

BAŞKENT ÜNİVERSİTESİ

EEM 612 ÖRÜNTÜ TANIMA VE

MAKİNE ÖĞRENMESİ

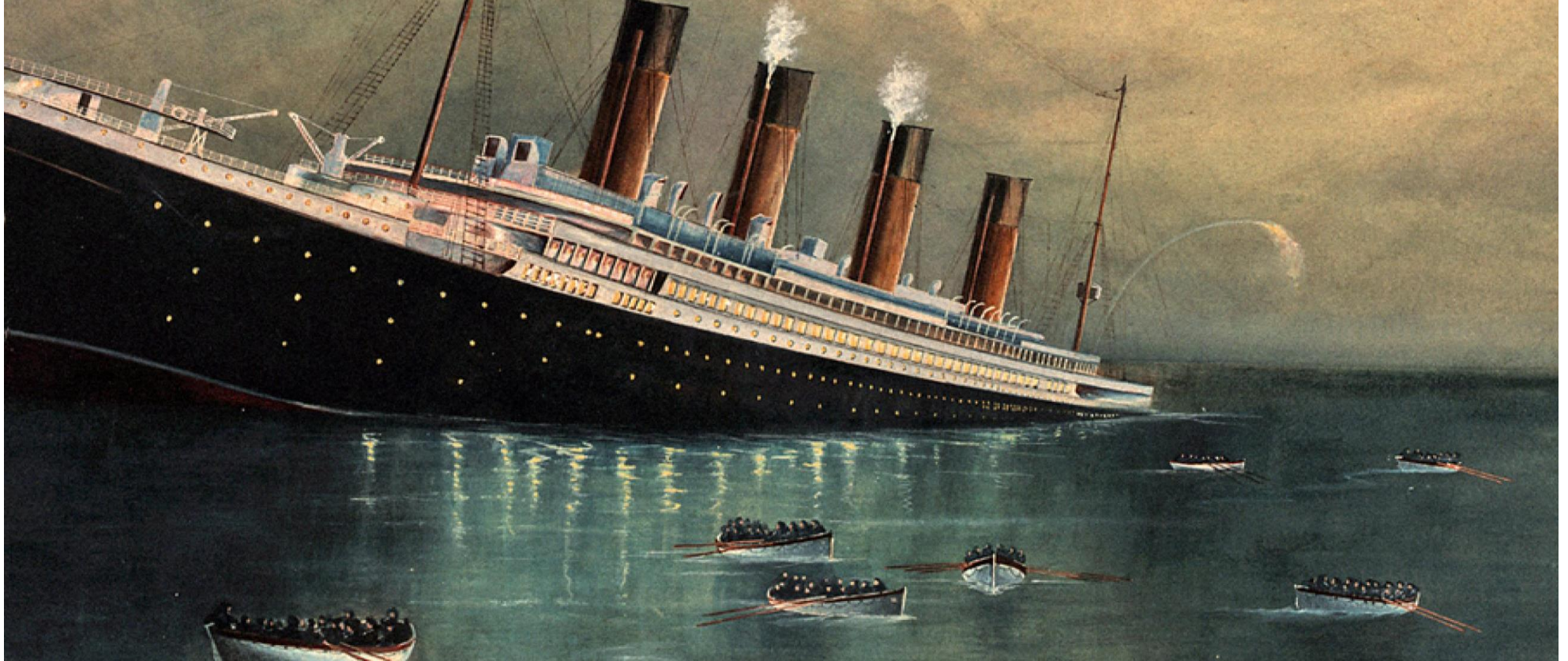
Titanik felaketinden kurtulanların makine öğrenmesi yöntemi ile tahmin edilmesi.

Sinan GÜVEN

22110324

Konu

Titanik felaketinden kurtulanların makine öğrenmesi yöntemi ile tahmin edilmesi.



Araştırma önemi ve ihtiyacı

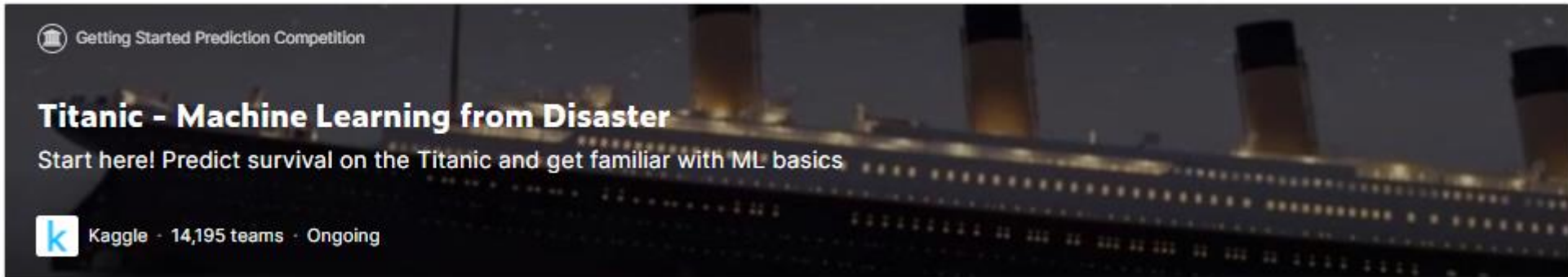
- Titanik felaketinden kurtulanlarla ilgili veri setinin makine öğrenme teknikleri açısından faydalı olmasıdır.
- Veri setinin bir yarışma şeklinde olması makine öğrenmesi konusundaki motivasyonu arttıracaktır.

Araştırma önemi ve ihtiyacı

- Çalışma ile ilgili kodlar ve veri seti aşağıdaki linkte mevcuttur.
[sinanguven87/Master EEM612 MachineLearning \(github.com\)](https://github.com/sinanguven87/Master_EEM612_MachineLearning)

Veri Seti

- Veri Seti olarak [Machine Learning from Disaster](#) linkindeki veri seti kullanılmıştır.
- Kurtulanların tahminine dayalı bir veri setidir.
- Bu veri seti ile ilgili tahminleri içeren bir yarışma vardır. [Sıralama](#)



Veri Seti

- Veri Setine bir yarış girilmiştir. Tahminler puanlanmış ve sıralanmıştır.

14,195	14,392	61,213
Teams	Competitors	Entries

Veri Seti

Veri setinde 3 adet dosya mevcuttur.

- training set (train.csv)
- test set (test.csv)
- gender_submission.csv (Tüm kadınların kurtulduğu ve kurtulanların sadece kadınlar olduğu senaryoda tahminlerin nasıl olması gerektiği ile ilgili örnek dosya)

Veri Seti

Variable	Definition	Key
survival	Survival	0 = No, 1 = Yes
pclass	Ticket class	1 = 1st, 2 = 2nd, 3 = 3rd
sex	Sex	
Age	Age in years	
sibsp	# of siblings / spouses aboard the Titanic	
parch	# of parents / children aboard the Titanic	
ticket	Ticket number	
fare	Passenger fare	
cabin	Cabin number	
embarked	Port of Embarkation	C = Cherbourg, Q = Queenstown, S = Southampton

Veri Seti

- **pclass:** Sosyo-ekonomik statü

1st = Üst







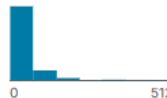
2nd = Orta

3rd = Alt

sibsp: Aile ilişkilerini tanımlar.(Sibling = brother, sister, stepbrother, stepsister Spouse = husband, wife)

parch: Aile ilişkilerini tanımlar.(Parent = mother, father
Child = daughter, son, stepdaughter, stepson. Some children travelled only with a nanny, therefore parch=0 for them.)

