Ex No. 11	Implementation of Logistic Regression to Classify Problems
Date:	

Aim

To implement Logistic Regression algorithm to classify problems such as Diabetes prediction and Spam detection.

Data Sets

1. diabetes.csv (https://www.kaggle.com/datasets/saurabh00007/diabetescsv)
2. SMSSpamCollection.csv (https://archive.ics.uci.edu/ml/machine-learning-databases/00228/)

Definition

Logistic Regression

Logistic regression estimates the probability of an event occurring, such as voted or didn't vote, based on a given dataset of independent variables. Since the outcome is a probability, the dependent variable is bounded between 0 and 1.

Procedure

Open PyCharm Community Edition.

Go to File menu → New Project → Specify the project name → Press "Create" button.

Right Click on Project name → New → Python File → Specify the file name → Press Enter.

Type the following codes. Right click on file name or coding window → Select "Run" to view the result.

Diabetes Prediction:

Logisticdia.py

```
import pandas as pd
from sklearn.linear_model import LogisticRegression
from sklearn.model_selection import train_test_split
from sklearn import metrics

data = pd.read_csv("C:/Users/2mca2/Downloads/diabetes.csv")

print(data.head)

print(data.dtypes)

print(data.describe())

X = data.drop("Outcome", axis=1)
Y = data[["Outcome"]]

X_train, X_test, Y_train, Y_test = train_test_split(X, Y, test_size=0.30, random_state=7)

model = LogisticRegression()
model.fit(X_train, Y_train)
Y_predict = model.predict(X_test)
model_score = model.score(X_test, Y_test)

print(model_score)
print(metrics.confusion_matrix(Y_test, Y_predict))
```

Output

```
C:\Users\2mca2\PycharmProjects\sumaiya\venv\Scripts\python.exe
C:/Users/2mca2/PycharmProjects/sumaiya/logisticdia.py

<bound method NDFrame.head of      Pregnancies  Glucose  ...  Age  Outcome

0              6      148  ...   50        1

1              1       85  ...   31        0
```

```
2              8      183  ...   32        1
3              1       89  ...   21        0
4              0      137  ...   33        1
..           ...      ...  ...  ...      ...
763           10      101  ...   63        0
764            2      122  ...   27        0
765            5      121  ...   30        0
766            1      126  ...   47        1
767            1       93  ...   23        0

[768 rows x 9 columns]>
Pregnancies                 int64
Glucose                     int64
BloodPressure               int64
SkinThickness               int64
Insulin                     int64
BMI                       float64
DiabetesPedigreeFunction  float64
Age                         int64
Outcome                     int64
dtype: object
       Pregnancies     Glucose  ...         Age     Outcome
count   768.000000  768.000000  ...  768.000000  768.000000
mean      3.845052  120.894531  ...   33.240885    0.348958
std       3.369578   31.972618  ...   11.760232    0.476951
min       0.000000    0.000000  ...   21.000000    0.000000
25%       1.000000   99.000000  ...   24.000000    0.000000
```

50% 3.000000 117.000000 ... 29.000000 0.000000

75% 6.000000 140.250000 ... 41.000000 1.000000

max 17.000000 199.000000 ... 81.000000 1.000000

[8 rows x 9 columns]

0.7489177489177489

[[127 20]

 [38 46]]

Process finished with exit code 0

Spam Detection:

Logisticspam.py

```
import pandas as pd
import numpy as np
from sklearn.feature_extraction.text import TfidfVectorizer
from sklearn.linear_model import LogisticRegression
from sklearn.model_selection import train_test_split, cross_val_score

df = pd.read_csv("C:/Users/2mca2/Downloads/SMSSpamCollection.csv", delimiter='\t', header=None)

print(df.describe)
print(df.dtypes)
print(df.head)
print(df.shape)
X_train_raw, X_test_raw, y_train, y_test = train_test_split(df[1],df[0])

vectorizer = TfidfVectorizer()
X_train = vectorizer.fit_transform(X_train_raw)
classifier = LogisticRegression()
classifier.fit(X_train, y_train)

X_test = vectorizer.transform( ['URGENT! Your Mobile No 1234 was awarded a Prize', 'Hey honey,
whats up?'] )
predictions = classifier.predict(X_test)
print("Result:")
print(predictions)
```

Output

C:\Users\2mca2\PycharmProjects\sumaiya\venv\Scripts\python.exe
C:/Users/2mca2/PycharmProjects/sumaiya/Logisticspam.py

<bound method NDFrame.describe of 0 1

0 ham Go until jurong point, crazy.. Available only ...

1 ham Ok lar... Joking wif u oni...

2 spam Free entry in 2 a wkly comp to win FA Cup fina...

3 ham U dun say so early hor... U c already then say...

4 ham Nah I don't think he goes to usf, he lives aro...

... ...

5567 spam This is the 2nd time we have tried 2 contact u...

5568 ham Will ü b going to esplanade fr home?

5569 ham Pity, * was in mood for that. So...any other s...

5570 ham The guy did some bitching but I acted like i'd...

5571 ham Rofl. Its true to its name

[5572 rows x 2 columns]>

0 object

1 object

dtype: object

<bound method NDFrame.head of 0 1

0 ham Go until jurong point, crazy.. Available only ...

1 ham Ok lar... Joking wif u oni...

2 spam Free entry in 2 a wkly comp to win FA Cup fina...

3 ham U dun say so early hor... U c already then say...

4 ham Nah I don't think he goes to usf, he lives aro...

... ...

5567 spam This is the 2nd time we have tried 2 contact u...

5568 ham Will ü b going to esplanade fr home?

5569 ham Pity, * was in mood for that. So...any other s...

5570 ham The guy did some bitching but I acted like i'd...

5571 ham   Rofl. Its true to its name

[5572 rows x 2 columns]>

(5572, 2)

Result:

['spam' 'ham']

Process finished with exit code 0

Result

Thus, Logistic Regression algorithm for Diabetes Prediction and Spam Detection has been implemented successfully.