

Import Required Libraries

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns

from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler
from sklearn.linear_model import LogisticRegression
from sklearn.metrics import classification_report, confusion_matrix, accuracy_score
```

Import dataset

```
from sklearn.datasets import load_breast_cancer
import pandas as pd

# Load dataset
data = load_breast_cancer()
df = pd.DataFrame(data.data, columns=data.feature_names)
df['target'] = data.target
df.head()
```



	mean radius	mean texture	mean perimeter	mean area	mean smoothness	mean compactness	mean concavity	mean concave points	mean symmetry	me fract dimensi
0	17.99	10.38	122.80	1001.0	0.11840	0.27760	0.3001	0.14710	0.2419	0.078
1	20.57	17.77	132.90	1326.0	0.08474	0.07864	0.0869	0.07017	0.1812	0.056
2	19.69	21.25	130.00	1203.0	0.10960	0.15990	0.1974	0.12790	0.2069	0.059
3	11.42	20.38	77.58	386.1	0.14250	0.28390	0.2414	0.10520	0.2597	0.097
4	20.29	14.34	135.10	1297.0	0.10030	0.13280	0.1980	0.10430	0.1809	0.058

5 rows × 31 columns

Data Preprocessing & Exploration

```
df.info()
df.describe()
df.isnull().sum()
sns.heatmap(df.corr(), annot=True)
```



```
<class 'pandas.core.frame.DataFrame'>
```

```
RangeIndex: 569 entries, 0 to 568
```

```
Data columns (total 31 columns):
```

#	Column	Non-Null Count	Dtype
---	-----	-----	-----
0	mean radius	569 non-null	float64
1	mean texture	569 non-null	float64
2	mean perimeter	569 non-null	float64
3	mean area	569 non-null	float64
4	mean smoothness	569 non-null	float64
5	mean compactness	569 non-null	float64
6	mean concavity	569 non-null	float64
7	mean concave points	569 non-null	float64
8	mean symmetry	569 non-null	float64
9	mean fractal dimension	569 non-null	float64
10	radius error	569 non-null	float64
11	texture error	569 non-null	float64
12	perimeter error	569 non-null	float64
13	area error	569 non-null	float64
14	smoothness error	569 non-null	float64
15	compactness error	569 non-null	float64
16	concavity error	569 non-null	float64
17	concave points error	569 non-null	float64
18	symmetry error	569 non-null	float64
19	fractal dimension error	569 non-null	float64
20	worst radius	569 non-null	float64
21	worst texture	569 non-null	float64
22	worst perimeter	569 non-null	float64
23	worst area	569 non-null	float64
24	worst smoothness	569 non-null	float64
25	worst compactness	569 non-null	float64
26	worst concavity	569 non-null	float64
27	worst concave points	569 non-null	float64