Veri Seti

train.csv (61.19 kB)											📄 ⚙️ ➡️
Detail Compact Column											10 of 12 columns ▾
About this file											
contains data											
🔍 PassengerId	# Survived	# Pclass	▲ Name	▲ Sex	# Age	# SibSp	# Parch	▲ Ticket	# Fare		
 1 891	 0 1	 1 3	891 unique values	male female 65% 35%	 0.42 80	 0 8	 0 6	681 unique values	 0 512		
1	0	3	Braund, Mr. Owen Harris	male	22	1	0	A/5 21171	7.25		
2	1	1	Cumings, Mrs. John Bradley (Florence Briggs Thayer)	female	38	1	0	PC 17599	71.2833		
3	1	3	Heikkinen, Miss. Laina	female	26	0	0	STON/O2. 3101282	7.925		
4	1	1	Futrelle, Mrs. Jacques Heath (Lily May Peel)	female	35	1	0	113803	53.1		
5	0	3	Allen, Mr. William Henry	male	35	0	0	373450	8.05		
6	0	3	Moran, Mr. James	male		0	0	330877	8.4583		
7	0	1	McCarthy, Mr. Timothy J	male	54	0	0	17463	51.8625		
8	0	3	Palsson, Master. Gosta Leonard	male	2	3	1	349909	21.075		

Veri Seti

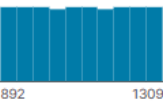

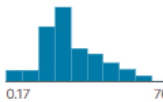


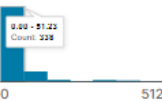
test.csv (28.63 kB)

DetailCompactColumn

10 of 11 columns

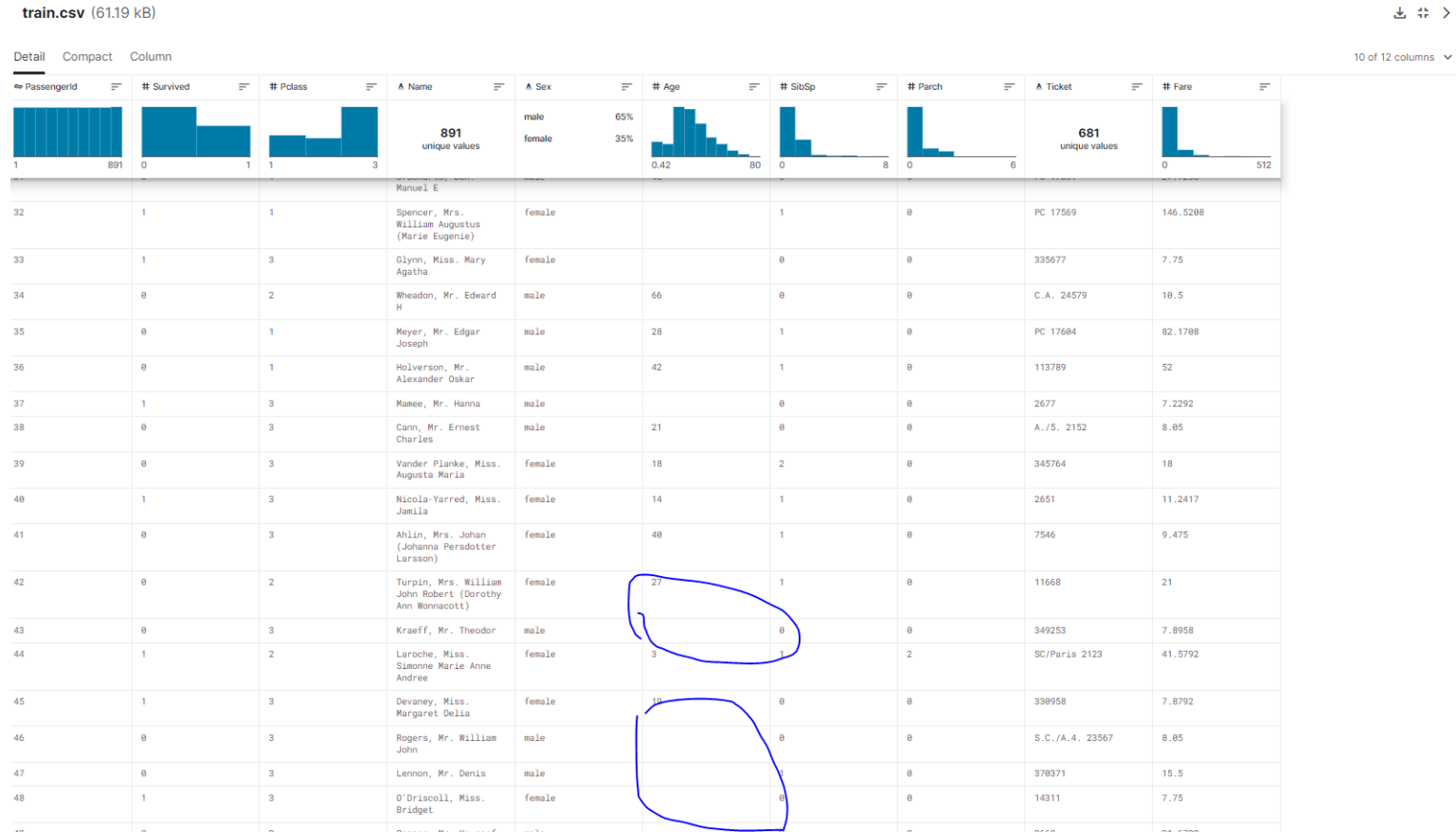
About this file

test data to check the accuracy of the model created

PassengerId	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin
		<div>418 unique values</div>	<div>male64% female36%</div>				<div>PC 176081% 1135031% Other (409)98%</div>		<div>[null]78% B57 B59 B63 B661% Other (88)21%</div>
892	3	Kelly, Mr. James	male	34.5	0	0	330911	7.8292	
893	3	Wilkes, Mrs. James (Ellen Needs)	female	47	1	0	363272	7	
894	2	Myles, Mr. Thomas Francis	male	62	0	0	240276	9.6875	
895	3	Wirz, Mr. Albert	male	27	0	0	315154	8.6625	
896	3	Hirvonen, Mrs. Alexander (Helga E Lindqvist)	female	22	1	1	3101298	12.2875	
897	3	Svensson, Mr. Johan Cervin	male	14	0	0	7538	9.225	
898	3	Connolly, Miss. Kate	female	30	0	0	330972	7.6292	
899	2	Caldwell, Mr. Albert Francis	male	26	1	1	248738	29	
900	3	Abraham, Mrs. Joseph (Sophie Halaut Easu)	female	18	0	0	2657	7.2292	
901	3	Davies, Mr. John Samuel	male	21	2	0	A/4 48871	24.15	
902	3	Tillett, Mr. Ylin	male		0	0	340770	7.8958	

Veri Seti

- Veri setinde bazı değerlerin olmadığı görülmüştür. Bu değerler ileriki adımlarda doldurulacaktır.



