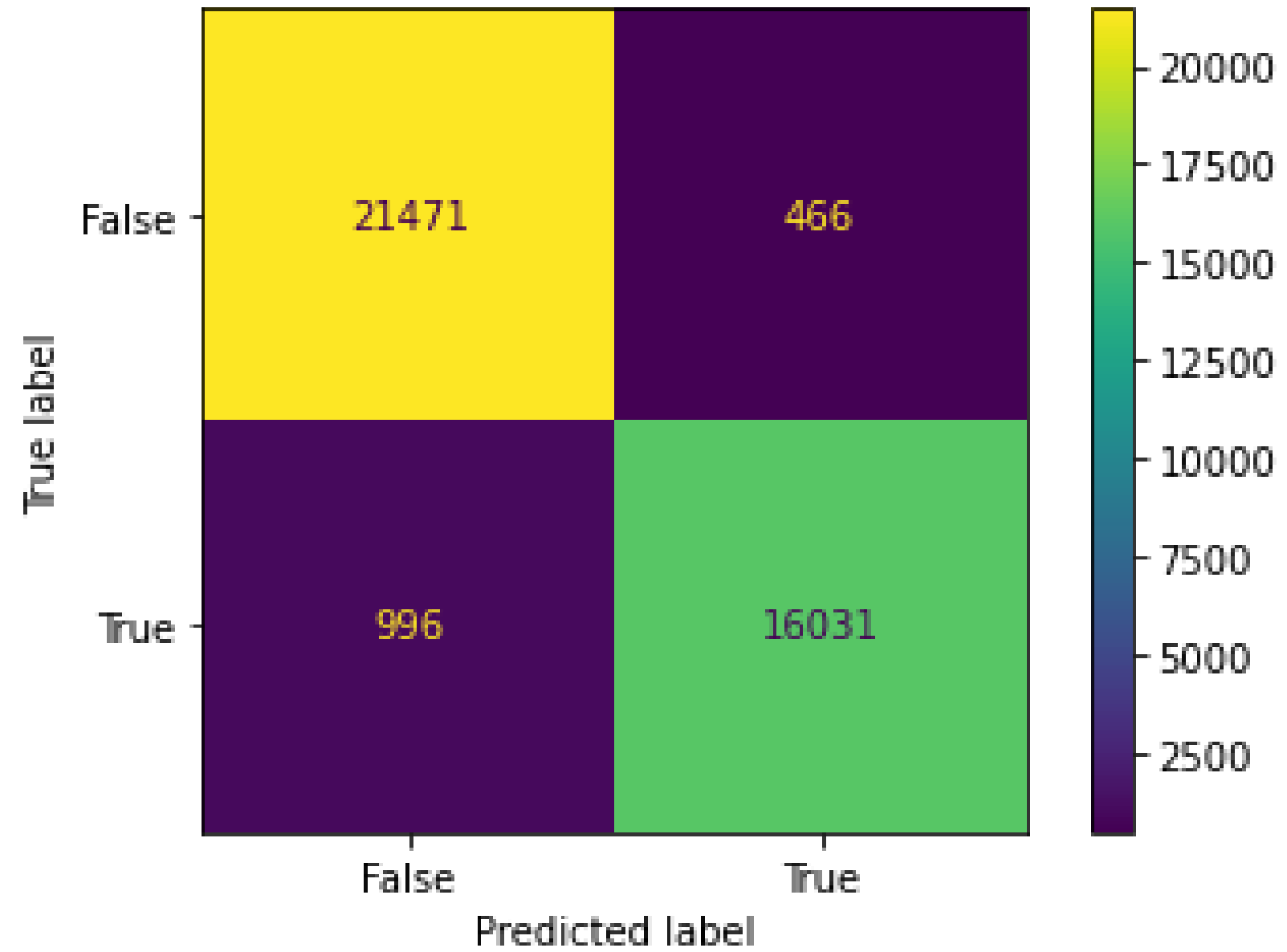
```
xgboost_model2 = XGBClassifier(gamma=5, learning_rate = 0.1, max_depth = 5, n_estimators = 500, subsample = 0.8)  
xgb_tunned = xgboost_model2.fit(scl_d_X_train, y_train)
```

```
y_pred = xgb_tunned.predict(scl_d_X_test)
```

