

```
import numpy as np
import pandas as pd
df=pd.read_csv('/content/apple_quality.csv')
df
```

	A_id	Size	Weight	Sweetness	Crunchiness	Juiciness	Ripeness	
0	0.0	-3.970049	-2.512336	5.346330	-1.012009	1.844900	0.329840	
1	1.0	-1.195217	-2.839257	3.664059	1.588232	0.853286	0.867530	
2	2.0	-0.292024	-1.351282	-1.738429	-0.342616	2.838636	-0.038033	
3	3.0	-0.657196	-2.271627	1.324874	-0.097875	3.637970	-3.413761	
4	4.0	1.364217	-1.296612	-0.384658	-0.553006	3.030874	-1.303849	
...	
3996	3996.0	-0.293118	1.949253	-0.204020	-0.640196	0.024523	-1.087900	
3997	3997.0	-2.634515	-2.138247	-2.440461	0.657223	2.199709	4.763859	
3998	3998.0	-4.008004	-1.779337	2.366397	-0.200329	2.161435	0.214488	
3999	3999.0	0.278540	-1.715505	0.121217	-1.154075	1.266677	-0.776571	
4000	NaN	NaN	NaN	NaN	NaN	NaN	NaN	Created_by_Nid

4001 rows × 9 columns

```
df.drop(['A_id'],axis=1,inplace=True)
df
```

	Size	Weight	Sweetness	Crunchiness	Juiciness	Ripeness	
0	-3.970049	-2.512336	5.346330	-1.012009	1.844900	0.329840	
1	-1.195217	-2.839257	3.664059	1.588232	0.853286	0.867530	
2	-0.292024	-1.351282	-1.738429	-0.342616	2.838636	-0.038033	
3	-0.657196	-2.271627	1.324874	-0.097875	3.637970	-3.413761	
4	1.364217	-1.296612	-0.384658	-0.553006	3.030874	-1.303849	
...	
3996	-0.293118	1.949253	-0.204020	-0.640196	0.024523	-1.087900	
3997	-2.634515	-2.138247	-2.440461	0.657223	2.199709	4.763859	
3998	-4.008004	-1.779337	2.366397	-0.200329	2.161435	0.214488	
3999	0.278540	-1.715505	0.121217	-1.154075	1.266677	-0.776571	
4000	NaN	NaN	NaN	NaN	NaN	NaN	Created_by_Nid

4001 rows × 8 columns

```
df.tail()
```

	Size	Weight	Sweetness	Crunchiness	Juiciness	Ripeness	
3996	-0.293118	1.949253	-0.204020	-0.640196	0.024523	-1.087900	
3997	-2.634515	-2.138247	-2.440461	0.657223	2.199709	4.763859	
3998	-4.008004	-1.779337	2.366397	-0.200329	2.161435	0.214488	
3999	0.278540	-1.715505	0.121217	-1.154075	1.266677	-0.776571	
4000	NaN	NaN	NaN	NaN	NaN	NaN	Created_by_Nid

```
df.head()
```

	Size	Weight	Sweetness	Crunchiness	Juiciness	Ripeness	Acidity	Quality
0	-3.970049	-2.512336	5.346330	-1.012009	1.844900	0.329840	-0.491590483	5
1	-1.195217	-2.839257	3.664059	1.588232	0.853286	0.867530	-0.722809367	5
2	-0.292024	-1.351282	-1.738429	-0.342616	2.838636	-0.038033	2.621636473	5
3	-0.657196	-2.271627	1.324874	-0.097875	3.637970	-3.413761	0.790723217	5
4	1.364217	-1.296612	-0.384658	-0.553006	3.030874	-1.303849	0.501984036	5