Çalışma

Aşağıdaki adımlar sırası ile uygulanmıştır:

- Verilerin Matlab'a aktarımı
- Verilerin yönetimi
- Lineer Regresyon
- Seçim Ağacı
- KNN komşuluk
- Classification Learner ile sınıflandırma
- Test Verilerinin Matlab'a Aktarımı ve İşlenmesi
- Test Çıktılarının Oluşturulması
- Test Çıktılarının Derecelendirilmesi
- Eğitim ve Test Sonuçlarının karşılaştırılması

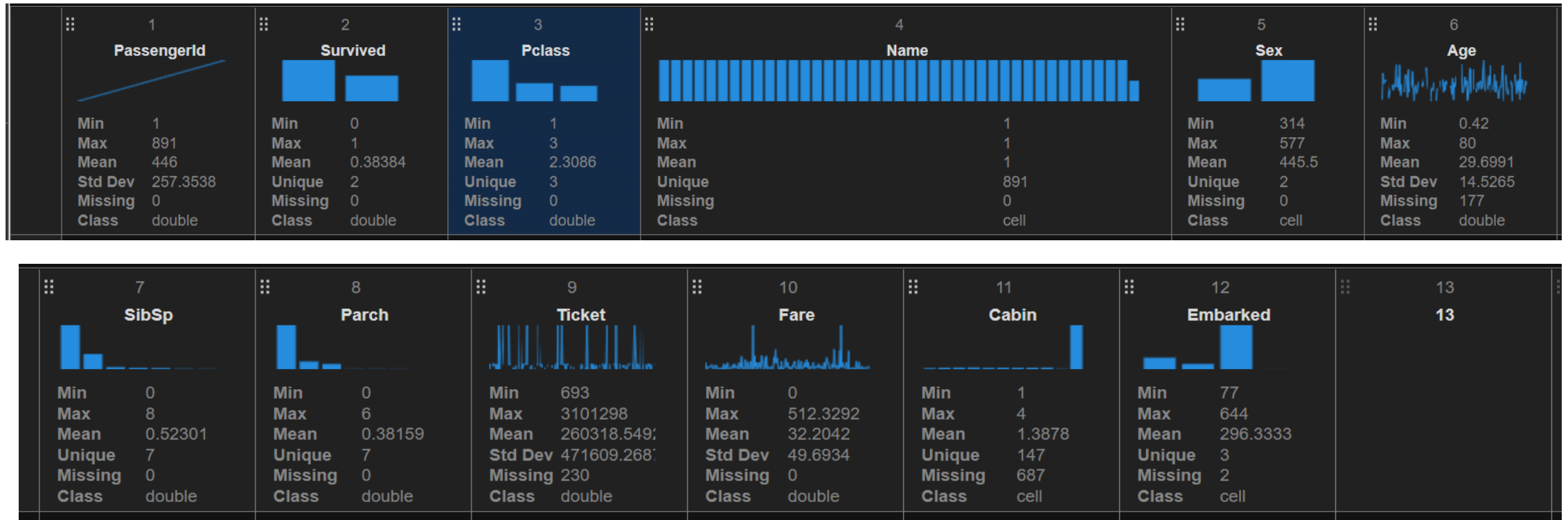
Verilerin MATLAB'a aktarımı

Veriler aşağıdaki kod bloğu ile MATLAB ortamına aktarılmıştır:

```
clc;  
clear;  
close all;  
% Eğitim ve test verisi oku  
Train = readtable('train.csv', 'Format', '%f%f%f%q%C%f%f%f%q%f%q%C');  
Test = readtable('test.csv', 'Format', '%f%f%q%C%f%f%f%q%f%q%C');
```

Verilerin MATLAB'a aktarımı

- Aktarım sonrası verinin analitik gösterimi



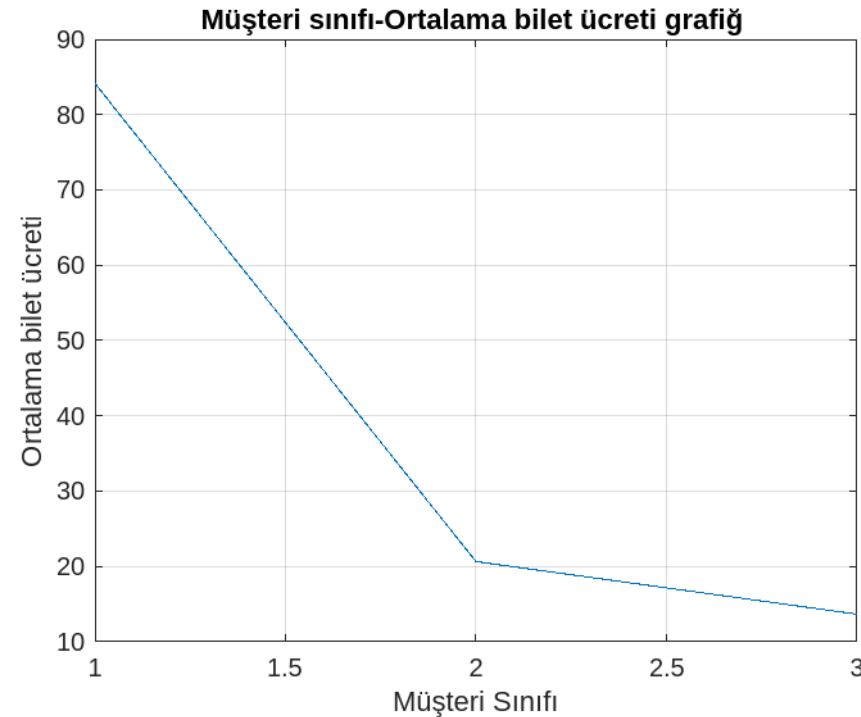
Verilerin Yönetimi

- Test matrisinden yolcu listesi çekilmiştir.
- Olmayan **age** değerleri ortalama değer ile değiştirilmiştir.

```
%Test matrisinden yolcu listesinin alınması
PassangerIdList=Test.PassengerId;
%Olmayan degerleri ortalama deger ile degistir.
avgAge = mean(Train.Age,'omitnan');
Train.Age(isnan(Train.Age)) = avgAge;
Test.Age(isnan(Test.Age)) = avgAge;
```


Verilerin Yönetimi

- Pclass değerlerine göre **fare** değerlerinin grafiği ortalama değerleri çizdirilmiştir. Çizdirilen grafiğe göre ortalama **fare** değeri ile Pclass arasında bağlantı görülmüştür.



