

2203a51521-ml3

March 18, 2024

```
[2]: import pandas as pd
d1= pd.read_csv('/content/test.csv')
d2=pd.read_csv('/content/train.csv')
print(d1)
print(d2)
```

	id	battery_power	blue	clock_speed	dual_sim	fc	four_g	int_memory	\
0	1	1043	1	1.8	1	14	0	5	
1	2	841	1	0.5	1	4	1	61	
2	3	1807	1	2.8	0	1	0	27	
3	4	1546	0	0.5	1	18	1	25	
4	5	1434	0	1.4	0	11	1	49	
..	...	...	...	...	...	...	...	...	
995	996	1700	1	1.9	0	0	1	54	
996	997	609	0	1.8	1	0	0	13	
997	998	1185	0	1.4	0	1	1	8	
998	999	1533	1	0.5	1	0	0	50	
999	1000	1270	1	0.5	0	4	1	35	

	m_dep	mobile_wt	...	pc	px_height	px_width	ram	sc_h	sc_w	\
0	0.1	193	...	16	226	1412	3476	12	7	
1	0.8	191	...	12	746	857	3895	6	0	
2	0.9	186	...	4	1270	1366	2396	17	10	
3	0.5	96	...	20	295	1752	3893	10	0	
4	0.5	108	...	18	749	810	1773	15	8	
..	...	...	...	...	...	...	...	...	...	
995	0.5	170	...	17	644	913	2121	14	8	
996	0.9	186	...	2	1152	1632	1933	8	1	
997	0.5	80	...	12	477	825	1223	5	0	
998	0.4	171	...	12	38	832	2509	15	11	
999	0.1	140	...	19	457	608	2828	9	2	

	talk_time	three_g	touch_screen	wifi
0	2	0	1	0
1	7	1	0	0
2	10	0	1	1
3	7	1	1	0
4	7	1	0	1

..	...	...	...	...	...
995	15	1		1	0
996	19	0		1	1
997	14	1		0	0
998	6	0		1	0
999	3	1		0	1

[1000 rows x 21 columns]

	battery_power	blue	clock_speed	dual_sim	fc	four_g	int_memory	\
0	842	0	2.2	0	1	0	7	
1	1021	1	0.5	1	0	1	53	
2	563	1	0.5	1	2	1	41	
3	615	1	2.5	0	0	0	10	
4	1821	1	1.2	0	13	1	44	
...	...	...	...	...	...	...	...	
1995	794	1	0.5	1	0	1	2	
1996	1965	1	2.6	1	0	0	39	
1997	1911	0	0.9	1	1	1	36	
1998	1512	0	0.9	0	4	1	46	
1999	510	1	2.0	1	5	1	45	

	m_dep	mobile_wt	n_cores	...	px_height	px_width	ram	sc_h	sc_w	\
0	0.6	188	2	...	20	756	2549	9	7	
1	0.7	136	3	...	905	1988	2631	17	3	
2	0.9	145	5	...	1263	1716	2603	11	2	
3	0.8	131	6	...	1216	1786	2769	16	8	
4	0.6	141	2	...	1208	1212	1411	8	2	
...	...	...	...	...	...	...	...	...	...	
1995	0.8	106	6	...	1222	1890	668	13	4	
1996	0.2	187	4	...	915	1965	2032	11	10	
1997	0.7	108	8	...	868	1632	3057	9	1	
1998	0.1	145	5	...	336	670	869	18	10	
1999	0.9	168	6	...	483	754	3919	19	4	

	talk_time	three_g	touch_screen	wifi	price_range
0	19	0	0	1	1
1	7	1	1	0	2
2	9	1	1	0	2
3	11	1	0	0	2
4	15	1	1	0	1
...	...	...	...	...	...
1995	19	1	1	0	0
1996	16	1	1	1	2
1997	5	1	1	0	3
1998	19	1	1	1	0
1999	2	1	1	1	3

[2000 rows x 21 columns]

```
[11]: x= d2.drop('touch_screen', axis=1)
      y= d2['touch_screen']
      print(x)
```

	battery_power	blue	clock_speed	dual_sim	fc	four_g	int_memory	\
0	842	0	2.2	0	1	0	7	
1	1021	1	0.5	1	0	1	53	
2	563	1	0.5	1	2	1	41	
3	615	1	2.5	0	0	0	10	
4	1821	1	1.2	0	13	1	44	
...	...	...	...	...	...	...	...	
1995	794	1	0.5	1	0	1	2	
1996	1965	1	2.6	1	0	0	39	
1997	1911	0	0.9	1	1	1	36	
1998	1512	0	0.9	0	4	1	46	
1999	510	1	2.0	1	5	1	45	

