```
import seaborn as sns

# "mpg" veri setini yükle
df = sns.load_dataset("mpg")
```

Min max gibi degerleri tek tek yazdim. aksi durumda sadece en son verdigin komutu uyguluyor

Elimizde categorical variable'lar da oldugu icin her zaman fonksiyon halini kullanamayiz.

```
#1
df.info , df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 398 entries, 0 to 397
Data columns (total 9 columns):
 #   Column        Non-Null Count  Dtype  
---  ------        --------------  -----  
 0   mpg           398 non-null    float64
 1   cylinders     398 non-null    int64  
 2   displacement  398 non-null    float64
 3   horsepower    392 non-null    float64
 4   weight        398 non-null    int64  
 5   acceleration  398 non-null    float64
 6   model_year    398 non-null    int64  
 7   origin        398 non-null    object 
 8   name          398 non-null    object 
dtypes: float64(4), int64(3), object(2)
memory usage: 28.1+ KB
(<bound method DataFrame.info of       mpg  cylinders  displacement  horsepower  weight  acceleration  \
 0    18.0          8         307.0       130.0    3504          12.0   
 1    15.0          8         350.0       165.0    3693          11.5   
 2    18.0          8         318.0       150.0    3436          11.0   
 3    16.0          8         304.0       150.0    3433          12.0   
 4    17.0          8         302.0       140.0    3449          10.5   
 ..    ...        ...           ...         ...     ...           ...   
 393  27.0          4         140.0        86.0    2790          15.6   
 394  44.0          4          97.0        52.0    2130          24.6   
 395  32.0          4         135.0        84.0    2295          11.6   
 396  28.0          4         120.0        79.0    2625          18.6   
 397  31.0          4         119.0        82.0    2720          19.4   

      model_year  origin                       name  
 0            70     usa  chevrolet chevelle malibu  
 1            70     usa          buick skylark 320  
 2            70     usa         plymouth satellite  
 3            70     usa              amc rebel sst  
 4            70     usa                ford torino  
 ..          ...     ...                        ...  
 393          82     usa           ford mustang gl  
 394          82  europe                  vw pickup  
 395          82     usa              dodge rampage  
 396          82     usa                ford ranger  
 397          82     usa                 chevy s-10  

 [398 rows x 9 columns]>,
 None)
```

```
#2
df.head(10)
```

	mpg	cylinders	displacement	horsepower	weight	acceleration	model_year	origin	name
0	18.0	8	307.0	130.0	3504	12.0	70	usa	chevrolet chevelle malibu

#3.1
df.max()

```
mpg                    46.6
cylinders                 8
displacement            455.0
horsepower              230.0
weight                 5140
acceleration             24.8
model_year               82
origin                  usa
name          vw rabbit custom
dtype: object
```

#3.2
df.min()

```
mpg                       9.0
cylinders                   3
displacement             68.0
horsepower               46.0
weight                   1613
acceleration              8.0
model_year                 70
origin                 europe
name          amc ambassador brougham
dtype: object
```

#3.3
df.var

```
<bound method NDFrame._add_numeric_operations.<locals>.var of      mpg  cylinders  displacement  horsepower  weight  acceleration \
0    18.0          8         307.0       130.0    3504          12.0
1    15.0          8         350.0       165.0    3693          11.5
2    18.0          8         318.0       150.0    3436          11.0
3    16.0          8         304.0       150.0    3433          12.0
4    17.0          8         302.0       140.0    3449          10.5
..    ...        ...           ...         ...     ...           ...
393  27.0          4         140.0        86.0    2790          15.6
394  44.0          4          97.0        52.0    2130          24.6
395  32.0          4         135.0        84.0    2295          11.6
396  28.0          4         120.0        79.0    2625          18.6
397  31.0          4         119.0        82.0    2720          19.4

     model_year  origin                       name
0            70     usa  chevrolet chevelle malibu
1            70     usa          buick skylark 320
2            70     usa         plymouth satellite
3            70     usa              amc rebel sst
4            70     usa                ford torino
..          ...     ...                        ...
393          82     usa           ford mustang gl
394          82  europe                  vw pickup
395          82     usa              dodge rampage
396          82     usa                ford ranger
397          82     usa                 chevy s-10

[398 rows x 9 columns]>
```