Verilerin Yönetimi

- Pclass değerlerinden yola çıkarak kayıp olan **fare** değerlerinin tahmini yapılmıştır. Burada fare değerleri tahmin edilirken ortalama sınıf değerlerine göre **fare** ataması yapılmıştır.

```
%Pclass değerlerinden yola çıkarak kayıp olan fare değerlerinin tahmini
%Burada fare değerleri tahmin edilirken ortalama sınıf değerlerine göre
%fare ataması yapılmıştır.
fare = grpstats(Train(:,{'Pclass','Fare'}),'Pclass'); % sınıf ortalama değerini hesapla
figure;
plot(fare.Pclass,fare.mean_Fare);
title('Müşteri sınıfı-Ortalama bilet ücreti grafiği');
xlabel('Müşteri Sınıfı')
ylabel('Ortalama bilet ücreti')
grid on
for i = 1:height(fare)
    Train.Fare(Train.Pclass == i & isnan(Train.Fare)) = fare.mean_Fare(i);
    Test.Fare(Test.Pclass == i & isnan(Test.Fare)) = fare.mean_Fare(i);
end
```

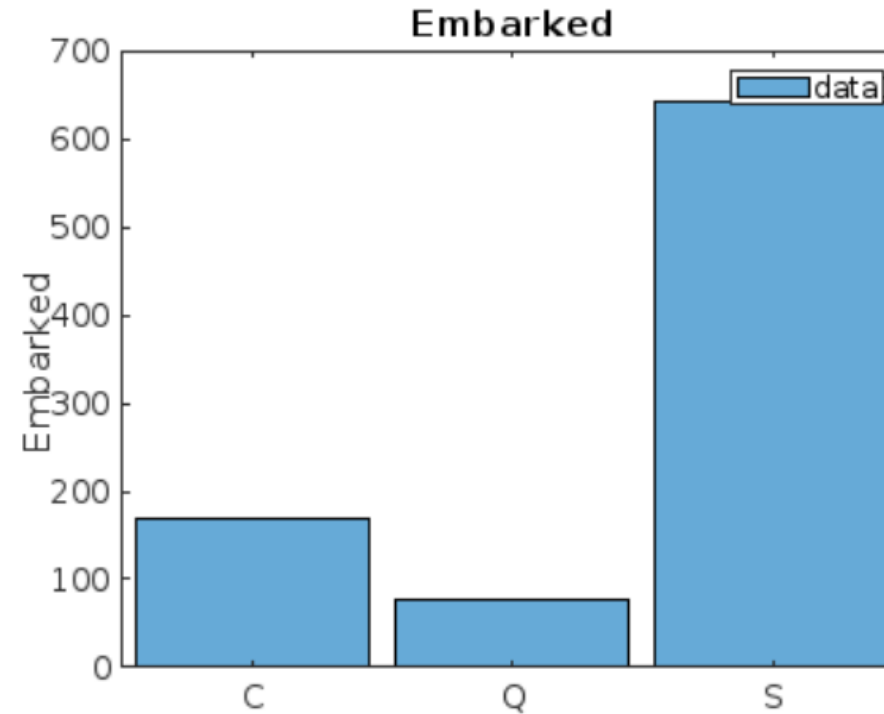
Verilerin Yönetimi

- Olmayan embarked değerleri daha sık olan S ile değiştirilmiştir.

```
% Olmayan embarked değerlerini daha sık olan S ile değiştir.  
histogram(Train.Embarked);  
ylabel("Embarked");  
title("Embarked");  
legend("show");  
% En fazla Embarked değerini bul  
freqVal = mode(Train.Embarked);  
  
% Embarked değeri boş kısımlara en sık değeri uygula  
Train.Embarked(isundefined(Train.Embarked)) = freqVal;  
Test.Embarked(isundefined(Test.Embarked)) = freqVal;  
  
% Kategorik değerden sayısal değere çevir  
Train.Embarked = double(Train.Embarked);  
Test.Embarked = double(Test.Embarked);
```

Verilerin Yönetimi

- Histogramdan da görüleceği üzere Embarked değerlerinin yoğun olarak bulunduğu değer S değeridir. Verilerin S ile değiştirilmesinin sebebi budur.



Verilerin Yönetimi

- Cinsiyet değerleri sayısal değere çevrilmiştir.

```
%Cinsiyet değerlerini sayısal değere çevir  
Train.Sex = double(Train.Sex);  
Test.Sex = double(Test.Sex);
```

Verilerin Yönetimi

- Yaş ve fare değerleri gruplandırılmıştır.

```
% Yaş değerlerini gruplandır
Train.Age = double(discretize(Train.Age, [0:10:20 65 80], ...
    'categorical',{'cocuk','genc','ortayas','yasli'}));
Test.Age = double(discretize(Test.Age, [0:10:20 65 80], ...
    'categorical',{'cocuk','genc','ortayas','yasli'}));

% Fare değerlerini gruplandır.
Train.Fare = double(discretize(Train.Fare, [0:10:30, 100, 520], ...
    'categorical',{'<10','10-20','20-30','30-100','>100'}));
Test.Fare = double(discretize(Test.Fare, [0:10:30, 100, 520], ...
    'categorical',{'<10','10-20','20-30','30-100','>100'}));
```

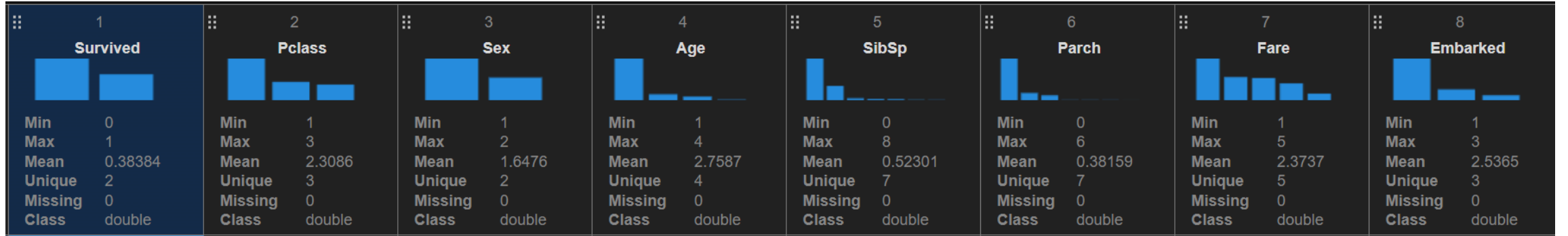
Verilerin Yönetimi

- Name, PassengerId, Ticket ve Cabin öznitelikleri kullanılamayacak durumdadır. Bu yüzden bu değişkenler tablodan atılmıştır.

```
%Name, PassengerId,Ticket ve Cabin öznitelikleri kullanılamayacak  
%durumdadır.  
Train(:,{'Name','PassengerId','Ticket','Cabin'}) = [];  
Test(:,{'Name','PassengerId','Ticket','Cabin'}) = [];
```

Verilerin Yönetimi

- Yapılan işlemler sonucu öznitelikler ve dağılımlarını gösteren çizim aşağıdaki gibidir. Şekilden de görüleceği üzere kayıp bir değer yoktur.



Lineer Regresyon

fitglm fonksiyonu kullanarak lineer regresyon bulunur.

Doğruluk =0.8103

```
% Mantıksal Regression işlemi
tbl=Train(:,2:8);
tbl.Survived=Train.Survived;
reg_model = fitglm(tbl, 'Distribution','binomial');
ypred = predict(reg_model,tbl(:,1:end-1));
ypred = round(ypred); %Olasılıklar 0-1' yuvarlanır.
Confusion_Matrix = confusionmat(tbl.Survived,ypred);
AccuracyReg = trace(Confusion_Matrix)/sum(Confusion_Matrix, 'all');
```

Karar Ağacı

Fitctree fonksiyonu kullanarak karar ağacı oluşturulur. Yapılan işlemde çapraz doğrulama ve en iyi seviyede kırpma işlemi gerçekleştirilmiştir. Bu işlemler sonunda doğruluk değeri(0.8586) bulunmuştur.