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

```
Size          1
Weight        1
Sweetness     1
Crunchiness   1
Juiciness     1
Ripeness      1
Acidity       0
Quality       1
dtype: int64
```

```
df['Size']=df['Size'].fillna(df['Size'].mean())
df['Weight']=df['Weight'].fillna(df['Weight'].mean())
df['Sweetness']=df['Sweetness'].fillna(df['Sweetness'].mean())
df['Crunchiness']=df['Crunchiness'].fillna(df['Crunchiness'].mean())
df['Juiciness']=df['Juiciness'].fillna(df['Juiciness'].mean())
df['Ripeness']=df['Ripeness'].fillna(df['Ripeness'].mean())
df['Quality']=df['Quality'].fillna(df['Quality'].mode()[0])
df.isna().sum()
```

```
Size          0
Weight        0
Sweetness     0
Crunchiness   0
Juiciness     0
Ripeness      0
Acidity       0
Quality       0
dtype: int64
```

```
df.dtypes
```

```
Size          float64
Weight        float64
Sweetness     float64
Crunchiness   float64
Juiciness     float64
Ripeness      float64
Acidity       object
Quality       object
dtype: object
```

```
from sklearn.preprocessing import LabelEncoder
encode=LabelEncoder()
df['Quality']=encode.fit_transform(df['Quality'])
```

```
df['Acidity']=encode.fit_transform(df['Acidity'])
```

```
df.dtypes
```

```
Size          float64
Weight        float64
Sweetness     float64
Crunchiness   float64
Juiciness     float64
Ripeness      float64
Acidity       int64
Quality       int64
dtype: object
```

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

```
array([[ -3.97004852e+00,  -2.51233638e+00,   5.34632961e+00,  ...,
         1.84490036e+00,   3.29839797e-01,   3.97000000e+02],
       [ -1.19521719e+00,  -2.83925653e+00,   3.66405876e+00,  ...,
         8.53285795e-01,   8.67530082e-01,   5.50000000e+02],
       [ -2.92023862e-01,  -1.35128199e+00,  -1.73842916e+00,  ...,
         2.83863551e+00,  -3.80333280e-02,   3.54300000e+03],
       ...,
       [ -4.00800374e+00,  -1.77933711e+00,   2.36639697e+00,  ...,
         2.16143512e+00,   2.14488384e-01,   1.42800000e+03],
       [  2.78539650e-01,  -1.71550503e+00,   1.21217251e-01,  ...,
         1.26667740e+00,  -7.76571470e-01,   3.04100000e+03],
       [ -5.03014630e-01,  -9.89546545e-01,  -4.70478520e-01,  ...,
         5.12117968e-01,   4.98277428e-01,   4.00000000e+03]])
```

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

```
array([1, 1, 0, ..., 1, 1, 1])

from sklearn.model_selection import train_test_split
x_train,x_test,y_train,y_test=train_test_split(x,y,test_size=0.30,random_state=1)
x_train

array([[ -1.54241097e+00, -9.42480815e-01, -5.08999257e-01, ...,
        -8.19818136e-01,  8.48231585e-01,  2.64800000e+03],
       [ 2.94586351e-01, -3.53755252e-01, -5.40281839e-01, ...,
        6.71989996e-01,  1.29183819e+00,  2.63100000e+03],
       [-5.14301592e-01, -2.78214336e+00,  3.14528232e+00, ...,
        9.31536359e-01,  1.03334310e+00,  3.61000000e+02],
       ...,
       [-2.49258269e+00, -2.06591657e+00,  1.08117994e+00, ...,
       -1.60072238e+00, -2.83194749e+00,  2.43300000e+03],
       [-4.98846599e+00, -1.90696332e-01,  3.45144007e+00, ...,
        1.72614427e+00, -8.59105780e-02,  1.24900000e+03],
       [-6.09625194e-01, -1.74773411e+00,  9.44680078e-01, ...,
       -3.97477816e-01,  1.17733095e+00,  3.17200000e+03]])

from sklearn.preprocessing import StandardScaler
scale=StandardScaler()
scale.fit(x_train)
x_train=scale.transform(x_train)
x_test=scale.transform(x_test)

from sklearn.svm import SVC
model=SVC()
model.fit(x_train,y_train)
y_pred=model.predict(x_test)
y_pred

array([0, 1, 1, ..., 1, 0, 0])

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

[[534  76]
 [ 64 527]]

score=accuracy_score(y_test,y_pred)
score

0.8834304746044963
```