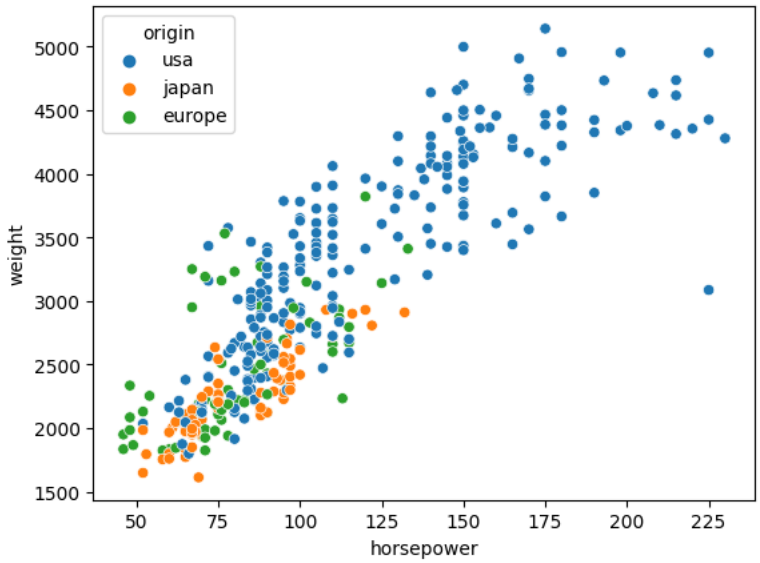
df["horsepower"] ,df.weight

```
(0      130.0
 1      165.0
 2      150.0
 3      150.0
 4      140.0
        ...
 393     86.0
 394     52.0
 395     84.0
 396     79.0
 397     82.0
 Name: horsepower, Length: 398, dtype: float64,
 0     3504
 1     3693
```

```
2     3436
3     3433
4     3449
      ...
393   2790
394   2130
395   2295
396   2625
397   2720
Name: weight, Length: 398, dtype: int64)
```

```
import seaborn as sns
sns.scatterplot(x="horsepower",y="weight",hue="origin",data=df)
```

```
<Axes: xlabel='horsepower', ylabel='weight'>
```




```
df_corr=df["mpg"].corr(df.acceleration)
print(df_corr)
```

```
0.42028891210165065
```

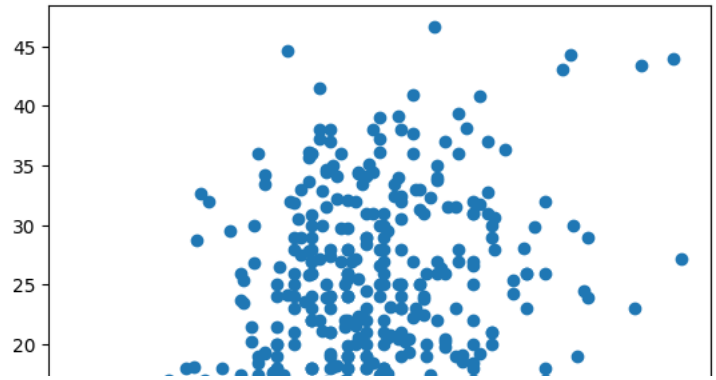
```
#17
df.isnull().sum()
```

```
mpg             0
cylinders       0
displacement    0
horsepower      6
weight          0
acceleration    0
model_year      0
origin          0
name            0
dtype: int64
```

```
#17
import pandas as pd
X_horse=df.iloc[:,3:4]
X_weight=df.iloc[:,4:5]
X_acc=df.iloc[:, 5:6]
y=df.iloc[:, :1]
```

```
#18
import matplotlib.pyplot as plt
plt.scatter(X_acc,y)
```

