|  |  |
| --- | --- |
| **TITLE** | **Data Analytics II** |
| **PROBLEM STATEMENT/ DEFINITION** | 1. Implement logistic regression using Python /R to perform classification on Social\_Network\_Ads.csv dataset. 2. ComputeConfusionmatrixtofindTP,FP,TN,FN,Accuracy, Errorrate, Precision,Recall on the given dataset. |
| **OBJECTIVE** | To understand how logistic regression works on the given dataset. |
| **OUTCOME** | To find the best scenario for the result to be achieved for a given data set using logistic regression. |
| **S/W PACKAGES AND HARDWARE APPARATUS USED** | Core 2 DUO/i3/i5/i7 64-bit processor  OS-LINUX 64 bit OS  Editor-gedit/Eclipse  S/w- Jupyter Notebook/ Weka/ Python |
| **REFERENCES** | 1. Chirag Shah, “A Hands-On Introduction To Data Science”, Cambridge University Press, (2020), ISBN : ISBN 978-1-108-47244-9. Curriculum for Third Year of Computer Engineering (2019 Course), Savitribai Phule Pune University http://collegecirculars.unipune.ac.in/sites/documents/Syllabus2020/Forms/AllItems.aspx #57/87 2. Giuseppe Bonaccorso, “ Machine Learning   Algorithms”, Packt Publishing Limited, ISBN-10:  1785889621, ISBN-13: 978-1785889622 |
| **STEPS** | **Refer to student activity flow chart if found necessary by subject teacher and relevant to the subjectmanual.**  **Describe steps only.** |
| **INSTRUCTIONS FOR WRITING JOURNAL** | 1. title 2. Problem statement 3. Learning objective 4. Learning outcome 5. Theory (includes methods, libraries and functions, 6. Analysis (as per assignment), 7. conclusion. |

P:F:-LTL-UG / 03 /R1

**TITLE- Data Analytics II**

**PROBLEM STATEMENT/ DEFINITION-**Implement logistic regression using Python /R to perform classification on Social\_Network\_Ads.csv dataset.

ComputeConfusionmatrixtofindTP,FP,TN,FN,Accuracy,Errorrate,Precision,

Recall on the given dataset.

**LEARNING OBJECTIVE-**

To understand how logistic regression works on the given dataset.

**LEARNING OUTCOME**

To find the best scenario for the result to be achieved for a given data set using logistic regression

**THEORY-** Logistic regression is a classification method which is based on the probability for

a sample to belong to a class. As our probabilities must be continuous in R and bounded

between (0, 1), it's necessary to introduce a threshold function to filter the term z. The name

logistic comes from the decision to use the sigmoid (or logistic) function: 

A solution for classification is logistic regression. Instead of fitting a straight line or hyperplane, the logistic regression model uses the logistic function to squeeze the output of a linear equation between 0 and 1.

Methods-

from sklearn.model\_selection import train\_test\_split

>>> X\_train, X\_test, Y\_train, Y\_test = train\_test\_split(X, Y,test\_size=0.25)

Now we can train the model using the default parameters:

from sklearn.linear\_model import LogisticRegression

>>> lr = LogisticRegression()

>>> lr.fit(X\_train, Y\_train)

LogisticRegression(C=1.0, class\_weight=None, dual=False, fit\_intercept=True,

intercept\_scaling=1, max\_iter=100, multi\_class='ovr', n\_jobs=1,

penalty='l2', random\_state=None, solver='liblinear', tol=0.0001,verbose=0, warm\_start=False)

>>> lr.score(X\_test, Y\_test)

0.95199999999999996

It's also possible to check the quality through a cross-validation (like for linear regression):

from sklearn.model\_selection import cross\_val\_score

>>> cross\_val\_score(lr, X, Y, scoring='accuracy', cv=10)

array([ 0.96078431, 0.92156863, 0.96 , 0.98 , 0.96 ,

0.98 , 0.96 , 0.96 , 0.91836735, 0.97959184])

**Classification metrics**

A classification task can be evaluated in many different ways to achieve specific objectives.

Of course, the most important metric is the accuracy, often expressed as:



In scikit-learn, it can be assessed using the built-in accuracy\_score() function:

from sklearn.metrics import accuracy\_score

>>> accuracy\_score(Y\_test, lr.predict(X\_test))

Let us understand the confusion matrix. In many cases, it's necessary to be able to differentiate between different kinds of misclassifications (we're considering the binary case with the conventional notation: 0-negative, 1-positive), because the relative weight is quite different. For this reason, we introduce the following definitions:

**True positive: A positive sample correctly classified**

**False positive: A negative sample classified as positive**

**True negative: A negative sample correctly classified**

**False negative: A positive sample classified as negative**

Scikit learn supports the following method to compute the confusion matrix.

from sklearn.metrics import confusion\_matrix

>>> cm = confusion\_matrix(y\_true=Y\_test, y\_pred=lr.predict(X\_test))

cm[::-1, ::-1]

**CONCLUSION-** Thus, logistic regression model on the given data set is applied .The results shows of fitting a logistic regression model on the given dataset and shown the features used in the model, their estimated weights the standard errors of the estimated weights.