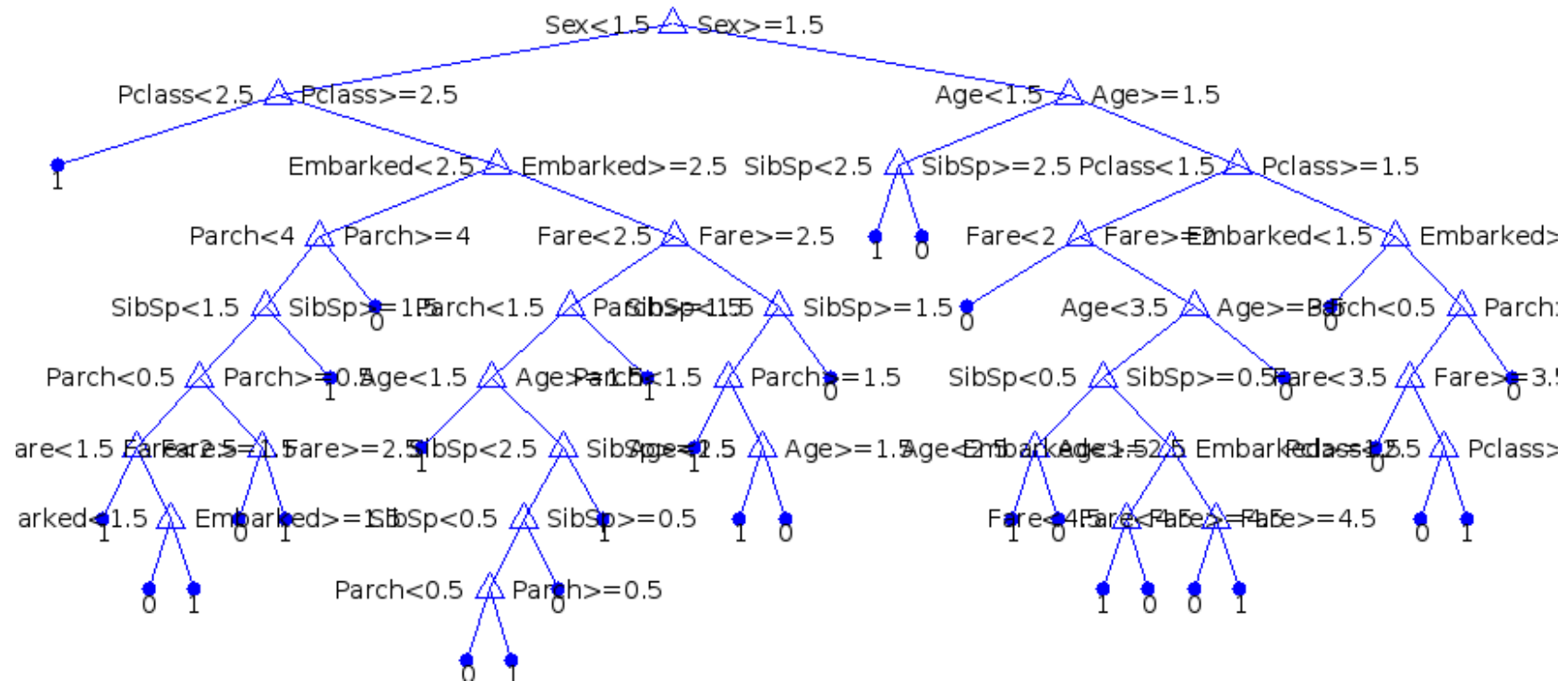
```
% Karar ağacı oluşturulur ve çizdirilir.
mytree = fitctree(Train(:,2:end),Train(:,1));
view(mytree, 'Mode', 'graph')

%Karar ağacında çapraz validasyon hataları gidermek için en uygun budama
%seviyesi belirlenir.Aynı zamanda gereksiz seviyeler budanır.
[~,~,~,BestLevel] = cvloss(mytree,'subtrees','all','treesize','min');
prunetree = prune(mytree,'Level',BestLevel);
view(prunetree,'mode','graph')

% Karar ağacı doğruluk değeri bulunur.
label = predict(prunetree,Train(:,2:end));
Confusion_Matrix_Tree = confusionmat(Train.Survived,label);
Accuracy_Tree = trace(Confusion_Matrix_Tree)/sum(Confusion_Matrix_Tree, 'all');
```

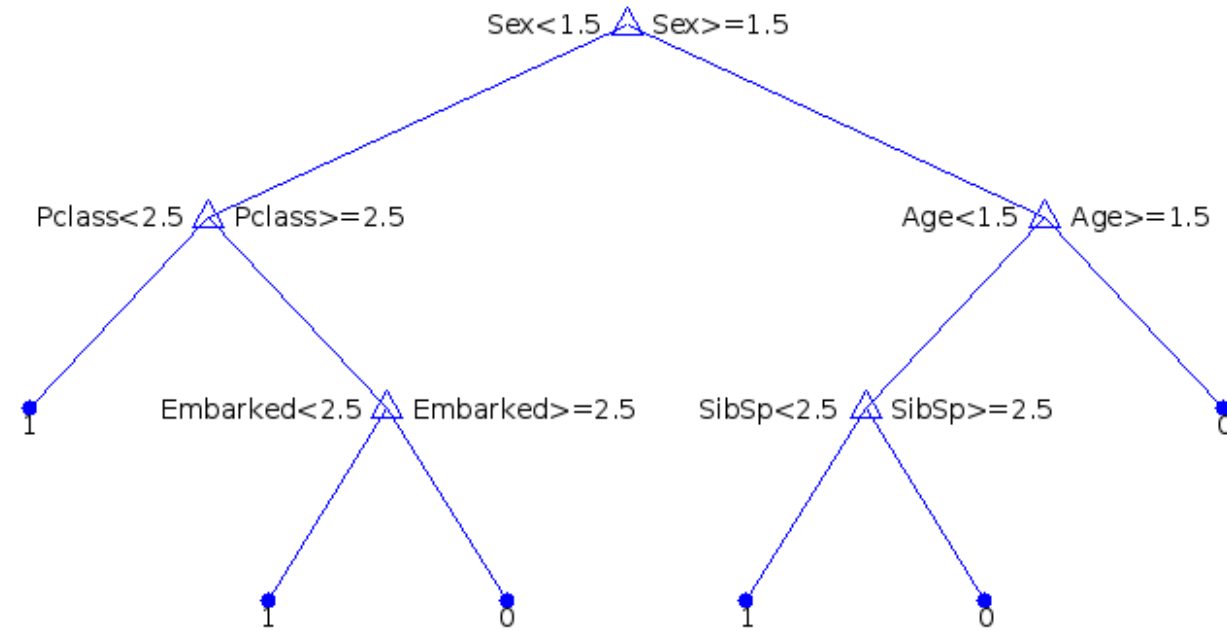
Karar Ağacı

Karar ağacının kırpılmamış hali aşağıdaki gibidir:



Karar Ağacı

Karar ağacının kırpılmış hali aşağıdaki gibidir:



