

MACHINE LEARNING PROJECT REPORT

The source of the data: Previously I choose dataset ([satinalma.csv](#)). This dataset includes "Kullanıcılarda", "Cinsiyet", "Yaş", "TahminiMaas", "SatinAldimi". Dataset from a survey conducted by an automobile company for its customers.

The type of the data: Dataset consists of categorical data.

The size of the data: Size of the dataset: 400 rows x 5 columns

The format of the data: 4 inputs and 1 output.

Inputs represent "Kisild", "Cinsiyet", "Yaş", "TahminiMaas" but I use "Yaş" and "TahminiMaas"


Output represents "SatinAldimi"

The type of the problem: Classifying

I didn't use all of the data in dataset because I wanted to question whether he bought the car by salary and age. So I use DecisionTree Algorithm from classification because It is easy to understand and interpret. Can be used to process both numerical and class data. This dataset shows two results by comparing two values. Yes(1) or No(0). I used this algorithm to find out which results are based on which variables in general.

 I upload dataset to the spyder. For this section I write this code;

```
veriler=pd.read_csv('satinalma.csv')
```

 Then describe the dependent variable for column's "SatinAldimi" and describe independent variables for column's "Yaş", "TahminiMaas". I did not take any action on this subject since there is no missing data in the data set.

```
X=veriler.iloc[:, [2,3]].values #independent variable  
y=veriler.iloc[:, 4].values     #the dependent variable
```

✚ Later I split these data for testing ,

```
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size = 0.20, random_state = 0)
```

I used test_size=0.20 then I tried different values and observed their effects. I will tell in result section.

✚ Then I made feature scaling because “yas” and “TahminiMaas” are not in same unit,

```
from sklearn.preprocessing import StandardScaler
sc_X = StandardScaler()
X_train = sc_X.fit_transform(X_train)
X_test = sc_X.transform(X_test)
```

✚ For these dataset I applied DecisionTreeClassifier and also use entropy. By selecting the smallest entropy value, it puts this value at the beginning of the tree and then the other branches of the tree are created by calculating the entropy for other data.

```
from sklearn.tree import DecisionTreeClassifier
classifier = DecisionTreeClassifier(criterion = 'entropy', random_state=0)
```

✚ Learn y_train from X_train

```
classifier.fit(X_train, y_train)
```

✚ Guess from X_test

```
y_pred = classifier.predict(X_test)
```

✚ I put the arguments in the tree one by one and found the corresponding dependent variables. I compared these dependent variable values.

```
from sklearn.metrics import confusion_matrix
cm = confusion_matrix(y_test, y_pred)
print(cm)
```

✚ These codes give result

```
[400 rows x 5 columns]
[[53  5]
 [ 3 19]]
```

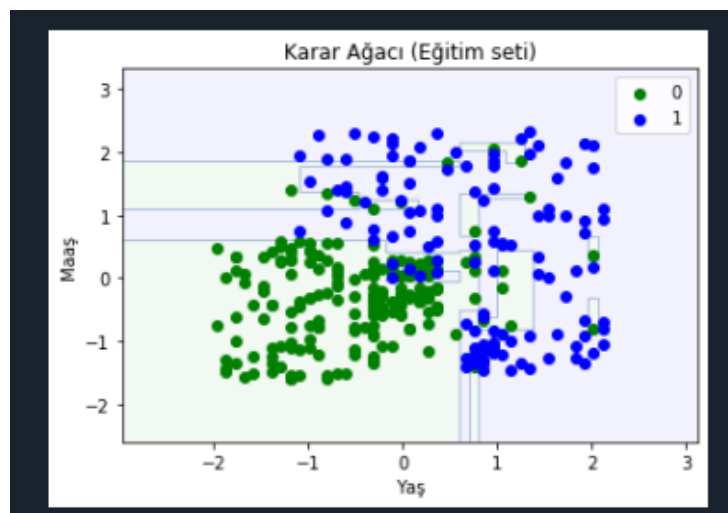
For test_size=0.20 ;

Result can show , [3+19]=21 of the data buy a car (1)

✚ I drew graphs for training , this code;

```
from matplotlib.colors import ListedColormap
X_set, y_set = X_train, y_train
X1, X2 = np.meshgrid(np.arange(start = X_set[:, 0].min() - 1, stop = X_set[:, 0].max() + 1, step = 0.01),
                     np.arange(start = X_set[:, 1].min() - 1, stop = X_set[:, 1].max() + 1, step = 0.01))
plt.contourf(X1, X2, classifier.predict(np.array([X1.ravel(), X2.ravel()]).T).reshape(X1.shape),
             alpha = 0.05, cmap = ListedColormap(('green', 'blue')))
plt.xlim(X1.min(), X1.max())
plt.ylim(X2.min(), X2.max())
for i, j in enumerate(np.unique(y_set)):
    plt.scatter(X_set[y_set == j, 0], X_set[y_set == j, 1],
               c = ListedColormap(('green', 'blue'))(i), label = j)
plt.title('Karar Ağacı (Eğitim seti)')
plt.xlabel('Yaş')
plt.ylabel('Maaş')
plt.legend()
plt.show()
```

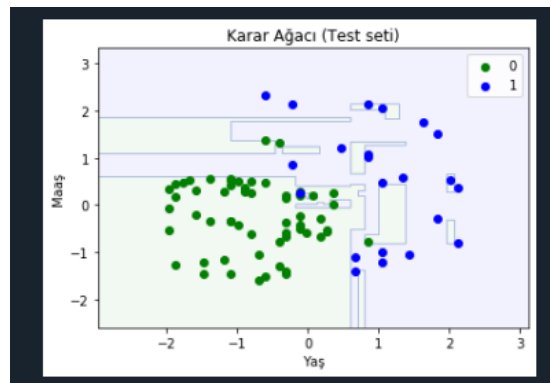
✚ Blue is (1) so costumer buy an otomobile
Green is (0) so costumer doesn't buy an otomobile.



