

```
import numpy as np
import pandas as pd
df=pd.read_csv('/content/archive (2).zip')
```

```
df.head()
```

	Age	Gender	Height	Weight	CALC	FAVC	FCVC	NCP	SCC	SMOKE	CH2O	family_history_with_overweight	FAF	TUE	CAEC		
0	21.0	Female	1.62	64.0	no	no	2.0	3.0	no	no	2.0		yes	0.0	1.0	Sometimes	Public_
1	21.0	Female	1.52	56.0	Sometimes	no	3.0	3.0	yes	yes	3.0		yes	3.0	0.0	Sometimes	Public_
2	23.0	Male	1.80	77.0	Frequently	no	2.0	3.0	no	no	2.0		yes	2.0	1.0	Sometimes	Public_
3	27.0	Male	1.80	87.0	Frequently	no	3.0	3.0	no	no	2.0		no	2.0	0.0	Sometimes	
4	22.0	Male	1.78	89.8	Sometimes	no	2.0	1.0	no	no	2.0		no	0.0	0.0	Sometimes	Public_

```
df.tail()
```

⊗

	Age	Gender	Height	Weight	CALC	FAVC	FCVC	NCP	SCC	SMOKE	CH2O	family_history_with_overweight	FAF
2106	20.976842	Female	1.710730	131.408528	Sometimes	yes	3.0	3.0	no	no	1.728139	yes	1.676269
2107	21.982942	Female	1.748584	133.742943	Sometimes	yes	3.0	3.0	no	no	2.005130	yes	1.341390
2108	22.524036	Female	1.752206	133.689352	Sometimes	yes	3.0	3.0	no	no	2.054193	yes	1.414209
2109	24.361936	Female	1.739450	133.346641	Sometimes	yes	3.0	3.0	no	no	2.852339	yes	1.139107
2110	23.664709	Female	1.738836	133.472641	Sometimes	yes	3.0	3.0	no	no	2.863513	yes	1.026452

```
df.isna().sum()
```

```
Age          0
Gender       0
Height       0
Weight       0
CALC         0
FAVC         0
FCVC         0
NCP          0
SCC          0
SMOKE        0
CH2O         0
family_history_with_overweight  0
FAF          0
TUE          0
CAEC         0
MTRANS       0
NObeyesdad   0
dtype: int64
```

```
df.dtypes
```

```
Age          float64
Gender       object
Height       float64
Weight       float64
CALC         object
FAVC         object
FCVC         float64
NCP          float64
SCC          object
SMOKE        object
CH2O         float64
family_history_with_overweight  object
FAF          float64
TUE          float64
CAEC         object
MTRANS       object
NObeyesdad   object
dtype: object
```

```
#encoding
from sklearn.preprocessing import LabelEncoder
```

```

lb=LabelEncoder()
df['Gender']=lb.fit_transform(df['Gender'])
df['CALC']=lb.fit_transform(df['CALC'])
df['FAVC']=lb.fit_transform(df['FAVC'])
df['SCC']=lb.fit_transform(df['SCC'])
df['SMOKE']=lb.fit_transform(df['SMOKE'])
df['family_history_with_overweight']=lb.fit_transform(df['family_history_with_overweight'])
df['CAEC']=lb.fit_transform(df['CAEC'])
df['MTRANS']=lb.fit_transform(df['MTRANS'])
df['NObeyesdad']=lb.fit_transform(df['NObeyesdad'])

```

```

x=df.iloc[:, :-1].values
y=df.iloc[:, -1].values

```

```

from sklearn.model_selection import train_test_split
x_train,x_test,y_train,y_test=train_test_split(x,y,train_size=0.30,random_state=42)

```

```

from sklearn.preprocessing import MinMaxScaler
scalar=MinMaxScaler()
scalar.fit(x_train)
x_train=scalar.transform(x_train)
x_test=scalar.transform(x_test)

```

```

from sklearn.neighbors import KNeighborsClassifier
from sklearn.naive_bayes import BernoulliNB
from sklearn.svm import SVC
from sklearn.metrics import confusion_matrix,accuracy_score,classification_report
knn=KNeighborsClassifier(n_neighbors=7)
nb=BernoulliNB()
sv=SVC()
lst=[knn,nb,sv]

```

```

for i in lst:
    print("Model started")
    print(i)
    i.fit(x_train,y_train)
    print("predicted value is")
    y_pred=i.predict(x_test)
    print(y_pred)
    print("confusion metrics is")
    print(confusion_matrix(y_test,y_pred))
    print("accuracy score is")
    print(accuracy_score(y_test,y_pred))
    print("performance report is")
    print(classification_report(y_test,y_pred))

```

```

Model started
KNeighborsClassifier(n_neighbors=7)
predicted value is
[0 4 0 ... 0 3 1]
confusion metrics is
[[153  9  8  0  0  5 12]
 [ 73 54 25  9  1 16 39]
 [  1  3 184 31  6  6 18]
 [  1  1  6 204  0  0  3]
 [  0  0  0  1 211  0  0]
 [ 35 13 41 13  3 76 22]
 [ 13  6 31 22  4 10 109]]
accuracy score is
0.6705006765899865
performance report is

```

	precision	recall	f1-score	support
0	0.55	0.82	0.66	187
1	0.63	0.25	0.36	217
2	0.62	0.74	0.68	249
3	0.73	0.95	0.82	215
4	0.94	1.00	0.97	212
5	0.67	0.37	0.48	203
6	0.54	0.56	0.55	195
accuracy			0.67	1478
macro avg	0.67	0.67	0.64	1478
weighted avg	0.67	0.67	0.65	1478

```

Model started

```

```
BernoulliNB()
predicted value is
[4 4 5 ... 1 5 1]
confusion metrics is
[[ 84 16  3 56 22  3  3]
 [ 45 73  8 21 14 27 29]
 [  1  8 43 112 39 24 22]
 [  0  6  0 189  0  0 20]
 [  0  0  0  1 211  0  0]
 [ 24 19 27 58 26 43  6]
 [  4 15 25 78  9 10 54]]
accuracy score is
0.47158322056833557
performance report is
```

	precision	recall	f1-score	support
0	0.53	0.45	0.49	187
1	0.53	0.34	0.41	217
2	0.41	0.17	0.24	249
3	0.37	0.88	0.52	215
4	0.66	1.00	0.79	212
5	0.40	0.21	0.28	203
6	0.40	0.28	0.33	195
accuracy			0.47	1478
macro avg	0.47	0.47	0.44	1478
weighted avg	0.47	0.47	0.43	1478