KNN komşuluk değeri

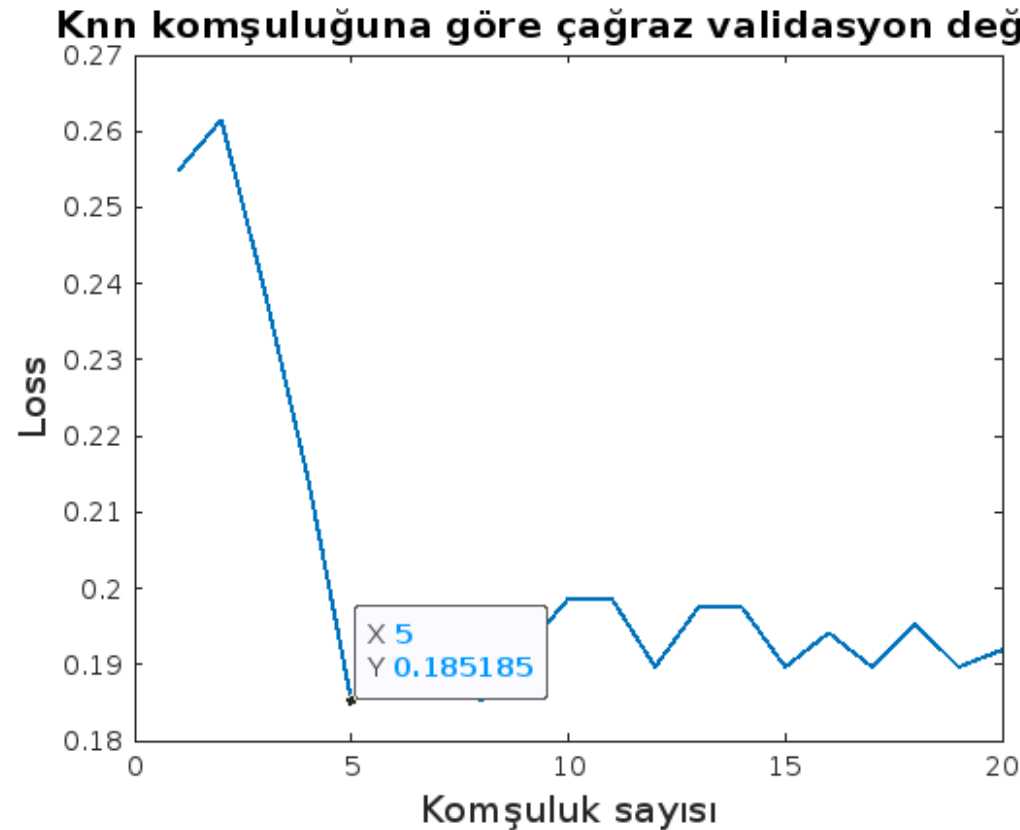
- Optimum komşuluk sayısını bulabilmek için komşuluk değerleri 1:20 arasında değiştirilmiştir. N değerine göre Knn çapraz validasyon kaybı değerleri çizdirilmiştir.

```
%%KNN komşuluk
%Optimum komşuluk sayısını bulabilmek için komşuluk değerlerini 1:20 arasında değiştir.
cv_loss_knn=zeros(20,1);
Neighbors= transpose(1:20);
for num_neighbours = 1:20
    rng(1);
    knn_model = ClassificationKNN.fit(Train(:,2:8),Train(:,1), 'NumNeighbors',num_neighbours);
    cvc_model = crossval(knn_model);
    cv_loss = kfoldLoss(cvc_model);
    cv_loss_knn(num_neighbours)=cv_loss;
end

% Knn değerlerini çizdir.
figure('Name', 'KNN')
plot(Neighbors, cv_loss_knn, 'LineWidth',1.5)
title('Knn komşuluğuna göre çapraz validasyon değeri', 'FontSize',14);
xlabel('Komşuluk sayısı', 'FontSize',14);
ylabel('Loss', 'FontSize',14);
```

KNN komşuluk değeri

- Elde edilen grafikte en az kayıplı n değerinin 5 olduğu görülmüştür.



KNN komşuluk değeri

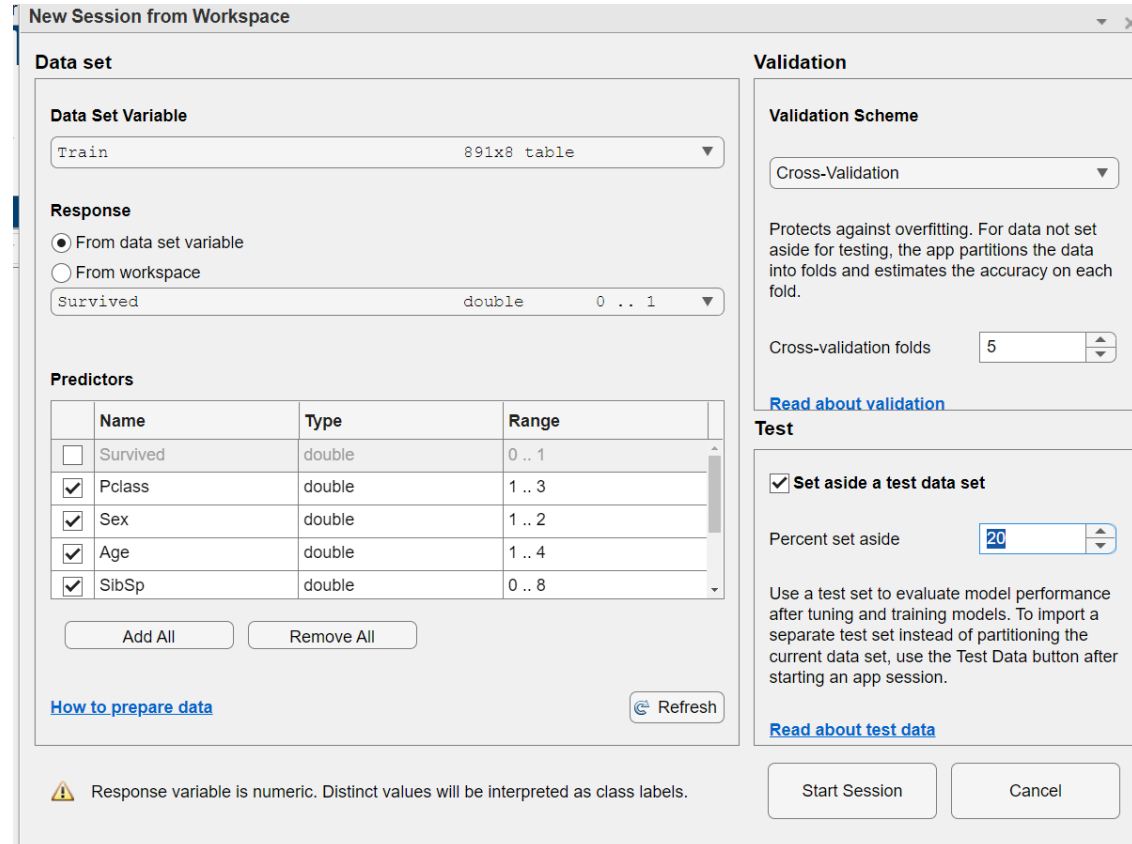
- Optimum n değerine göre modeli elde edilmiş ve doğruluk değeri (0.8462) bulunmuştur.

```
% Optimal komşuluk sayısı için Knn modelini elde et
[M,minimumIndex] = min(cv_loss_knn);
knn_model = ClassificationKNN.fit(Train(:,2:end),Train(:,1), 'NumNeighbors',num_neighbours);

% Doğruluk değerini bul.
label_knn = predict(knn_model,Train(:,2:end));
Confusion_Matrix_knn = confusionmat(Train.Survived,label_knn);
Accuracy_knn = trace(Confusion_Matrix_knn)/sum(Confusion_Matrix_knn, 'all');
```

Classification Learner ile Sınıflandırma

- Sınıflandırma için veri seti seçilmiştir. Verilerin %20'si test için kullanılmıştır.