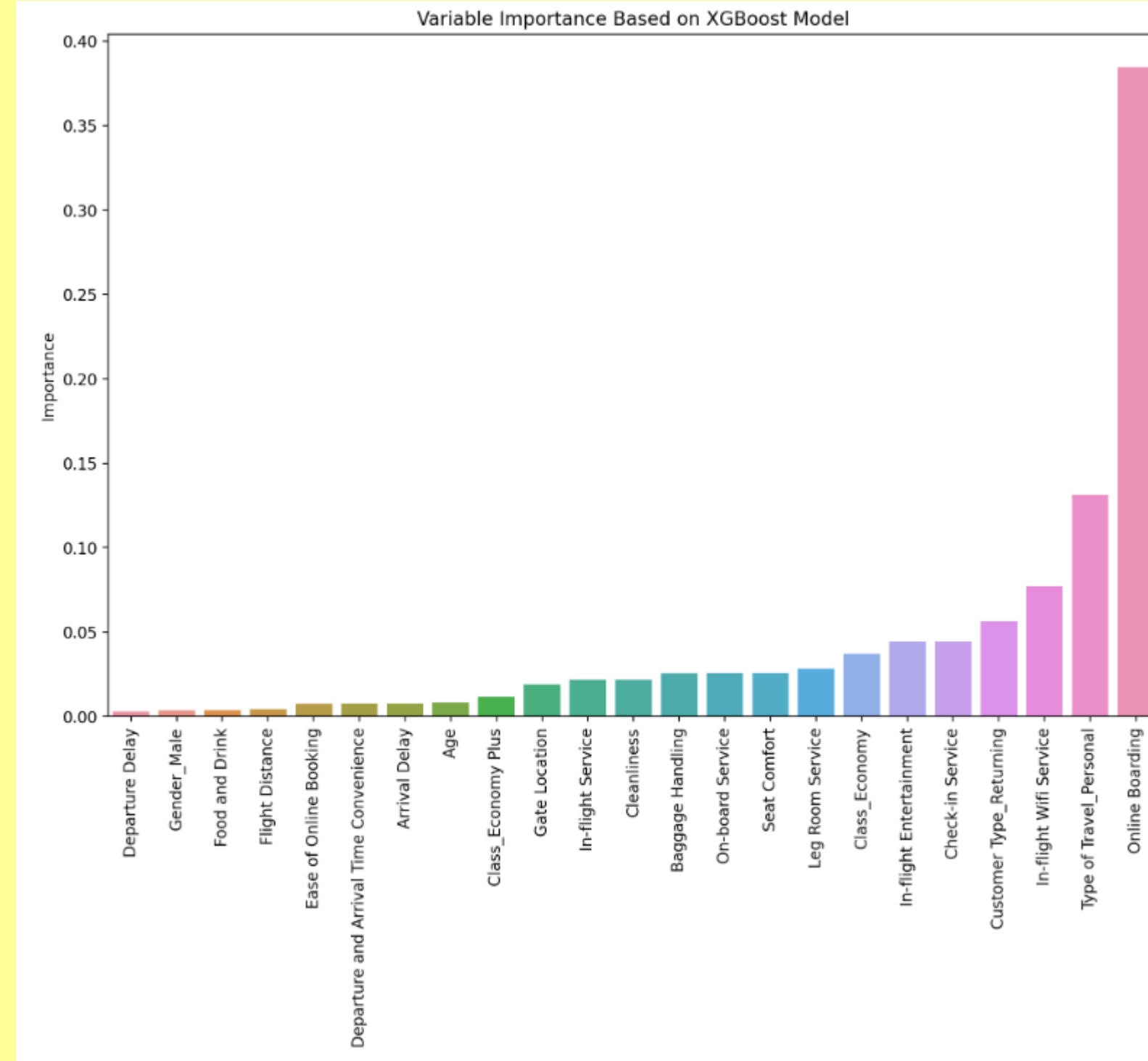
```
print(f'Classification Report: {classification_report(y_test,y_pred)}')  
confusion_matrix = metrics.confusion_matrix(y_test,y_pred)  
cm_display = metrics.ConfusionMatrixDisplay(confusion_matrix = confusion_matrix, display_labels = [False, True])  
  
cm_display.plot()  
plt.grid(False)  
plt.show()
```

Classification Report:		precision	recall	f1-score	support
0	0.96	0.98	0.97		21937
1	0.97	0.94	0.96		17027
accuracy			0.96		38964
macro avg		0.96	0.96	0.96	38964
weighted avg		0.96	0.96	0.96	38964

4. Sonuçlar



4. Sonuçlar



4. Sonular



XGBoost, diğ er makine  ğrenmesi modellerine g re daha hızlı ve daha optimum sonu verdiğinden dolayı tercih edilmektedir. alıřmamızda da bu modeli kullanıp etkin sonular aldık. Bu sonuları yorumlayacak olursak;

