In [1]:
```
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
import pandas as pd
from sklearn.preprocessing import StandardScaler
from IPython.display import display, HTML
import re
```

In [2]:
```
data = pd.read_csv('sign_language_keypoints_csv.csv', delimiter = ";")
df = data.copy()
```

In [3]:
```
def getting_primary_info(df):
    print("-------------------------------------------------------")
    print("Veri setinin şekli", df.shape)
    print("-------------------------------------------------------")
    # print("Veri seti değişken tipleri:\n", df.dtypes)
    print("-------------------------------------------------------")
    print("Veri setinin ilk 5 satırı")
    display(HTML(df.head().to_html()))
    print("-------------------------------------------------------")
    print("Veri setinin istatistiki verileri")
    description = df.describe()
    display(HTML(description.to_html()))
    print("-------------------------------------------------------")
getting_primary_info(df)
```

```
-------------------------------------------------------
Veri setinin şekli (664, 43)
-------------------------------------------------------
-------------------------------------------------------
Veri setinin ilk 5 satırı
```

	Harf	X1	X2	X3	X4	X5	X6	X7	X8	X9	X10	X11	X12	X13	X14	X15	X16	X17	X18	X19	X20	X21	Y1	Y2	Y3	Y4	Y5	Y6	Y7	Y8	Y9	Y10	Y11	Y12	Y13	Y14	Y15	Y16	Y17	Y18	Y19	Y20	Y21
0	a	97	144	184	198	192	157	163	156	152	125	130	126	125	93	96	99	102	59	66	74	78	197	176	134	90	53	92	60	93	122	90	57	101	133	92	63	108	139	99	70	98	121
1	a	109	142	173	177	160	155	148	147	151	129	123	129	135	106	100	107	113	82	78	87	92	185	170	138	98	77	99	74	105	115	102	78	119	121	108	91	126	129	115	104	128	133
2	a	119	143	156	158	164	135	147	146	142	120	134	132	130	106	120	121	121	94	106	111	114	167	151	127	107	91	104	109	127	139	104	114	133	147	108	119	136	148	115	120	133	142
3	a	116	137	150	154	163	139	143	142	139	124	129	128	127	109	114	117	118	94	100	104	105	164	150	126	106	92	107	93	110	124	108	96	117	133	112	103	124	138	117	106	119	130
4	a	121	158	188	196	188	153	161	166	167	123	133	142	144	93	106	117	119	63	78	91	94	203	178	129	88	55	87	66	103	128	93	76	120	144	103	87	129	151	117	102	130	147

```
-------------------------------------------------------
Veri setinin istatistiki verileri
```

	X1	X2	X3	X4	X5	X6	X7	X8	X9	X10	X11	X12	X13	X14	X15	X16	X17	X18	6
count	664.000000	664.000000	664.000000	664.000000	664.000000	664.000000	664.000000	664.000000	664.000000	664.000000	664.000000	664.000000	664.000000	664.000000	664.000000	664.000000	664.000000	664.000000	
mean	129.725904	148.661145	156.230422	148.843373	138.632530	141.343373	142.013554	142.295181	141.183735	125.156627	124.167169	128.109940	129.787651	110.769578	109.682229	118.721386	123.945783	97.650602	
std	32.362905	23.429389	18.041002	21.009217	29.889801	20.071430	24.947634	29.601683	35.375162	15.677640	16.438855	20.774991	25.665767	16.188079	14.235572	17.256973	19.996759	21.822779	
min	43.000000	61.000000	79.000000	77.000000	71.000000	69.000000	53.000000	39.000000	24.000000	57.000000	42.000000	27.000000	12.000000	55.000000	52.000000	62.000000	66.000000	40.000000	
25%	109.000000	133.000000	145.000000	136.000000	118.000000	130.000000	128.000000	131.000000	130.000000	118.000000	115.000000	118.000000	119.000000	103.000000	101.000000	108.000000	112.000000	84.000000	
50%	124.000000	149.000000	158.000000	149.000000	135.000000	144.500000	148.000000	150.000000	150.000000	127.000000	127.000000	131.000000	133.000000	110.000000	110.000000	118.000000	122.000000	94.000000	
75%	141.250000	163.000000	168.000000	162.000000	158.000000	156.000000	161.000000	162.000000	165.000000	134.000000	135.000000	142.000000	144.250000	119.000000	119.000000	129.000000	135.000000	108.000000	1
max	229.000000	209.000000	195.000000	211.000000	228.000000	189.000000	191.000000	200.000000	226.000000	193.000000	165.000000	187.000000	209.000000	192.000000	158.000000	189.000000	201.000000	189.000000	1

```
-------------------------------------------------------
```

In [5]:
```
df.head()
```

Out[5]:

	Harf	X1	X2	X3	X4	X5	X6	X7	X8	X9	...	Y12	Y13	Y14	Y15	Y16	Y17	Y18	Y19	Y20	Y21
0	a	97	144	184	198	192	157	163	156	152	...	101	133	92	63	108	139	99	70	98	121
1	a	109	142	173	177	160	155	148	147	151	...	119	121	108	91	126	129	115	104	128	133
2	a	119	143	156	158	164	135	147	146	142	...	133	147	108	119	136	148	115	120	133	142
3	a	116	137	150	154	163	139	143	142	139	...	117	133	112	103	124	138	117	106	119	130
4	a	121	158	188	196	188	153	161	166	167	...	120	144	103	87	129	151	117	102	130	147

5 rows × 43 columns

In [6]:
```
df.to_csv("hand_sign_deneme.csv",index = False)
```

In [7]:
```
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LogisticRegression
from sklearn.metrics import accuracy_score

X = df.drop(columns="Harf")
y = df['Harf']

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=47)

model = LogisticRegression()
model.fit(X_train, y_train)

y_pred = model.predict(X_test)

accuracy = accuracy_score(y_test, y_pred)
print("Test Accuracy:", accuracy)

y_pred_train = model.predict(X_train)
accuracy_train = accuracy_score(y_train, y_pred_train)
print("Train Accuracy:", accuracy_train)
```

```
Test Accuracy: 0.8796992481203008
Train Accuracy: 0.9981167608286252
C:\Users\Baki Akgun\New folder\Lib\site-packages\sklearn\linear_model\_logistic.py:460: ConvergenceWarning: lbfgs failed to converge (status=1):
STOP: TOTAL NO. of ITERATIONS REACHED LIMIT.

Increase the number of iterations (max_iter) or scale the data as shown in:
    https://scikit-learn.org/stable/modules/preprocessing.html
Please also refer to the documentation for alternative solver options:
```

```
    https://scikit-learn.org/stable/modules/linear_model.html#logistic-regression
  n_iter_i = _check_optimize_result(
```

In [ ]: