

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

| | pH | Temprature | Taste | Odor | Fat | Turbidity | Colour | Grade |
|------|-----|------------|-------|------|-----|-----------|--------|--------|
| 0 | 6.6 | 35 | 1 | 0 | 1 | 0 | 254 | high |
| 1 | 6.6 | 36 | 0 | 1 | 0 | 1 | 253 | high |
| 2 | 8.5 | 70 | 1 | 1 | 1 | 1 | 246 | low |
| 3 | 9.5 | 34 | 1 | 1 | 0 | 1 | 255 | low |
| 4 | 6.6 | 37 | 0 | 0 | 0 | 0 | 255 | medium |
| ... | ... | ... | ... | ... | ... | ... | ... | ... |
| 1054 | 6.7 | 45 | 1 | 1 | 0 | 0 | 247 | medium |
| 1055 | 6.7 | 38 | 1 | 0 | 1 | 0 | 255 | high |
| 1056 | 3.0 | 40 | 1 | 1 | 1 | 1 | 255 | low |
| 1057 | 6.8 | 43 | 1 | 0 | 1 | 0 | 250 | high |
| 1058 | 8.6 | 55 | 0 | 1 | 1 | 1 | 255 | low |

1059 rows × 8 columns

```
df.head()
```

| | pH | Temprature | Taste | Odor | Fat | Turbidity | Colour | Grade |
|---|-----|------------|-------|------|-----|-----------|--------|--------|
| 0 | 6.6 | 35 | 1 | 0 | 1 | 0 | 254 | high |
| 1 | 6.6 | 36 | 0 | 1 | 0 | 1 | 253 | high |
| 2 | 8.5 | 70 | 1 | 1 | 1 | 1 | 246 | low |
| 3 | 9.5 | 34 | 1 | 1 | 0 | 1 | 255 | low |
| 4 | 6.6 | 37 | 0 | 0 | 0 | 0 | 255 | medium |

```
df.tail()
```

| | pH | Temprature | Taste | Odor | Fat | Turbidity | Colour | Grade |
|------|-----|------------|-------|------|-----|-----------|--------|--------|
| 1054 | 6.7 | 45 | 1 | 1 | 0 | 0 | 247 | medium |
| 1055 | 6.7 | 38 | 1 | 0 | 1 | 0 | 255 | high |
| 1056 | 3.0 | 40 | 1 | 1 | 1 | 1 | 255 | low |
| 1057 | 6.8 | 43 | 1 | 0 | 1 | 0 | 250 | high |
| 1058 | 8.6 | 55 | 0 | 1 | 1 | 1 | 255 | low |

```
df.shape
```

(1059, 8)

```
df.dtypes
```

```
pH          float64
Temprature   int64
Taste        int64
Odor          int64
Fat           int64
Turbidity     int64
Colour        int64
Grade        object
dtype: object
```

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

```
pH          0
Temprature   0
Taste        0
Odor          0
```

```
df.describe()
```



Training and Testing data

```
array(['high', 'high', 'low', ..., 'low', 'high', 'low'], dtype=object)
```

Training and Testing data

```
x_train
```

x_test

y_train

```
array(['high', 'high', 'low', 'low', 'medium', 'low', 'low', 'medium',  
      'medium', 'high', 'medium', 'low', 'medium', 'medium', 'medium',  
      'high', 'high', 'low', 'low', 'high', 'high', 'high', 'low', 'low',  
      'medium', 'low', 'low', 'low', 'high', 'low', 'medium', 'high',  
      'medium', 'medium', 'low', 'medium', 'medium', 'medium', 'high',  
      'medium', 'medium', 'low', 'medium', 'medium', 'medium', 'low',  
      'low', 'high', 'low', 'medium', 'high', 'low', 'low', 'low', 'low',  
      'low', 'low', 'high', 'low', 'high', 'medium', 'high', 'high',  
      'high', 'low', 'high', 'low', 'low', 'low', 'medium', 'high', 'high',  
      'low', 'low', 'medium', 'low', 'low', 'low', 'low', 'low', 'medium',  
      'low', 'low', 'low', 'medium', 'high', 'medium', 'high', 'medium',  
      'high', 'low', 'low', 'medium', 'medium', 'low', 'low', 'low',  
      'high', 'high', 'low', 'low', 'low', 'low', 'low', 'medium', 'low',  
      'high', 'medium', 'high', 'medium', 'medium', 'low', 'low', 'low',  
      'medium', 'medium', 'medium', 'low', 'low', 'low', 'low', 'low',
```



```
'medium', 'low', 'medium', 'low', 'high', 'medium', 'low', 'low',  
'low', 'high', 'medium', 'medium', 'medium', 'medium', 'medium',  
'medium', 'medium', 'medium', 'low', 'high'], dtype=object)
```

