

In [35]:

```
pip install scikit-learn pandas
```

Requirement already satisfied: scikit-learn in c:\users\h s computer s hyd\anaconda3\lib\site-packages (0.24.2)
Requirement already satisfied: pandas in c:\users\h s computer s hyd\anaconda3\lib\site-packages (1.3.4)
Requirement already satisfied: numpy>=1.13.3 in c:\users\h s computer s hyd\anaconda3\lib\site-packages (from scikit-learn) (1.22.4)
Requirement already satisfied: joblib>=0.11 in c:\users\h s computer s hyd\anaconda3\lib\site-packages (from scikit-learn) (1.1.0)
Requirement already satisfied: scipy>=0.19.1 in c:\users\h s computer s hyd\anaconda3\lib\site-packages (from scikit-learn) (1.7.1)
Requirement already satisfied: threadpoolctl>=2.0.0 in c:\users\h s computer s hyd\anaconda3\lib\site-packages (from scikit-learn) (2.2.0)
Requirement already satisfied: python-dateutil>=2.7.3 in c:\users\h s computer s hyd\anaconda3\lib\site-packages (from pandas) (2.8.2)
Requirement already satisfied: pytz>=2017.3 in c:\users\h s computer s hyd\anaconda3\lib\site-packages (from pandas) (2021.3)
Requirement already satisfied: six>=1.5 in c:\users\h s computer s hyd\anaconda3\lib\site-packages (from python-dateutil>=2.7.3->pandas) (1.16.0)
Note: you may need to restart the kernel to use updated packages.

In [40]:

```
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.ensemble import RandomForestClassifier
from sklearn.metrics import accuracy_score, classification_report
```

Load the Titanic dataset

In [77]:

```
# Load the dataset
titanic_data_path = "titanic.csv"
titanic_data = pd.read_csv(titanic_data_path)
```

In [76]:

```
# Display the first few rows of the dataset
print(titanic_data.head())
```

	Survived	Pclass	Sex	Age	SibSp	Parch	Fare
0	0	3	NaN	22.0	1	0	7.2500
1	1	1	NaN	38.0	1	0	71.2833
2	1	3	NaN	26.0	0	0	7.9250

3	1	1	NaN	35.0	1	0	53.1000
4	0	3	NaN	35.0	0	0	8.0500

Data Cleaning

```
In [78]: # For simplicity, we'll drop some columns and handle missing values
titanic_data = titanic_data.drop(['Name', 'Ticket', 'Cabin', 'Embarked', 'PassengerId'], axis=1)
titanic_data = titanic_data.dropna()
```

Data Preprocessing

```
In [79]: # Convert 'Sex' feature to numerical
titanic_data['Sex'] = titanic_data['Sex'].map({'female': 0, 'male': 1})
```

Split the data into Training and Testing sets

```
In [80]: from sklearn.model_selection import train_test_split

# Define features (X) and target variable (y)
X = titanic_data.drop('Survived', axis=1)
y = titanic_data['Survived']

# Split the data into training and testing sets
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)
```

Build a Random Forest Classifier

```
In [81]: from sklearn.ensemble import RandomForestClassifier
```

```
In [82]: # Build a Random Forest Classifier
model = RandomForestClassifier(n_estimators=100, random_state=42)
model.fit(X_train, y_train)
```

```
Out[82]: RandomForestClassifier(random_state=42)
```

Make predictions on the test set

```
In [83]: # Make predictions on the test set
y_pred = model.predict(X_test)
```

Evaluate the model

```
In [84]: from sklearn.metrics import accuracy_score, classification_report

# Evaluate the model
accuracy = accuracy_score(y_test, y_pred)
print(f"Accuracy: {accuracy:.2f}")

# Display classification report
print("Classification Report:")
print(classification_report(y_test, y_pred))
```

Accuracy: 0.80

Classification Report:

	precision	recall	f1-score	support
0	0.83	0.84	0.83	87
1	0.75	0.73	0.74	56
accuracy			0.80	143
macro avg	0.79	0.79	0.79	143
weighted avg	0.80	0.80	0.80	143