

1.

```
x = df[["Sex", "Age", "SibSp", "Parch", "Embarked_C", "Embarked_Q", "Embarked_S", "Pclass_1", "Pclass_2", "Pclass_3"]]
y = df[["Survived"]]
```

2.

```
clf = LogisticRegression().fit(X_train, Y_train)
Y_pred = clf.predict(X_test)
print("Accuracy:", accuracy_score(Y_test, Y_pred)* 100, "%")
```

✓ 0.0s

Accuracy: 81.111111111111 %

3.

```
print("Accuracy:", accuracy_score(Y_test, Y_pred)* 100, "%")
print("Theta (Coefficients):", clf.coef_)
print("Intercept (Theta_0):", clf.intercept_)
```

✓ 0.0s

Accuracy: 81.111111111111 %

Theta (Coefficients): [[-2.53048465 -0.03983331 -0.29063977 -0.03670257 0.16300861 0.09197773
-0.28380321 0.96907599 0.13748739 -1.13538024]]

Intercept (Theta_0): [2.86065169]

4.

```
test_data = {
    'PassengerId': [2024, 2025, 2026],
    'Pclass': [1, 2, 3],
    'Name': ['Tony Stark', 'Purple Thanos', 'Pink Kirby'],
    'Sex': ['male', 'male', 'female'],
    'Age': [50, 40, 30],
    'SibSp': [1, 1, 0],
    'Parch': [2, 1, 2],
    'Ticket': ['A/5 21172', 'PC 17600', 'STON/O2. 3101283'],
    'Fare': [100, 50, 25],
    'Cabin': ['C04', 'C03', 'C02'],
    'Embarked': ['S', 'C', 'Q']
}

test_df = pd.DataFrame(test_data)

# Preprocessing
# Encoding categorical data
label_encoder = LabelEncoder()
test_df['Sex'] = label_encoder.fit_transform(test_df['Sex']) # Convert 'Sex' to numerical
test_df['Embarked'].fillna(test_df['Embarked'].value_counts().idxmax(), inplace=True)
test_df['Age'].fillna(test_df['Age'].median(skipna=True), inplace=True)
print(test_df.isnull().sum())
# df['Embarked'].fillna('S', inplace=True) # Fill missing 'Embarked' with 'S' for Southampton
# df['Embarked'] = label_encoder.fit_transform(df['Embarked']) # Convert 'Embarked' to numerical
test_df = pd.get_dummies(test_df, columns=['Embarked', 'Pclass'])
# Convert True/False values to 0/1
test_df['Embarked_C'] = test_df['Embarked_C'].astype(int)
test_df['Embarked_Q'] = test_df['Embarked_Q'].astype(int)
test_df['Embarked_S'] = test_df['Embarked_S'].astype(int)
# Convert True/False values to 0/1 for Pclass
test_df['Pclass_1'] = test_df['Pclass_1'].astype(int)
test_df['Pclass_2'] = test_df['Pclass_2'].astype(int)
test_df['Pclass_3'] = test_df['Pclass_3'].astype(int)

test_df.drop(['Cabin', 'PassengerId', 'Name', 'Ticket', 'Fare'], axis=1, inplace=True)

# Handling missing values
# df['Age'].fillna(df['Age'].median(), inplace=True) # Replace missing 'Age' with median value
# Displaying predictions in a readable format
test_prediction = clf.predict(test_df)

for i, x in enumerate(test_prediction):
    print(f"The person number {i+1} is predicted to {'survive' if x == 1 else 'not survive'}")
```

✓ 0.0s

Python

```

PassengerId    0
Pclass         0
Name           0
Sex            0
Age           0
SibSp         0
Parch         0
Ticket        0
Fare          0
Cabin         0
Embarked      0
dtype: int64
The person number 1 is predicted to not survive
The person number 2 is predicted to not survive
The person number 3 is predicted to survive