Normalisation

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

x_train

```
array([[ -0.01298657,  -0.73413845,   0.93338744, ...,   0.66666667,
         0.98792712,   0.72665791],
       [ -0.01298657,  -0.73413845,   0.93338744, ...,   0.66666667,
         0.98792712,   0.72665791],
       [ -1.35738312,  -0.63281668,   0.93338744, ...,   0.66666667,
        -1.01222041,   0.72665791],
       ...,
       [ -0.08374428,  -0.83546022,  -1.07136646, ...,   0.66666667,
        -1.01222041,   0.72665791],
       [ -0.01298657,   0.58304458,  -1.07136646, ...,  -1.5         ,
         0.98792712,  -0.47043749],
       [  0.12852886,   0.07643572,   0.93338744, ...,   0.66666667,
        -1.01222041,  -1.66753288]])
```

x_test

```
array([[ -0.01298657,  -0.43017314,   0.93338744, ...,   0.66666667,
         0.98792712,   0.72665791],
       [ -1.49889854,   1.5962623 ,  -1.07136646, ...,   0.66666667,
         0.98792712,  -0.47043749],
       [  1.68519854,  -0.12620782,   0.93338744, ...,   0.66666667,
         0.98792712,  -0.47043749],
       ...,
       [  0.12852886,   0.07643572,  -1.07136646, ...,  -1.5         ,
         0.98792712,   0.72665791],
       [ -0.08374428,   0.58304458,   0.93338744, ...,   0.66666667,
        -1.01222041,  -1.42811381],
       [  0.05777114,  -0.63281668,   0.93338744, ...,   0.66666667,
        -1.01222041,   0.72665791]])
```

Model creation

```
from sklearn.neighbors import KNeighborsClassifier
knn=KNeighborsClassifier(n_neighbors=7)
knn.fit(x_train,y_train)
y_pred=knn.predict(x_test)
y_pred
```

```
array(['high', 'low', 'low', 'medium', 'medium', 'low', 'low', 'high',
       'low', 'medium', 'low', 'medium', 'high', 'medium', 'low', 'low',
       'high', 'medium', 'medium', 'medium', 'low', 'medium', 'high',
       'medium', 'medium', 'medium', 'medium', 'high', 'low', 'low',
       'medium', 'medium', 'high', 'medium', 'high', 'medium', 'medium',
       'high', 'medium', 'medium', 'low', 'low', 'low', 'medium', 'high',
       'low', 'high', 'high', 'medium', 'high', 'medium', 'low', 'low',
       'medium', 'medium', 'low', 'high', 'low', 'low', 'medium',
       'medium', 'low', 'high', 'high', 'high', 'low', 'medium', 'medium',
       'low', 'medium', 'low', 'low', 'high', 'medium', 'low', 'low',
       'low', 'medium', 'low', 'medium', 'medium', 'medium', 'high',
       'high', 'medium', 'medium', 'medium', 'low', 'medium', 'high',
       'low', 'low', 'high', 'medium', 'low', 'medium', 'high', 'medium',
       'low', 'medium', 'low', 'low', 'low', 'low', 'medium', 'medium',
       'low', 'low', 'medium', 'high', 'high', 'medium', 'medium', 'low',
       'high', 'low', 'medium', 'low', 'high', 'high', 'medium', 'low',
       'low', 'low', 'medium', 'high', 'medium', 'medium', 'high',
       'medium', 'low', 'medium', 'low', 'medium', 'low', 'low', 'medium',
       'low', 'medium', 'medium', 'medium', 'high', 'low', 'medium',
       'low', 'low', 'medium', 'low', 'low', 'medium', 'medium', 'medium',
       'medium', 'low', 'medium', 'low', 'low', 'high', 'low', 'high',
       'medium', 'medium', 'medium', 'low', 'medium', 'high', 'low',
       'low', 'medium', 'low', 'medium', 'high', 'low', 'low', 'high',
       'low', 'medium', 'medium', 'medium', 'medium', 'low', 'medium',
       'high', 'low', 'high', 'low', 'medium', 'high', 'medium', 'low',
```

```

'low', 'medium', 'medium', 'low', 'high', 'low', 'low', 'high',
'medium', 'medium', 'medium', 'medium', 'high', 'high', 'medium',
'low', 'medium', 'medium', 'medium', 'medium', 'high', 'high',
'medium', 'medium', 'high', 'low', 'low', 'high', 'medium', 'low',
'medium', 'medium', 'medium', 'medium', 'low', 'high', 'low',
'low', 'medium', 'high', 'high', 'low', 'medium', 'low', 'medium',
'low', 'medium', 'low', 'medium', 'low', 'low', 'high', 'high',
'low', 'low', 'high', 'low', 'medium', 'low', 'low', 'low', 'high',
'medium', 'high', 'medium', 'medium', 'low', 'medium', 'low',
'low', 'medium', 'high', 'high', 'high', 'medium', 'low', 'high',
'high', 'medium', 'medium', 'medium', 'medium', 'low', 'high',
'medium', 'low', 'low', 'low', 'medium', 'high', 'medium', 'high',
'medium', 'medium', 'low', 'low', 'low', 'medium', 'high',
'medium', 'medium', 'low', 'medium', 'high', 'medium', 'medium',
'high', 'high', 'medium', 'low', 'medium', 'low', 'high', 'medium',
'low', 'low', 'low', 'high', 'medium', 'medium', 'medium',
'medium', 'medium', 'medium', 'medium', 'medium', 'low', 'high'],
dtype=object)

