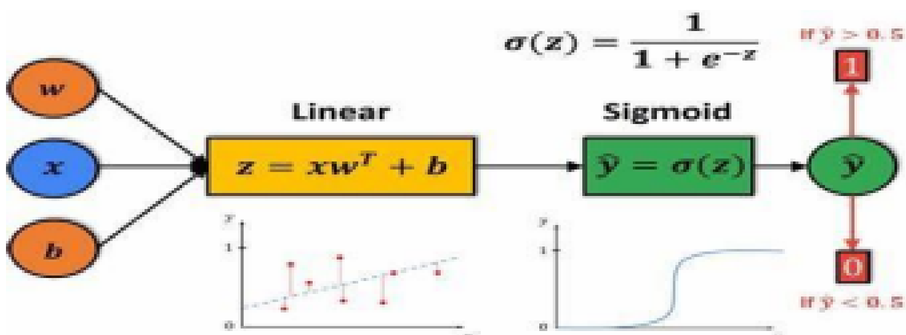
Experiment No.3

Aim- To study, understand and implement a logistic regression algorithm.

Theory

Logistic regression aims to solve classification problems. It does this by predicting categorical outcomes, unlike linear regression that predicts a continuous outcome. In the simplest case there are two outcomes, which is called binomial, an example of which is predicting if a tumor is malignant or benign. Other cases have more than two outcomes to classify; in this case it is called multinomial. A common example for multinomial logistic regression would be predicting the class of an iris flower between 3 different species.The logistic regression model transforms the linear regression function continuous value output into categorical value output using a sigmoid function, which maps any real-valued set of independent variables input into a value between 0 and 1. This function is known as the logistic function.



Logistic Function – Sigmoid Function

· The sigmoid function is a mathematical function used to map the predicted values to probabilities.

· It maps any real value into another value within a range of 0 and 1. The value of the logistic regression must be between 0 and 1, which cannot go beyond this limit, so it forms a curve like the "S" form.

· The S-form curve is called the Sigmoid function or the logistic function. · In logistic regression, we use the concept of the threshold value, which defines the probability of either 0 or 1. Such as values above the threshold value tends to 1, and a value below the threshold values tends to 0.

Types of Logistic Regression

On the basis of the categories, Logistic Regression can be classified into three types:

1. Binomial: In binomial Logistic regression, there can be only two possible types of the dependent variables, such as 0 or 1, Pass or Fail, etc.

2. Multinomial: In multinomial Logistic regression, there can be 3 or more possible unordered types of the dependent variable, such as “cat”, “dogs”, or “sheep”

Discussion-

· Logistic regression predicts the output of a categorical dependent variable. Therefore, the outcome must be a categorical or discrete value.

· It can be either Yes or No, 0 or 1, true or False, etc. but instead of giving the exact value as 0 and 1, it gives the probabilistic values which lie between 0 and 1. · In Logistic regression, instead of fitting a regression line, we fit an “S” shaped logistic function, which predicts two maximum values (0 or 1).

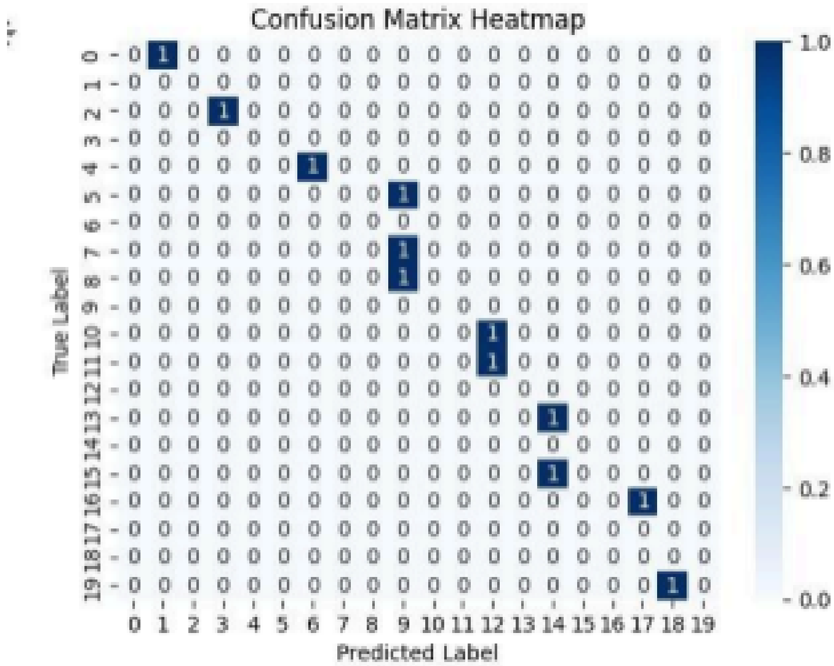
Program :

```
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn import linear_model, metrics
from sklearn.metrics import confusion_matrix
import seaborn as sns
import matplotlib.pyplot as plt
# Load your dataset
# Replace 'your_dataset.csv' with the path to your CSV
file df = pd.read_csv('/content/home (1).csv')
# Assume 'target' is the column with labels and the rest are
features X = df.drop(columns='Area_Square_Feet')
y = df['Area_Square_Feet']
# Split X and y into training and testing sets
X_train, X_test, y_train, y_test = train_test_split(X, y,
test_size=0.3, random_state=1)
# Create logistic regression object
reg = linear_model.LogisticRegression(max_iter=200)
# Train the model using the training sets
reg.fit(X_train, y_train)
# Make predictions on the testing set
y_pred = reg.predict(X_test)
# Compare actual response values (y_test) with predicted response
values (y_pred)
accuracy = metrics.accuracy_score(y_test, y_pred) * 100
```

```
print("Logistic Regression model accuracy (in %):",
accuracy) # Print confusion matrix
cf_matrix = confusion_matrix(y_test, y_pred)
print("Confusion Matrix:\n", cf_matrix)
# Print confusion matrix using Seaborn library
sns.heatmap(cf_matrix, annot=True, fmt='d',
cmap='Blues') plt.xlabel('Predicted Label')
plt.ylabel('True Label')
plt.title('Confusion Matrix Heatmap')
plt.show()
```

OUTPUT :-

```
Logistic Regression model accuracy (in %):
90.52294853963839 Confusion Matrix:
 [[0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0]
 [0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0]
 [0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0]
 [0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0]
 [0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0]
 [0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0]
 [0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0]
 [0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0]
 [0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0]
 [0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0]
 [0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0]
 [0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0]
 [0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0]
 [0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0]
 [0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0]
 [0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0]
 [0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0]
 [0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0]
 [0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0]
 [0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0]]
```



Conclusion

Logistic regression effectively transforms continuous input into categorical outcomes using the sigmoid function, making it suitable for classification tasks with binary or multinomial outcomes. It provides probabilistic predictions that are constrained between 0 and 1, facilitating decision-making in various classification problems.