ASSIGNMENT-NO-5

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.

TITLE- Data Analytics II

PROBLEM STATEMENT/ DEFINITION-Implement logisticregression using Python /R to perform classification on Social_Network_Ads.csv dataset. ComputeConfusionmatrixtofind TP,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:

$$\sigma(z) = \frac{1}{1+e^{-z}} \; which\ becomes\ \sigma(\bar{x};\bar{w}) = \frac{1}{1+e^{-\bar{x}\cdot\bar{w}}}$$

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:

$$Generic\ accuracy = 1 - \frac{Number\ of\ misclassified\ samples}{Total\ number\ of\ samples}$$

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.

ASSIGNMENT-NO-6

TITLE	**Data Analytics III**
PROBLEM STATEMENT/ DEFINITION	Implement Simple Naïve Bayes classification algorithm using Python/R on iris.csv dataset. Compute Confusion matrix to find TP, FP, TN, FN, Accuracy, Error rate, Precision, Recall on the given dataset.
OBJECTIVE	To understand how Naïve Bayes classification algorithm 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	3. 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 4. 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.

ASSIGNMENT-NO-6

TITLE- Data Analytics III

PROBLEM STATEMENT/ DEFINITION-Implement logisticregression using Python /R to perform classification on Social_Network_Ads.csv dataset. ComputeConfusionmatrixtofind TP,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-
(includes methods, libraries and functions,

A naive Bayes classifier is called so because it's based on a naive condition, which implies the conditional independence of causes. This can seem very difficult to accept in many contexts where the probability of a particular feature is strictly correlated to another one.

For example, in spam filtering, a text shorter than 50 characters can increase the probability of the presence of an image, or if the domain has been already blacklisted for sending the same spam emails to million users, it's likely to find particular keywords.

Following three classification methods can be applied to the data set.

1. BernoulliNB()
2. GaussianNB()
3. MultinomialNB()

Bernoulli naive Bayes expects binary feature vectors; however, the class BernoulliNB has a binarize parameter, which allows us to specify a threshold that will be used internally to transform the features:

```
from sklearn.datasets import make_classification
>>> nb_samples = 300
>>> X, Y = make_classification(n_samples=nb_samples, n_features=2,
n_informative=2, n_redundant=0)

 from sklearn.naive_bayes import BernoulliNB
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)
```

```
>>> bnb = BernoulliNB(binarize=0.0)
>>> bnb.fit(X_train, Y_train)
>>> bnb.score(X_test, Y_test)

from sklearn.datasets import load_digits
from sklearn.model_selection import cross_val_score
>>> digits = load_digits()
>>> gnb = GaussianNB()
>>> mnb = MultinomialNB()
```
Analysis (as per assignment)

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:

$$Generic\ accuracy = 1 - \frac{Number\ of\ misclassified\ samples}{Total\ number\ of\ samples}$$

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 and calculating the precision and recall

```
from sklearn.metrics import confusion_matrix

>>> cm = confusion_matrix(y_true=Y_test, y_pred=lr.predict(X_test))
cm[::-1, ::-1]

from sklearn.metrics import precision_score

>>> precision_score(Y_test, lr.predict(X_test))

from sklearn.metrics import recall_score

>>> recall_score(Y_test, lr.predict(X_test))
```

CONCLUSION-Thus, Naïve Bay's Classifier model on the given data set is applied .The results shows the classification using various methods with precision and recall.