```

y_test

```

array(['high', 'low', 'low', 'medium', 'medium', 'low', 'low', 'high',
'low', 'medium', 'low', 'medium', 'high', 'medium', 'low', 'low',
'high', 'medium', 'medium', 'medium', 'low', 'medium', 'high',
'medium', 'medium', 'medium', 'low', 'high', 'low', 'low',
'medium', 'high', 'high', 'medium', 'high', 'medium', 'medium',
'high', 'medium', 'medium', 'low', 'low', 'low', 'medium', 'high',
'low', 'high', 'high', 'medium', 'high', 'medium', 'low', 'low',
'medium', 'medium', 'low', 'high', 'low', 'low', 'medium',
'medium', 'low', 'high', 'high', 'high', 'low', 'medium', 'medium',
'low', 'medium', 'low', 'low', 'high', 'medium', 'low', 'low',
'low', 'medium', 'low', 'medium', 'medium', 'medium', 'high',
'high', 'medium', 'medium', 'medium', 'low', 'medium', 'high',
'low', 'low', 'high', 'medium', 'low', 'medium', 'high', 'high',
'low', 'high', 'low', 'low', 'low', 'low', 'medium', 'medium',
'low', 'low', 'medium', 'high', 'high', 'medium', 'medium', 'low',
'high', 'low', 'medium', 'low', 'low', 'high', 'medium', 'low',
'low', 'low', 'medium', 'high', 'medium', 'medium', 'high',
'medium', 'low', 'medium', 'low', 'medium', 'low', 'low', 'medium',
'low', 'medium', 'medium', 'medium', 'high', 'low', 'medium',
'low', 'low', 'high', 'low', 'low', 'medium', 'medium', 'medium',
'medium', 'low', 'medium', 'low', 'low', 'high', 'low', 'high',
'medium', 'medium', 'medium', 'low', 'medium', 'high', 'low',
'low', 'medium', 'low', 'medium', 'high', 'low', 'low', 'high',
'low', 'medium', 'medium', 'medium', 'medium', 'low', 'medium',
'high', 'low', 'high', 'low', 'medium', 'high', 'medium', 'low',
'high', 'medium', 'medium', 'medium', 'medium', 'low', 'high',
'medium', 'low', 'low', 'low', 'medium', 'high', 'medium', 'high',
'medium', 'high', 'low', 'low', 'low', 'medium', 'high', 'medium',
'low', 'low', 'medium', 'high', 'medium', 'medium', 'high', 'high',
'medium', 'low', 'medium', 'low', 'high', 'medium', 'low', 'low',
'low', 'high', 'medium', 'medium', 'medium', 'medium', 'medium',
'medium', 'medium', 'medium', 'low', 'high'], dtype=object)

```

Performance evaluation

```

from sklearn.metrics import confusion_matrix, accuracy_score
result=confusion_matrix(y_test,y_pred)
score=accuracy_score(y_test,y_pred)
result,score

```

```

(array([[ 69,   0,   7],
       [  1, 112,   2],
       [  0,   0, 127]]),
0.9685534591194969)

```

```

from sklearn.metrics import classification_report
print(classification_report(y_test,y_pred))

```

```

precision    recall  f1-score   support

```

| | | | | |
|--------------|------|------|------|-----|
| high | 0.99 | 0.91 | 0.95 | 76 |
| low | 1.00 | 0.97 | 0.99 | 115 |
| medium | 0.93 | 1.00 | 0.97 | 127 |
| accuracy | | | 0.97 | 318 |
| macro avg | 0.97 | 0.96 | 0.97 | 318 |
| weighted avg | 0.97 | 0.97 | 0.97 | 318 |

[Colab paid products](#) - [Cancel contracts here](#)