Data set

Data Set Variable
Train 891x8 table

Response
☒ From data set variable
☐ From workspace
Survived double 0 .. 1

Predictors

	Name	Type	Range
<input type="checkbox"/>	Survived	double	0 .. 1
<input checked="" type="checkbox"/>	Pclass	double	1 .. 3
<input checked="" type="checkbox"/>	Sex	double	1 .. 2
<input checked="" type="checkbox"/>	Age	double	1 .. 4
<input checked="" type="checkbox"/>	SibSp	double	0 .. 8

[How to prepare data](#) Refresh

Validation

Validation Scheme
Cross-Validation

Protects against overfitting. For data not set aside for testing, the app partitions the data into folds and estimates the accuracy on each fold.

Cross-validation folds 5

[Read about validation](#)

Test


☒ Set aside a test data set

Percent set aside 20

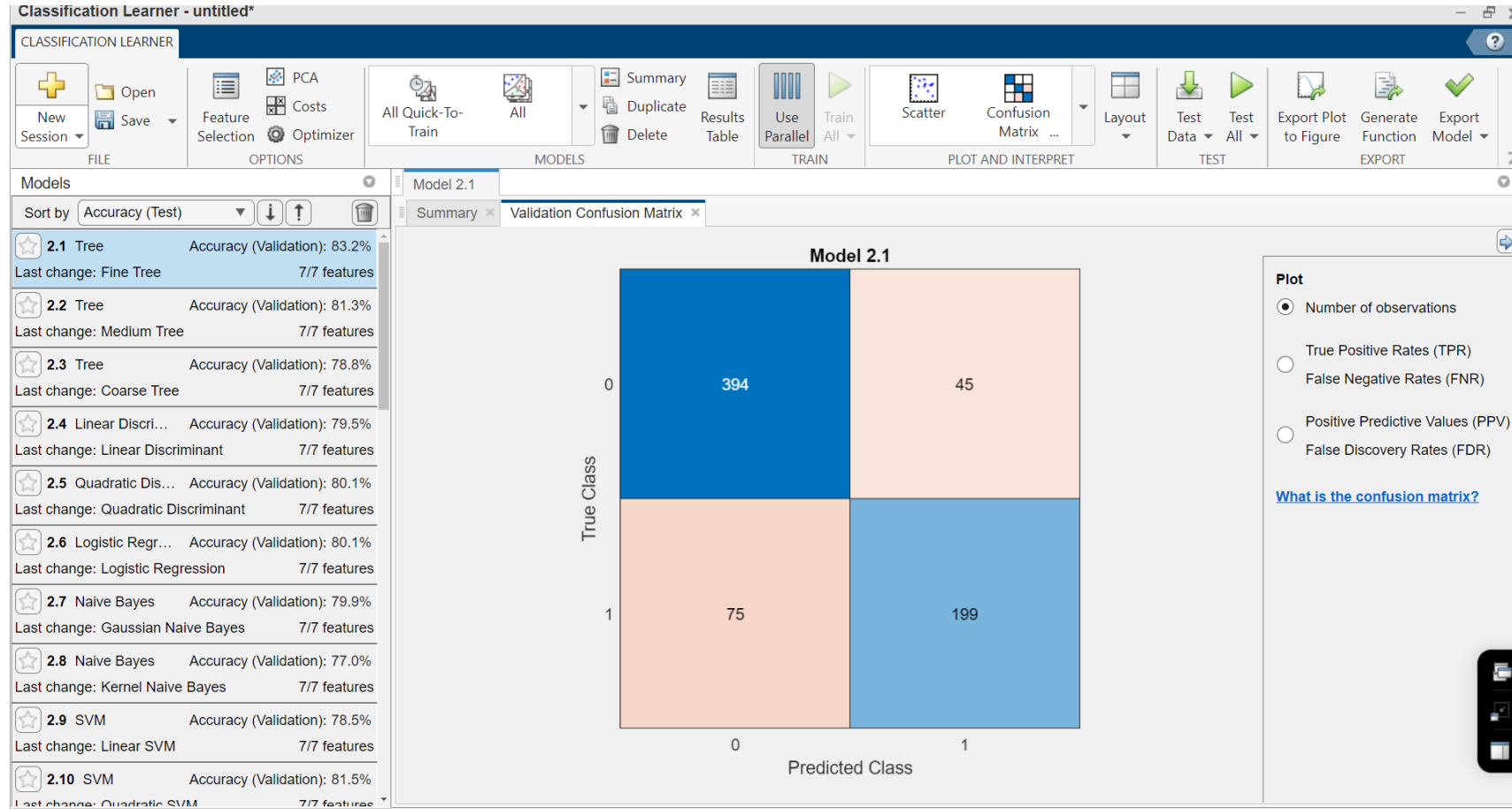
Use a test set to evaluate model performance after tuning and training models. To import a separate test set instead of partitioning the current data set, use the Test Data button after starting an app session.

[Read about test data](#)

[Start Session](#) [Cancel](#)

 Response variable is numeric. Distinct values will be interpreted as class labels.