I drew graphs for testing , this code;

```
from matplotlib.colors import ListedColormap
X_set, y_set = X_test, y_test
X1, X2 = np.meshgrid(np.arange(start = X_set[:, 0].min() - 1, stop = X_set[:, 0].max() + 1, step = 0.01),
                     np.arange(start = X_set[:, 1].min() - 1, stop = X_set[:, 1].max() + 1, step = 0.01))
plt.contourf(X1, X2, classifier.predict(np.array([X1.ravel(), X2.ravel()]).T).reshape(X1.shape),
             alpha = 0.05, cmap = ListedColormap(('green', 'blue')))
plt.xlim(X1.min(), X1.max())
plt.ylim(X2.min(), X2.max())
for i, j in enumerate(np.unique(y_set)):
    plt.scatter(X_set[y_set == j, 0], X_set[y_set == j, 1],
                c = ListedColormap(('green', 'blue'))(i), label = j)
plt.title('Karar Ağacı (Test seti)')
plt.xlabel('Yaş')
plt.ylabel('Maaş')
plt.legend()
plt.show()
```

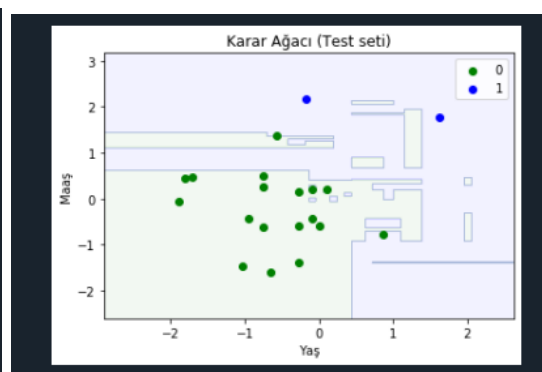
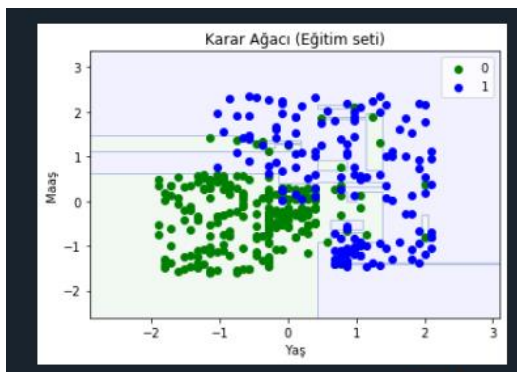
Blue dots indicate that the car has been picked up.



If I change test_size, for example test_size=0.05 result will be

```
[400 rows x 5 columns]
[[15  3]
 [ 0  2]]
```

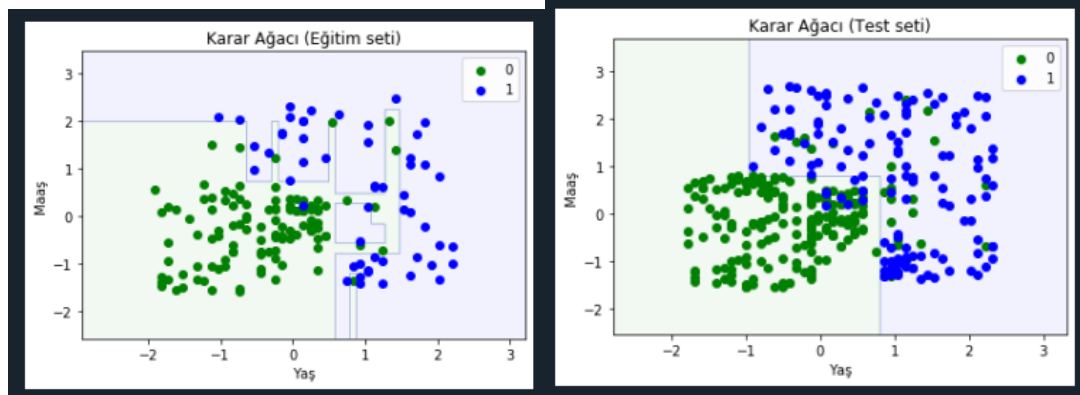
Result can show , $[0+2]=2$ of the data buy a car (1)



✚ If I change test_size, for example test_size=0.46 result will be

```
[400 rows x 5 columns]
[[105  9]
 [ 15 55]]
```

Result can show , $[15+55]=70$ of the data buy a car (1)



✚ Finally I applied decisiontree algorithm from classifying. I found the effect of “yas” and “tahminiMaas” on car buying.

030716014- Merve AŞIKUZUNOĞLU