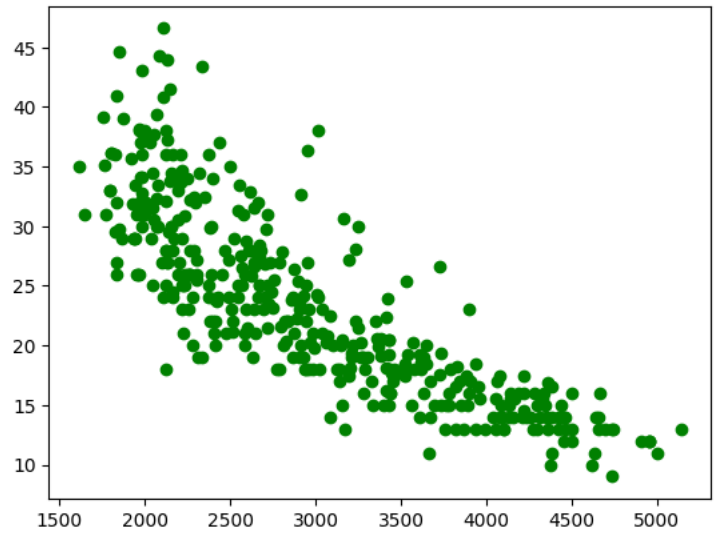
```
<matplotlib.collections.PathCollection at 0x7b04b990cb20>
```



#18

```
import matplotlib.pyplot as plt
plt.scatter(X_weight,y,c="g")
```

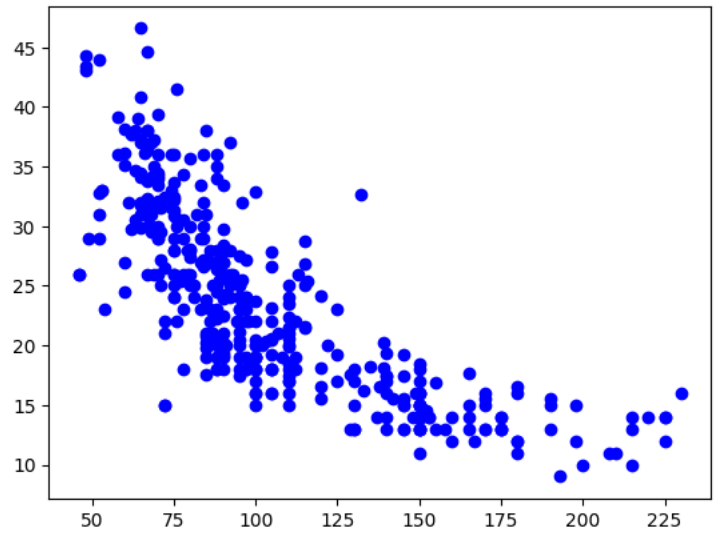
```
<matplotlib.collections.PathCollection at 0x7b04b9826650>
```



#18

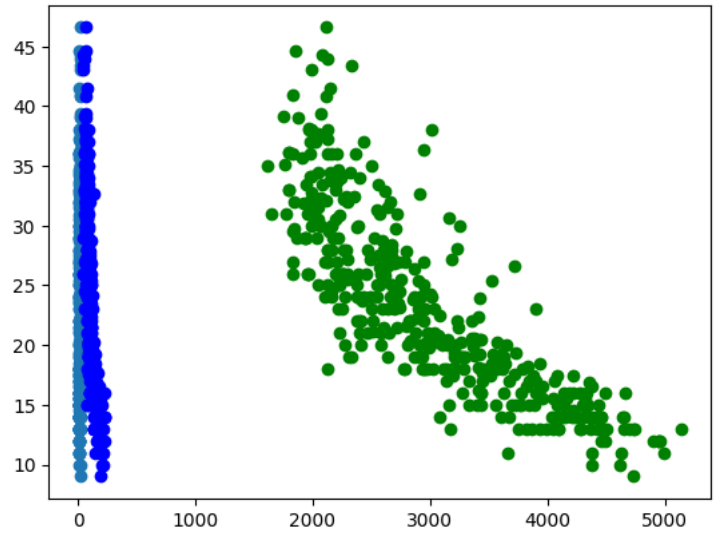
```
import matplotlib.pyplot as plt
plt.scatter(X_horse,y,c="b")
```

```
<matplotlib.collections.PathCollection at 0x7b04b98a90f0>
```



```
#18
import matplotlib.pyplot as plt
plt.scatter(X_acc,y)
plt.scatter(X_horse,y,c="b")
plt.scatter(X_weight,y,c="g")
```

