In [ ]:

```
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
from sklearn.model_selection import train_test_split
from sklearn.ensemble import RandomForestClassifier
from sklearn.metrics import classification_report
```

In [65]:

```
# Load the dataset
data = pd.read_csv('titanic.csv')
```

In [66]:

```
# Check the column names
print(data.columns)
```

```
Index(['PassengerId', 'Survived', 'Pclass', 'Name', 'Sex', 'Age', 'SibSp',
       'Parch', 'Ticket', 'Fare', 'Cabin', 'Embarked'],
      dtype='object')
```

In [67]:

```
# Handle missing values
data['Age'] = data['Age'].fillna(data['Age'].median())
```

In [68]:

```
# Strip any whitespace from column names
data.columns = data.columns.str.strip()
```

In [69]:

```
# Visualizations
plt.figure(figsize=(12, 6))
```

Out[69]:

```
<Figure size 1200x600 with 0 Axes>
```

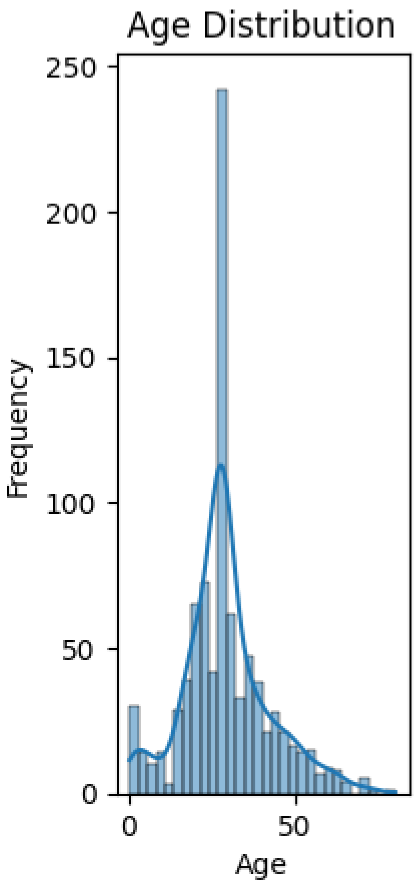
```
<Figure size 1200x600 with 0 Axes>
```

In [70]:

```
# 1. Distribution of Ages
plt.subplot(1, 3, 1)
sns.histplot(data['Age'], bins=30, kde=True)
plt.title('Age Distribution')
plt.xlabel('Age')
plt.ylabel('Frequency')
```

Out[70]:

```
Text(0, 0.5, 'Frequency')
```

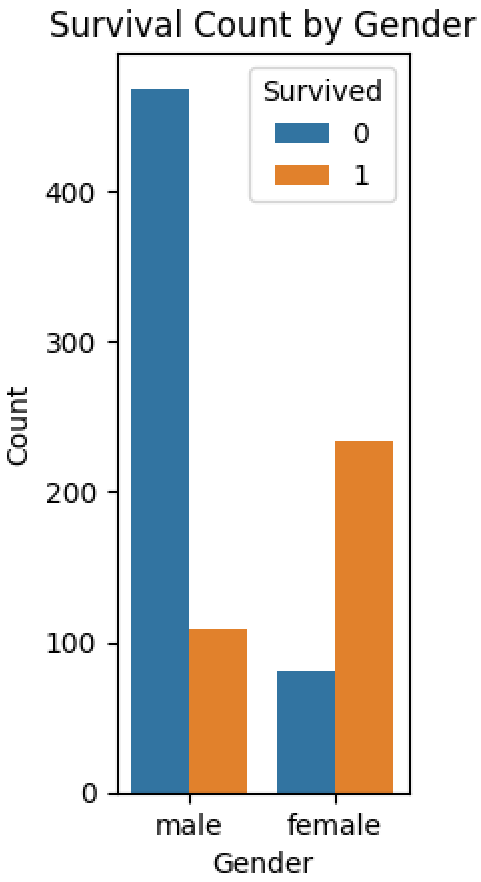


In [71]:

```
# 2. Survival Count by Gender
plt.subplot(1, 3, 2)
sns.countplot(x='Sex', hue='Survived', data=data)
plt.title('Survival Count by Gender')
plt.xlabel('Gender')
plt.ylabel('Count')
```

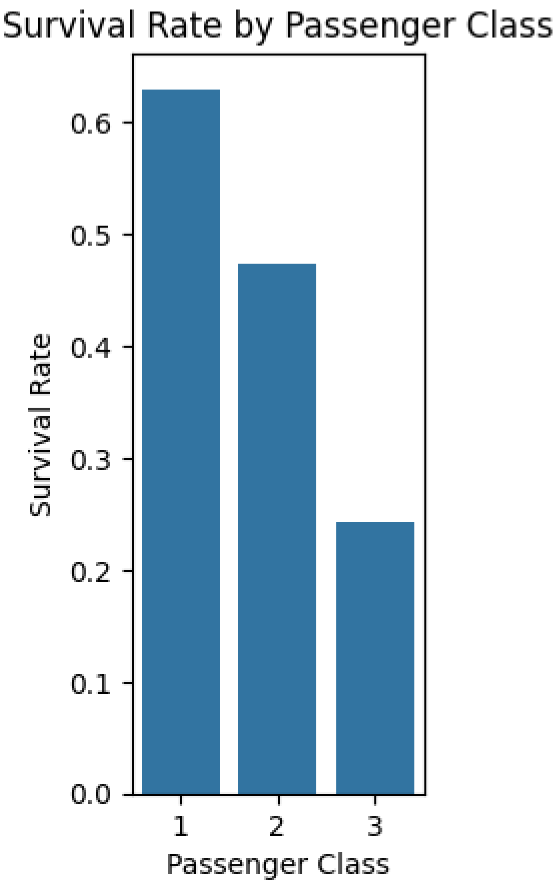
Out[71]:

```
Text(0, 0.5, 'Count')
```



```
In [72]: # 3. Survival Rate by Passenger Class
         plt.subplot(1, 3, 3)
         sns.barplot(x='Pclass', y='Survived', data=data, errorbar=None)  # Updated paramete
         plt.title('Survival Rate by Passenger Class')
         plt.xlabel('Passenger Class')
         plt.ylabel('Survival Rate')
```

```
Out[72]: Text(0, 0.5, 'Survival Rate')
```



In [73]:
```
plt.tight_layout()
plt.show()
```

```
<Figure size 640x480 with 0 Axes>
```

In [74]:
```
# Encode categorical variables
data = pd.get_dummies(data, columns=['Sex', 'Embarked'], drop_first=True)
```

In [75]:
```
# Select features and target variable
X = data[['Pclass', 'Age', 'SibSp', 'Parch', 'Fare', 'Sex_male', 'Embarked_Q', 'Emb
y = data['Survived']
```

In [76]:
```
# Split the data
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_sta
```

In [77]:
```
# Train a Random Forest Classifier
model = RandomForestClassifier()
model.fit(X_train, y_train)
```

Out[77]:
```
▾ RandomForestClassifier ⓘ ?
RandomForestClassifier()
```

In [78]:
```
# Make predictions
y_pred = model.predict(X_test)
```

In [79]:

```
# Evaluate the model
print(classification_report(y_test, y_pred))
```

```
              precision    recall  f1-score   support

           0       0.83      0.87      0.85       105
           1       0.80      0.76      0.78        74

    accuracy                           0.82       179
   macro avg       0.82      0.81      0.81       179
weighted avg       0.82      0.82      0.82       179
```

In [ ]:

```

```