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

A screenshot of a screenshot of a computer

AI-generated content may be incorrect.

Max iteration result

A screenshot of a computer

AI-generated content may be incorrect.