```
import pandas as pd
import numpy as np
```

```
df=pd.read_csv('samples_cancer.csv')
df.head()
```

	ID	Clump	UnifSize	UnifShape	MargAdh	SingEpiSize	BareNuc	BlandChrom	NormNucl	Mit	Class
0	1000025	5	1	1	1	2	1	3	1	1	2
1	1002945	5	4	4	5	7	10	3	2	1	2
2	1015425	3	1	1	1	2	2	3	1	1	2
3	1016277	6	8	8	1	3	4	3	7	1	2
4	1017023	4	1	1	3	2	1	3	1	1	2

Next steps: Generate code with df New interactive sheet

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 699 entries, 0 to 698
Data columns (total 11 columns):
 #   Column       Non-Null Count  Dtype 
---  ------       --------------  ----- 
 0   ID           699 non-null    int64 
 1   Clump        699 non-null    int64 
 2   UnifSize     699 non-null    int64 
 3   UnifShape    699 non-null    int64 
 4   MargAdh      699 non-null    int64 
 5   SingEpiSize  699 non-null    int64 
 6   BareNuc      699 non-null    object
 7   BlandChrom   699 non-null    int64 
 8   NormNucl     699 non-null    int64 
 9   Mit          699 non-null    int64 
 10  Class        699 non-null    int64 
dtypes: int64(10), object(1)
memory usage: 60.2+ KB
```

```
df['BareNuc'] = pd.to_numeric(df['BareNuc'], errors='coerce')
df = df.dropna()

X = df[['Clump', 'UnifSize', 'UnifShape', 'MargAdh', 'SingEpiSize',
          'BareNuc', 'BlandChrom', 'NormNucl', 'Mit']]
y = df['Class']
```

```
y = y.map({2:0, 4:1})
```

```
from sklearn.model_selection import train_test_split

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)
```

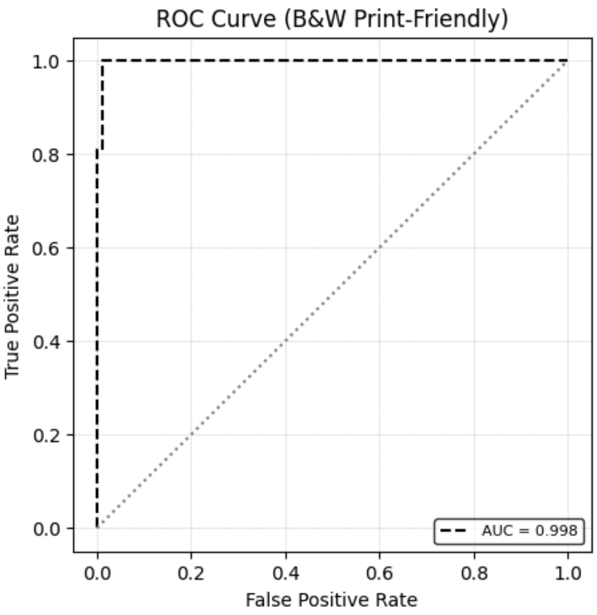
```
from sklearn.linear_model import LogisticRegression
model = LogisticRegression(max_iter=1000)
model.fit(X_train, y_train)
```

```
▾ LogisticRegression  i ?
LogisticRegression(max_iter=1000)
```

```
from sklearn.metrics import confusion_matrix, classification_report, roc_curve, auc
y_pred = model.predict(X_test)

cm = confusion_matrix(y_test, y_pred)
print("Confusion Matrix:\n", cm)

print("\nClassification Report:\n", classification_report(y_test, y_pred, target_names=["Benign", "Malignant"]))
```

```
Confusion Matrix:
 [[78  1]
 [ 5 53]]

Classification Report:
              precision    recall  f1-score   support

      Benign       0.94      0.99      0.96        79
   Malignant       0.98      0.91      0.95        58

    accuracy                           0.96       137
   macro avg       0.96      0.95      0.95       137
weighted avg       0.96      0.96      0.96       137
```

```
import matplotlib.pyplot as plt
y_prob = model.predict_proba(X_test)[:,1]

fpr, tpr, _ = roc_curve(y_test, y_prob)
roc_auc = auc(fpr, tpr)

plt.figure(figsize=(5,5))
plt.plot(fpr, tpr, linestyle='--', color='black', label=f'AUC = {roc_auc:.3f}')
plt.plot([0,1], [0,1], color='gray', linestyle=':')
plt.xlabel('False Positive Rate')
plt.ylabel('True Positive Rate')
plt.title('ROC Curve (B&W Print-Friendly)')
plt.legend(edgecolor='black', fontsize=8)
plt.grid(True, linestyle=':', linewidth=0.5)
plt.show()

print(f"AUC Score: {roc_auc:.4f}")
```



```
AUC Score: 0.9976
```