**50.How to build logistic regression model using sklearn in python?**

**Objective:**

* To build a logistic regression model for given data set using python.

**Process:**

* Import libraries.
* Read the sample data.
* Define X and y variables.
* Build the logistic regression model using sklearn.
* Split the sample data into training and test data.
* Fit the training data into the regression model.

**Input:**

* Data set(csv file).

**Output:**

* Summary of logistic regression model using sklearn.

**Source code:**

#import libraries

import pandas as pd

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

from sklearn.linear\_model import LogisticRegression

from sklearn import metrics

from sklearn.metrics import classification\_report

#read the data set

data=pd.read\_csv('/home/soft27/soft27/Sathish/

Pythonfiles/Employee.csv')

#creating data frame

df=pd.DataFrame(data)

print(df)

#assigning the independent variable

X = df[['rating','bonus']]

#assigning the dependent variable

y = df['logo']

#split data in training and test data

X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, test\_size=0.3, random\_state=0)

print(X\_train)

print(y\_train)

#build the model

logmodel = LogisticRegression()

logmodel.fit(X\_train,y\_train)

#predictions

predictions = logmodel.predict(X\_test)

#regression c-efficients and intercept

print("Regression intercept is",logmodel.intercept\_)

print("Regression coefficient is",logmodel.coef\_)

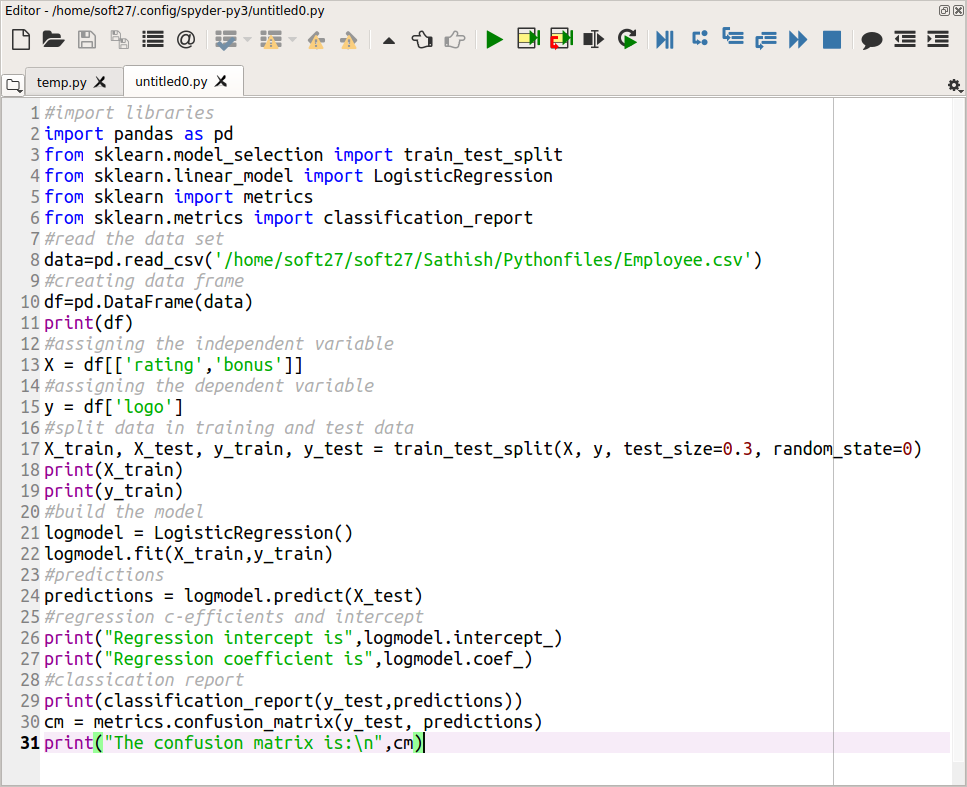
#classication report

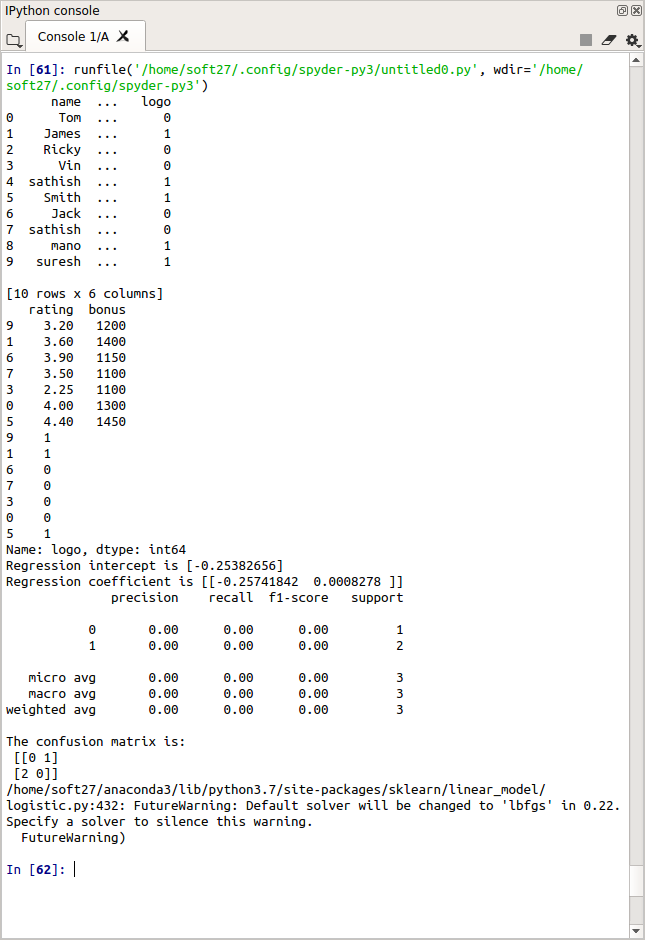
print(classification\_report(y\_test,predictions))

cm = metrics.confusion\_matrix(y\_test, predictions)

print("The confusion matrix is:\n",cm)

**Screen shots:**

****

****