```
from google.colab import drive
drive.mount('/content/drive')
```

```
Mounted at /content/drive
```

```
import pandas as pd
df = pd.read_csv('/content/drive/MyDrive/gender_submission.csv')
df.head()
```

	PassengerId	Survived
0	892	0
1	893	1
2	894	0
3	895	0
4	896	1

```
df.tail()
```

	PassengerId	Survived
413	1305	0
414	1306	1
415	1307	0
416	1308	0
417	1309	0

```
df.isnull().sum()
```

	0
PassengerId	0
Survived	0

dtype: int64

```
x = df.iloc[:, :-1].values
y = df.iloc[:, -1].values
```

```
import numpy as np
```

```
x = np.array(x).reshape(-1,1)
y = np.array(y).reshape(-1,1)
```

```
from sklearn.model_selection import train_test_split
```

```
from sklearn.svm import SVC
```

```
svm_linear = SVC(kernel='linear', class_weight = 'balanced')
```

```
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 = 0)
svm_linear.fit(x_train, y_train.ravel())
y_pred = svm_linear.predict(x_test)
```

```
from sklearn.metrics import accuracy_score
```

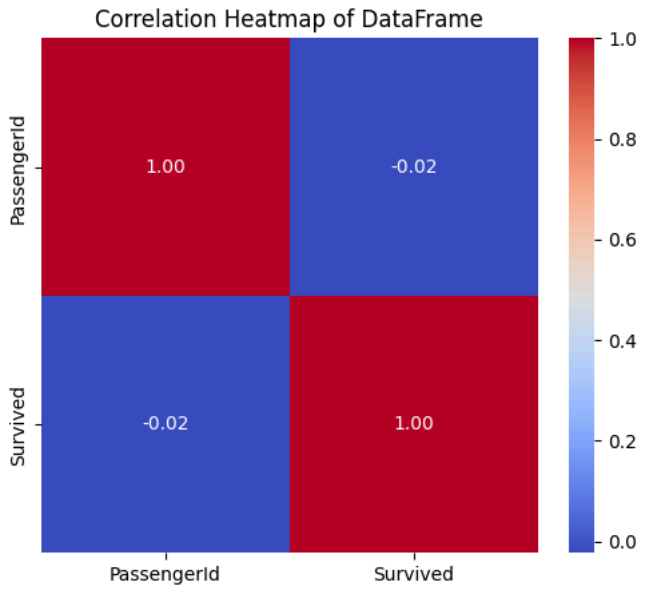
```
accuracy = accuracy_score(y_test, y_pred)
print(f"Accuracy: {accuracy:.2f}")
```

```
Accuracy: 0.39
```

```
import seaborn as sns
import matplotlib.pyplot as plt
```

```
# Calculate the correlation matrix
correlation_matrix = df.corr()

# Create a heatmap
plt.figure(figsize=(6, 5))
sns.heatmap(correlation_matrix, annot=True, cmap='coolwarm', fmt=".2f")
plt.title('Correlation Heatmap of DataFrame')
plt.show()
```




```
print("Training samples:", x_train.shape[0])
print("Testing samples:", x_test.shape[0])
```

```
Training samples: 334
Testing samples: 84
```

```
from sklearn.svm import SVC
```

```
svm_linear = SVC(kernel='rbf', class_weight='balanced', C=0.1)
```

```
svm_linear.fit(x_train, y_train)
```

```
/usr/local/lib/python3.12/dist-packages/sklearn/utils/validation.py:1408: DataConversionWarning: A column-vector y was passe
  y = column_or_1d(y, warn=True)
```

SVC

```
SVC(C=0.1, class_weight='balanced')
```

```
from sklearn.metrics import precision_score

precision = precision_score(y_test, y_pred, average='binary') # 'binary' is used for binary classification
print(f"Precision: {precision:.2f}")
```

```
Precision: 0.35
```

```
from sklearn.metrics import f1_score

f1 = f1_score(y_test, y_pred, average='binary')
print(f"F1 Score: {f1:.2f}")
```

```
F1 Score: 0.35
```

```
confusion_matrix(y_test, y_pred)
```

```
array([[19, 26],
       [25, 14]])
```