```

5.

```

> x = df[["Sex","Age","SibSp","Parch","Embarked_C","Embarked_Q","Embarked_S","Pclass_1","Pclass_2","Pclass_3"]]
> y = df[["Survived"]]
> X_train, X_test, Y_train, Y_test = train_test_split(x, y, test_size= 0.1, train_size=0.1, random_state=0)
318] ✓ 0.0s

clf = LogisticRegression().fit(X_train, Y_train)
Y_pred = clf.predict(X_test)
print("Accuracy:", accuracy_score(Y_test, Y_pred)* 100, "%")
print("Theta (Coefficients):", clf.coef_)
print("Intercept (Theta_0):", clf.intercept_)
319] ✓ 0.0s

... Accuracy: 77.7777777777779 %

> x = df[["Sex","Age","SibSp","Parch","Embarked_C","Embarked_Q","Embarked_S","Pclass_1","Pclass_2","Pclass_3"]]
> y = df[["Survived"]]
> X_train, X_test, Y_train, Y_test = train_test_split(x, y, test_size= 0.2, train_size=0.2, random_state=0)
27] ✓ 0.0s

clf = LogisticRegression().fit(X_train, Y_train)
Y_pred = clf.predict(X_test)
print("Accuracy:", accuracy_score(Y_test, Y_pred)* 100, "%")
print("Theta (Coefficients):", clf.coef_)
print("Intercept (Theta_0):", clf.intercept_)
28] ✓ 0.0s

Accuracy: 83.24022346368714 %
Theta (Coefficients): [[-2.13962303 -0.05488838 -0.39017978  0.03032586 -0.16967751  0.33282538
 -0.16581859  0.94376953  0.11850744 -1.06494768]]
Intercept (Theta_0): [3.16259223]

```

```

x = df[["Sex","Age","SibSp","Parch","Embarked_C","Embarked_Q","Embarked_S","Pclass_1","Pclass_2","Pclass_3"]]
y = df[["Survived"]]
X_train, X_test, Y_train, Y_test = train_test_split(x, y, test_size= 0.3, train_size=0.3, random_state=0)

clf = LogisticRegression().fit(X_train, Y_train)
Y_pred = clf.predict(X_test)
print("Accuracy:", accuracy_score(Y_test, Y_pred)* 100, "%")
print("Theta (Coefficients):", clf.coef_)
print("Intercept (Theta_0):", clf.intercept_)

Accuracy: 81.71641791044776 %
```

```

x = df[["Sex","Age","SibSp","Parch","Embarked_C","Embarked_Q","Embarked_S","Pclass_1","Pclass_2","Pclass_3"]]
y = df[["Survived"]]
X_train, X_test, Y_train, Y_test = train_test_split(x, y, test_size= 0.4, train_size=0.4, random_state=0)

clf = LogisticRegression().fit(X_train, Y_train)
Y_pred = clf.predict(X_test)
print("Accuracy:", accuracy_score(Y_test, Y_pred)* 100, "%")
print("Theta (Coefficients):", clf.coef_)
print("Intercept (Theta_0):", clf.intercept_)

Accuracy: 80.95238095238095 %
Theta (Coefficients): [[-2.2793064 -0.03091771 -0.33987795  0.04028949  0.34653232 -0.0442204
 -0.37333286  0.7525746  0.11329132 -0.93688686]]
Intercept (Theta_0): [2.26589526]
```

```

x = df[["Sex","Age","SibSp","Parch","Embarked_C","Embarked_Q","Embarked_S","Pclass_1","Pclass_2","Pclass_3"]]
y = df[["Survived"]]
X_train, X_test, Y_train, Y_test = train_test_split(x, y, test_size= 0.5, train_size=0.5, random_state=0)

clf = LogisticRegression().fit(X_train, Y_train)
Y_pred = clf.predict(X_test)
print("Accuracy:", accuracy_score(Y_test, Y_pred)* 100, "%")
print("Theta (Coefficients):", clf.coef_)
print("Intercept (Theta_0):", clf.intercept_)

Accuracy: 78.9237668161435 %
```