```

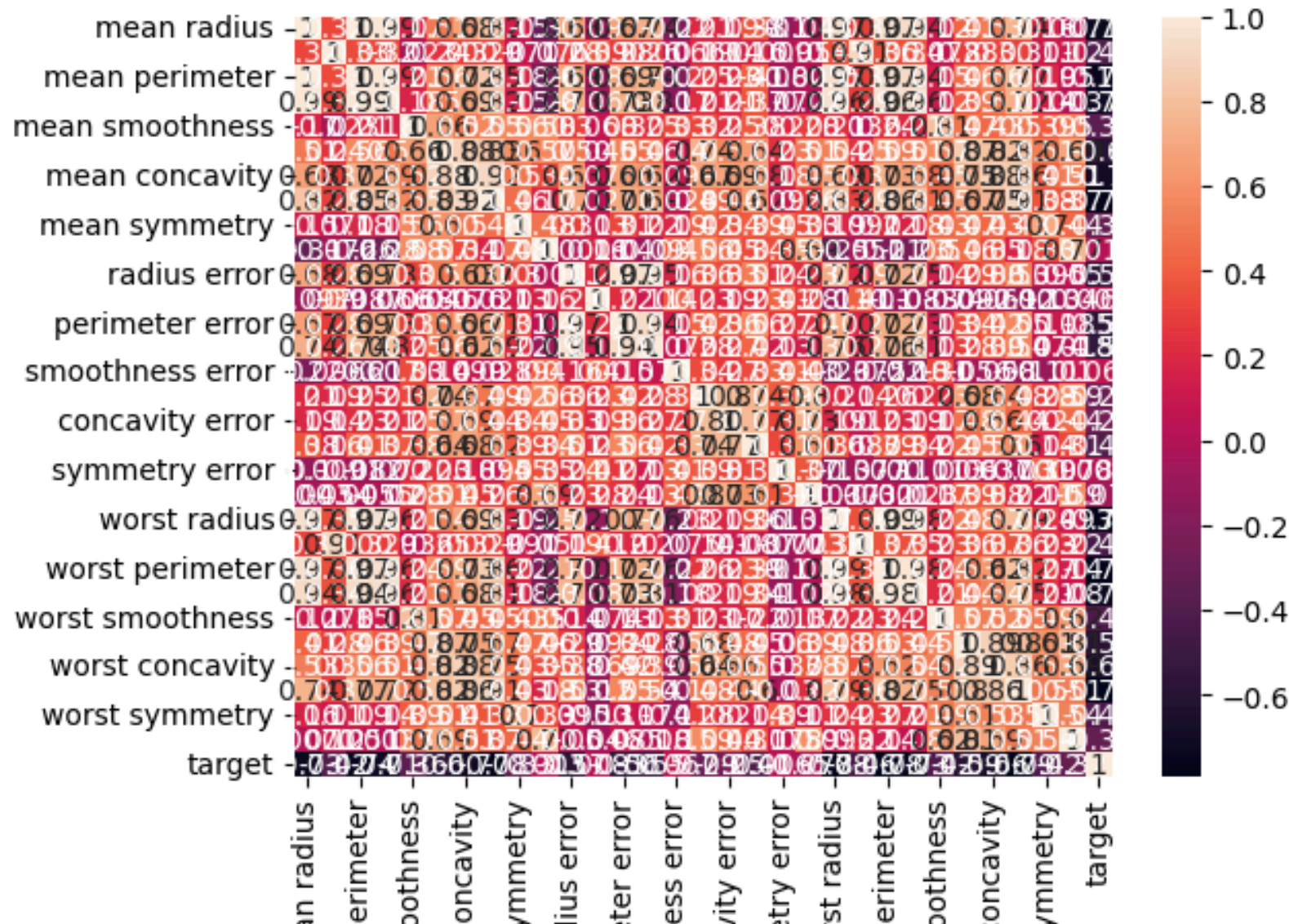
27 worst concave points      569 non-null      float64
28 worst symmetry            569 non-null      float64
29 worst fractal dimension    569 non-null      float64
30 target                    569 non-null      int64

```

dtypes: float64(30), int64(1)

memory usage: 137.9 KB

<Axes: >



Handle missing values

Encode categorical features if needed:

```
df = pd.get_dummies(df, drop_first=True)
```

Feature Selection

```
X = df.drop('target', axis=1)
y = df['target']
```

Train-Test Split

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

Feature Scaling

```
scaler = StandardScaler()  
X_train = scaler.fit_transform(X_train)  
X_test = scaler.transform(X_test)
```

Model Training

```
from sklearn.linear_model import LogisticRegression  
from sklearn.metrics import confusion_matrix, classification_report, accuracy_score
```

```
model = LogisticRegression(max_iter=1000)  
model.fit(X_train, y_train)
```



▼ LogisticRegression ⓘ ?
LogisticRegression(max_iter=1000)

```
y_pred = model.predict(X_test)
```

Model Evaluation

```
print("Confusion Matrix:\n", confusion_matrix(y_test, y_pred))  
print("\nClassification Report:\n", classification_report(y_test, y_pred))  
print("Accuracy Score:", accuracy_score(y_test, y_pred))
```

⇒ Confusion Matrix:

```
[[41  2]  
 [ 1 70]]
```

Classification Report:

	precision	recall	f1-score	support
0	0.98	0.95	0.96	43
1	0.97	0.99	0.98	71
accuracy			0.97	114
macro avg	0.97	0.97	0.97	114
weighted avg	0.97	0.97	0.97	114

Accuracy Score: 0.9736842105263158