

1. Select the features you intend to use as independent variables and identify your target (dependent) variable. Split the data into training and testing sets. Create a logistic regression classifier and fit the model.

```
# Independent Variables
X = df[['Pclass', 'Sex', 'Age', 'SibSp', 'Parch', 'Fare', 'Embarked_C',
'Embarked_Q', 'Embarked_S']]
# Dependent Variable
y = df['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)

# Create a logistic regression classifier
clf = LogisticRegression()

# Fit the model
clf.fit(X_train, y_train)
```

2. Utilize your model to make predictions on the testing data, calculate evaluation metrics such as accuracy and recall, and print the results.

```
# Make predictions
y_pred = clf.predict(X_test)

# Evaluate the model
## Accuracy
accuracy = accuracy_score(y_test, y_pred)
print('Accuracy:', accuracy)

## Recall
recall = recall_score(y_test, y_pred)
print('Recall:', recall)
```

Accuracy: 0.8044692737430168

Recall: 0.7297297297297297

3. Display the theta parameter values.

```
# DisplayTheta parameters values
theta = clf.coef_
print('Theta:', theta)
```

```
Theta: [[-0.81348658 -2.49646632 -0.02524414 -0.24483755 -0.12245511  0.00323061
         1.23133623  0.68274734  0.64198736]]
```

4. Create a DataFrame with 3 records (for 3 persons), use your model to make predictions, and print the predicted results using text descriptions such as 'survived' and 'not survived'.

```
# Creating a DataFrame with 3 records for prediction
new_data = pd.DataFrame({
    'Pclass': [3, 1, 2],
    'Sex': [1, 0, 0], # 1 for male, 0 for female
    'Age': [22, 38, 26],
    'SibSp': [1, 1, 0],
    'Parch': [0, 0, 0],
    'Fare': [7.25, 71.2833, 7.925],
    'Embarked_C': [0, 1, 0],
    'Embarked_Q': [0, 0, 0],
    'Embarked_S': [1, 0, 1]
})

# Making predictions on the new data
predictions = clf.predict(new_data)

# Interpreting the predictions
predicted_results = ['survived' if pred == 1 else 'not survived' for pred in
predictions]

# Printing the results
for i, result in enumerate(predicted_results):
    print(f'Person {i+1}: {result}')
```

```
Person 1: not survived
Person 2: survived
```

```
Person 3: survived
```

5. Alter the training/testing split fraction and the maximum iteration of the logistic regression model, observe and print the different outcomes.

Split ratio result

	Split Ratio float64	Accuracy float64	Recall float64	Theta object
0	0.1	0.84444444444	0.83333333333	[[-9.49156527e-0...
1	0.2	0.8044692737	0.7297297297	[[-0.81348658 -2...
2	0.3	0.8134328358	0.7297297297	[[-0.87169114 -2....
3	0.4	0.7955182073	0.6808510638	[[-0.85231032 -2....
4	0.5	0.8071748879	0.7094972067	[[-8.58604974e-...

Max iteration result

	Max Iteration int64	Accuracy float64	Recall float64	Theta object
0	100	0.8044692737	0.7297297297	[[-0.81348658 -2...
1	200	0.7932960894	0.7162162162	[[-0.8769898 -2....
2	300	0.8100558659	0.7432432432	[[-9.37088012e-...
3	400	0.8100558659	0.7432432432	[[-9.37088012e-...
4	500	0.8100558659	0.7432432432	[[-9.37088012e-...