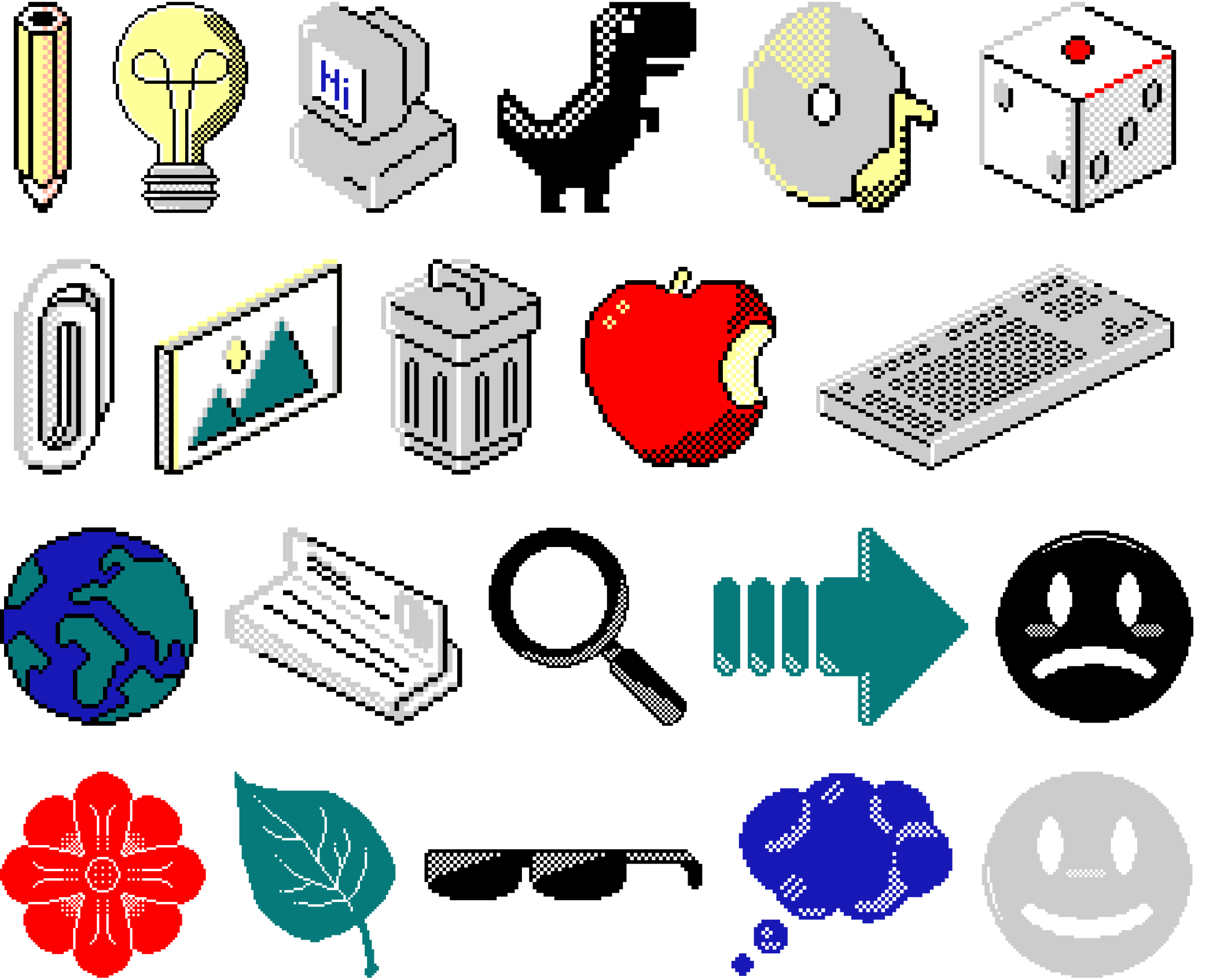
Modelimizde bağımlı değıřkenimize en ok etki eden bağımsız değıřkenin "Online_Boarding" olduėu tespit edilmiřtir.

Modelimiz bařarı metriğı olan "accuracy" değıřerine bakıldığında %96'lık bir bařarı saėladıėı tespit edilmiřtir.

Genel olarak m řteri memnuniyetine bakıldığında "memnun değıřim" kategorisi yaklařık 22 bin, "memnunum" kategorisi yaklařık 17 bin olduėu tespit edilmiřtir. Buna g re yolcuların memnun olmadıėını s yleyebiliriz.

Kaynakça

- <https://www.datascienceearth.com/extreme-gradient-boosting-xgboost/>
- <https://www.veribilimiokulu.com/xgboost-nasil-calisir/>
- https://teknoloji.org/kaggle-yarismalarinin-en-populer-algoritmasi-xgboost/#XGBoost_Nasil_Calisir
- <https://www.datascienceearth.com/boosting-algoritmaları/>





Teşekkürler!

Bizi dinlediğiniz için teşekkür ederiz.