Classification Learner

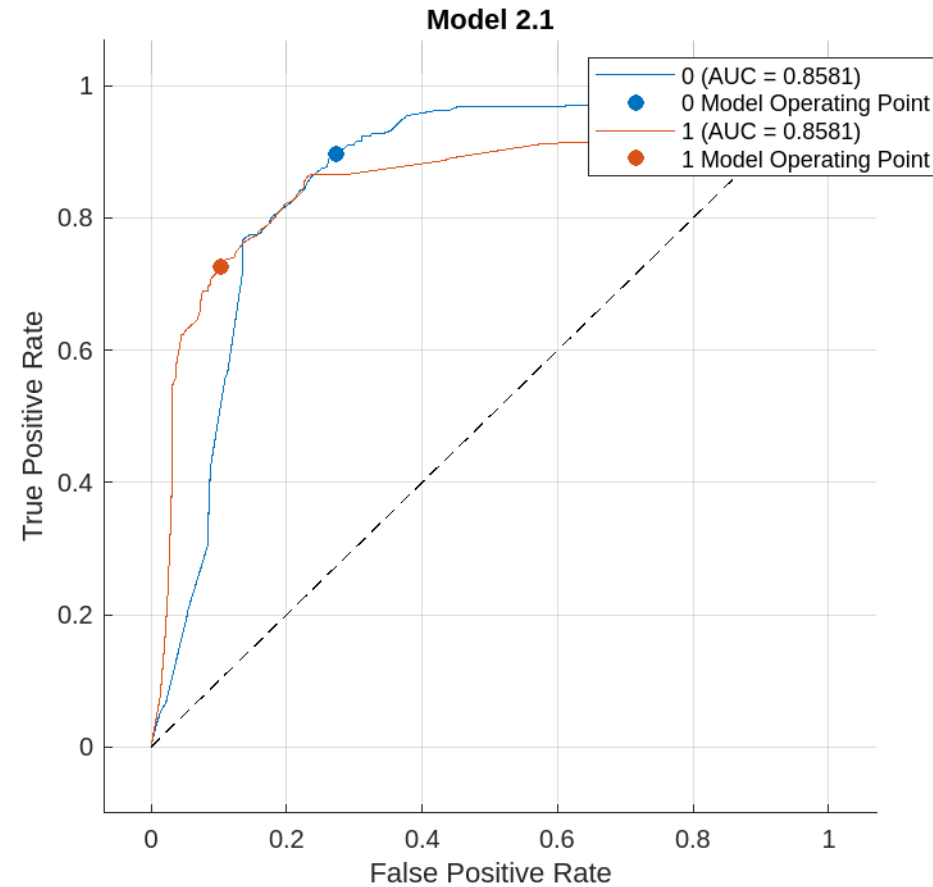


Classification Learner

Results Table							
Session: untitled							
Training Data: Train Observations: 802 Predictors: 7 Response Name: Survived Response Classes: 2							
Validation: 5-fold cross-validation							
Test Data: Train Observations: 89							
vorite	Model Number	Model Type	Status	Accuracy (Validation) ↓	Total Cost (Validation)	Accuracy (Test)	Total Cost (Test)
<input type="checkbox"/>	2.13	SVM	✔ Tested	81.92 %	145	84.27 %	14
<input type="checkbox"/>	2.19	KNN	✔ Tested	81.67 %	147	84.27 %	14
<input type="checkbox"/>	2.16	KNN	✔ Tested	81.67 %	147	84.27 %	14
<input type="checkbox"/>	2.20	KNN	✔ Tested	81.30 %	150	83.15 %	15
<input type="checkbox"/>	2.5	Discriminant	✔ Tested	81.05 %	152	82.02 %	16
<input type="checkbox"/>	2.21	Ensemble	✔ Tested	80.92 %	153	85.39 %	13
<input type="checkbox"/>	2.10	SVM	✔ Tested	80.80 %	154	83.15 %	15
<input type="checkbox"/>	2.22	Ensemble	✔ Tested	80.67 %	155	83.15 %	15
<input type="checkbox"/>	2.18	KNN	✔ Tested	80.55 %	156	83.15 %	15
<input type="checkbox"/>	2.25	Ensemble	✔ Tested	80.55 %	156	80.90 %	
<input type="checkbox"/>	2.3	Tree	✔ Tested	79.93 %	161	80.90 %	
<input type="checkbox"/>	2.2	Tree	✔ Tested	79.80 %	162	83.15 %	
<input type="checkbox"/>	2.17	KNN	✔ Tested	79.68 %	163	80.90 %	
<input type="checkbox"/>	2.1	Tree	✔ Tested	79.68 %	163	80.90 %	17

Classification Learner

- ROC eğrisi Validasyon (Fine Tree)



Test Çıktılarının Oluşturulması

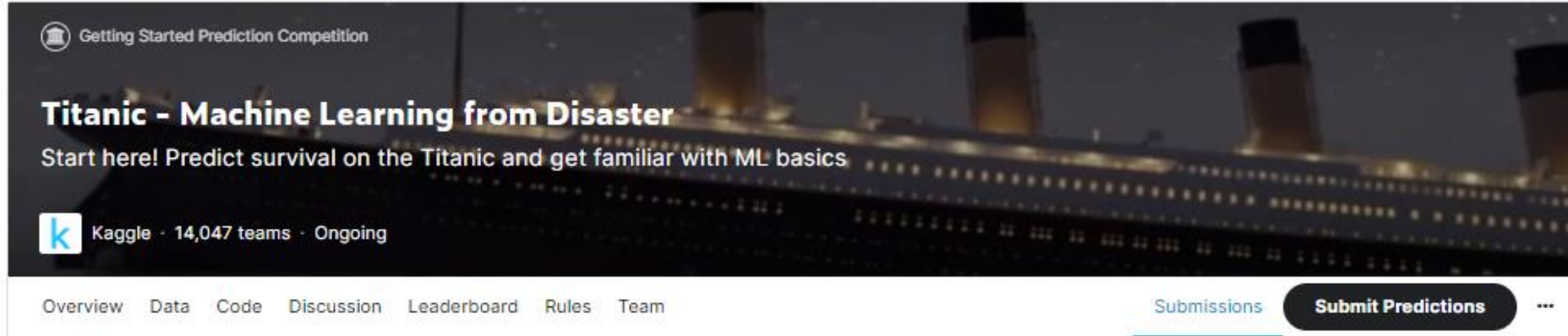
- Test çıktıları csv dosyalarına kaydedilmiştir.

```
%Regresyon modeli çıktılarını csv olarak yazdır
reg_modelTest = predict(reg_model,Test(:,1:end));
generate_csv(PassangerIdList, reg_modelTest,'predictionsReg.csv');

%Tree modeli çıktılarını csv olarak yazdır
prunetreeTest = predict(prunetree,Test(:,1:end));
generate_csv(PassangerIdList, prunetreeTest,'predictionsTree.csv');

%Knn modeli çıktılarını csv olarak yazdır
knn_modelTest = predict(knn_model,Test(:,1:end));
generate_csv(PassangerIdList, knn_modelTest,'predictionsKnn.csv');
```


Test Çıktılarının Derecelendirilmesi



Getting Started Prediction Competition

Titanic - Machine Learning from Disaster

Start here! Predict survival on the Titanic and get familiar with ML basics

 Kaggle · 14,047 teams · Ongoing

[Overview](#) [Data](#) [Code](#) [Discussion](#) [Leaderboard](#) [Rules](#) [Team](#) [Submissions](#) [Submit Predictions](#) [...](#)

Submissions

All		Successful	Errors	Recent ▾	
Submission and Description				Public Score ⓘ	
✓	Tree.csv	Complete · now			0.78468
✓	Regression.csv	Complete · 1s ago			0.76794
✓	Knn.csv	Complete · 6m ago			0.76076

Eğitim ve Test Sonuçlarının karşılaştırılması

Yöntem	Train	Kaggle Yüklenmesi Sonucu
Regresyon	0.8103	0.76794
Tree	0.8586	0.78468
Knn	0.8462	0.76076

Sonuç

- Sonuç olarak elde edilen verilerle test veriler arasında tutarlılık görülmüştür.
- Classification Learner ile çalışırken optimizasyon yapmak gerekmektedir bazı durumlarda elle optimizasyon yapmak gerekebilmektedir.
- Titanik kazası ile ilgili yarışma gerçekten zorlayıcı bir yarışmadır.

Kaynaklar

- [Kaggle Dataset](#)
- [Classification Learner](#)
- [EEM 612 ÖRÜNTÜ TANIMAVE MAKİNE ÖĞRENMESİ ders notları](#)
- [Knn](#)
- [Karar Ağacı](#)
- [Lineer Regresyon](#)
- [sinanguven87/Master EEM612 MachineLearning \(github.com\)](#)