	m_dep	mobile_wt	n_cores	pc	px_height	px_width	ram	sc_h	sc_w	\
0	0.6	188	2	2	20	756	2549	9	7	
1	0.7	136	3	6	905	1988	2631	17	3	
2	0.9	145	5	6	1263	1716	2603	11	2	
3	0.8	131	6	9	1216	1786	2769	16	8	
4	0.6	141	2	14	1208	1212	1411	8	2	
...	...	...	...	...	...	...	...	...	...	
1995	0.8	106	6	14	1222	1890	668	13	4	
1996	0.2	187	4	3	915	1965	2032	11	10	
1997	0.7	108	8	3	868	1632	3057	9	1	
1998	0.1	145	5	5	336	670	869	18	10	
1999	0.9	168	6	16	483	754	3919	19	4	

	talk_time	three_g	wifi	price_range
0	19	0	1	1
1	7	1	0	2
2	9	1	0	2
3	11	1	0	2
4	15	1	0	1
...	...	...	...	...
1995	19	1	0	0
1996	16	1	1	2
1997	5	1	0	3
1998	19	1	1	0
1999	2	1	1	3

[2000 rows x 20 columns]

```
[20]: X_test = d2.drop('touch_screen', axis=1)
      y_test = d2['touch_screen']
```

[28]:

```
[40]: from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2,
↳ random_state=42)
```

[41]: print(X\_train)

	battery_power	blue	clock_speed	dual_sim	fc	four_g	int_memory	\
968	1923	0	0.5	1	7	0	46	
240	633	1	2.2	0	0	1	49	
819	1236	0	0.9	1	2	1	57	
692	781	0	1.1	0	2	0	38	
420	1456	1	0.5	1	7	0	7	
...	...	...	...	...	...	...	...	
1130	1975	1	1.9	1	2	0	31	
1294	589	1	0.5	0	1	1	59	
860	1829	1	0.5	0	0	1	15	
1459	1927	0	0.9	1	3	0	11	
1126	635	1	0.6	1	1	1	50	

	m_dep	mobile_wt	n_cores	pc	px_height	px_width	ram	sc_h	sc_w	\
968	0.5	191	1	10	767	1759	1489	10	9	
240	0.1	139	8	1	529	1009	3560	11	1	
819	0.1	188	1	14	517	809	1406	14	12	
692	0.4	198	5	7	304	1674	3508	13	8	
420	0.4	105	5	12	823	1104	1587	6	5	
...	...	...	...	...	...	...	...	...	...	
1130	0.9	151	1	17	775	1607	3022	13	5	
1294	0.7	146	8	4	759	1858	362	16	10	
860	0.4	160	5	7	729	1267	2080	16	11	
1459	0.4	190	8	12	491	1506	2916	16	11	
1126	0.3	97	5	13	193	989	2107	13	12	

	talk_time	three_g	wifi	price_range
968	3	1	1	1
240	16	1	1	2
819	20	1	1	0
692	5	0	1	3
420	20	1	1	1
...	...	...	...	...
1130	19	0	1	3
1294	6	1	1	0
860	12	1	1	2
1459	18	0	1	3
1126	12	1	0	1

[1600 rows x 20 columns]

```
[42]: print(y_train)
```

```
968      1
240      1
819      0
692      0
420      0
..
1130     0
1294     1
860      0
1459     1
1126     0
Name: touch_screen, Length: 1600, dtype: int64
```

```
[43]: from sklearn.preprocessing import MinMaxScaler
scaler = MinMaxScaler()
X_train_scaled = scaler.fit_transform(x)
X_test_scaled = scaler.fit_transform(X_test)
print(X_train_scaled)
```

```
[[0.22778891 0.          0.68      ... 0.          1.          0.33333333]
 [0.34736139 1.          0.          ... 1.          0.          0.66666667]
 [0.04141617 1.          0.          ... 1.          0.          0.66666667]
 ...
 [0.94188377 0.          0.16      ... 1.          0.          1.          ]
 [0.6753507  0.          0.16      ... 1.          1.          0.          ]
 [0.00601202 1.          0.6        ... 1.          1.          1.          ]]
```

```
[34]: from sklearn.linear_model import LogisticRegression
from sklearn.metrics import accuracy_score
model = LogisticRegression()
model.fit(X_train_scaled, y)
y_pred = model.predict(X_test_scaled)
accuracy = accuracy_score(y_test, y_pred)
print(f'Accuracy: {accuracy}')
```

Accuracy: 0.5425

```
[44]: from sklearn.metrics import accuracy_score, precision_score, recall_score, \
      ↪confusion_matrix
accuracy = accuracy_score(y_test, y_pred)
precision = precision_score(y_test, y_pred, average='weighted')
recall = recall_score(y_test, y_pred, average='weighted')
conf_matrix = confusion_matrix(y_test, y_pred)
print("Accuracy:", accuracy)
```

```
print("Precision:", precision)
print("Recall:", recall)
print("Confusion Matrix:")
print(conf_matrix)
```

Accuracy: 0.5425

Precision: 0.5436003185115903

Recall: 0.5425

Confusion Matrix:

[[ 99 103]

[ 80 118]]