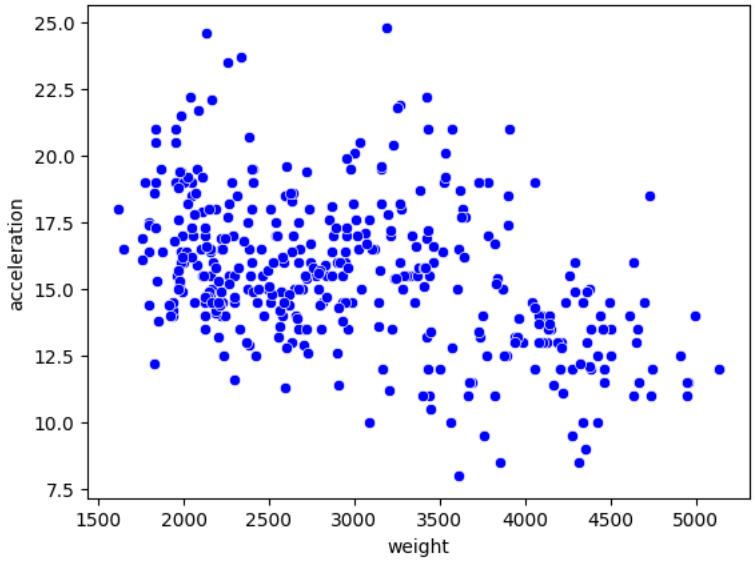
```
<matplotlib.collections.PathCollection at 0x7b04b973d450>
```




```
#18
from sklearn.linear_model import LinearRegression
from sklearn.metrics import r2_score,mean_absolute_error,mean_squared_error
from sklearn.preprocessing import StandardScaler
from sklearn.metrics.pairwise import cosine_similarity
mpg=df.mpg
```

```
#18
import pandas as pd
import numpy as np
import seaborn as sns
korelasyon=np.corrcoef(df.weight,df.acceleration)
sns.scatterplot(x="weight",y="acceleration",data=df,c="b")
```

```
<Axes: xlabel='weight', ylabel='acceleration'>
```




```
from sklearn.linear_model import LinearRegression
lr = LinearRegression()
```

```
#20
import pandas as pd
from sklearn.linear_model import LinearRegression
from sklearn.model_selection import train_test_split
from sklearn.metrics import mean_squared_error
temiz_df=df.dropna(inplace=True)
X = df[['horsepower', 'acceleration', 'weight']]
y = df['mpg']
lr.fit(X,y)
temiz_df=df.dropna(inplace=True)
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)


#19
import pandas as pd
df_corr=df["mpg"].corr(df.acceleration)
print(df_corr)
```

```
0.4233285369027874
```

```
#20
#MPG BIZIM Y YNAI TARGET COLONUMUZ DIGER DEGISKENLER ISE X COLONUMUZDUR.
from sklearn.linear_model import LinearRegression
lr = LinearRegression()
lr.fit(X_train,y_train)
target_tahmin=lr.predict(X_test)
mse = mean_squared_error(y_test, target_tahmin)


#20
import pandas as pd
from sklearn.linear_model import LinearRegression
intercept=lr.intercept_
coef=lr.coef_


#20
print(coef),print(intercept),print(lr.score(X_train,y_train))
```

```
[-0.05492359 -0.02881278 -0.00580125]
47.115174659445685
0.7150226018087609
(None, None, None)
```

```
eldeki_veri=[[130,13,3500]]
eldeki_df=pd.DataFrame(eldeki_veri)
eldeki_df.values.reshape(-1,1)
tahmini_mpg=lr.predict(eldeki_df)
##Tahmin yapti fakat gorsellestirmede eklentisiz kaldi
```

```
/usr/local/lib/python3.10/dist-packages/sklearn/base.py:439: UserWarning: X does not have valid feature names, but LinearRegression was fitted with feature names
  warnings.warn(
```

```
#23
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.ensemble import RandomForestClassifier
from sklearn.metrics import accuracy_score

veri = ['mpg', 'displacement', 'horsepower', 'weight', 'acceleration']
X = df[veri]
y = df['origin']


#23
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.8, random_state=42)



#23
from sklearn.ensemble import RandomForestClassifier
rf=RandomForestClassifier(random_state=42)


#23
rf.fit(X_train,y_train)
```

RandomForestClassifier
RandomForestClassifier(random_state=42)

#24
```
y_pred=rf.predict(X_test)
dogruluk=accuracy_score(y_test, y_pred)
print(dogruluk)
```

```
0.7579617834394905
```

#25
```
onem_sirasi=rf.feature_importances_
print(onem_sirasi)
```

```
[0.18548811 0.3211235  0.15512013 0.18508231 0.15318596]
```