الجامعة السعودية الإلكترونية
SAUDI ELECTRONIC UNIVERSITY
2011-1432
كلية الحوسبة والمعلوماتية |
College of Computing and Informatics

Semester 1 – 2021/2022

Course Code	DS550
Course Name	Machine Learning
Assignment type	Critical Thinking
Module	07

Student ID	G200007615
Student Name	Abdulaziz Alqumayzi
CRN	15062

Solutions:

Critical Thinking Assignment 3

Introduction

In this activity, we will define and explain Naive Bayes algorithm. Provide a python programming example on how to apply the algorithm on a dataset. The dataset used is **Blood Transfusion Service Center Data Set** from UCI Machine Learning Repository.

Data Set Information: To demonstrate the RFMTC marketing model (a modified version of RFM), this study adopted the donor database of Blood Transfusion Service Center in Hsin-Chu City in Taiwan. The center passes their blood transfusion service bus to one university in Hsin-Chu City to gather blood donated about every three months. To build a FRMTC model, we selected 748 donors at random from the donor database. These 748 donor data, each one included R (Recency - months since last donation), F (Frequency - total number of donation), M (Monetary - total blood donated in c.c.), T (Time - months since first donation), and a binary variable representing whether he/she donated blood in March 2007 (1 stand for donating blood; 0 stands for not donating blood).

Attribute information in Table 1.

Table 1

Attribute Information

Column	Data type
R (Recency - months since last donation)	Integer
F (Frequency - total number of donation)	Integer
M (Monetary - total blood donated in c.c.)	Integer
T (Time - months since first donation)	Integer
D (whether he/she donated blood in March 2007 “1 stand for donating blood; 0 stands for not donating blood”)	Integer

Naive Bayes Algorithms

Naive Bayes techniques are a collection of supervised learning algorithms based on Bayes' theorem and the "naive" assumption of conditional independence between every pair of features given the class variable's value.

The Bayes theorem is used to calculate the likelihood of an outcome given a collection of conditions. Naive Bayes algorithms are a family of effective and easy-to-train classifiers. The dynamic is based on inverting the conditional probabilities (which are linked to the causes) so that the inquiry may be stated as a function of measurable quantities. Naive Bayes algorithms are multi-purpose classifiers that may be found in a wide range of applications. However, in all circumstances where the likelihood of a class is governed by the probabilities of certain causative elements, their performance is very good.

Naive Bayes in scikit-learn

Gaussian, Multinomial, Complement, Bernoulli, and Categorical are five Naive Bayes variations implemented in scikit-learn, each based on the same number of distinct statistical distributions.

Gaussian Naive Bayes: *GaussianNB* implements the Gaussian Naive Bayes algorithm for classification.

Multinomial Naive Bayes: *MultinomialNB* implements the naive Bayes algorithm for multinomially distributed data and is one of the two classic naive Bayes variants used in text classification.

Complement Naive Bayes: *ComplementNB* implements the complement naive Bayes (CNB) algorithm. CNB is an adaptation of the standard multinomial naive Bayes (MNB) algorithm that is particularly suited for imbalanced data sets.

Bernoulli Naive Bayes: *BernoulliNB* implements the naive Bayes training and classification algorithms for data that is distributed according to multivariate Bernoulli distributions; i.e., there may be multiple features but each one is assumed to be a binary-valued (Bernoulli, boolean) variable.

Categorical Naive Bayes: *CategoricalNB* implements the categorical naive Bayes algorithm for categorically distributed data. It assumes that each feature, which is described by the index i, has its own categorical distribution.

Python Code

```
# importing needed packages
import numpy as np
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.naive_bayes import GaussianNB

#importing datasets
df = pd.read_csv('transfusion.csv')

# check if null values exists
df.info()

# descriptive statistics of the dataset
df.describe()

# exclude the target column from predictors columns
df_predictors = df.drop(columns='D')

# spilt the dataset into test and train sets. 80% test 20% train
X_train, X_test, y_train, y_test = train_test_split(df_predictors, df['D'],
test_size=0.8, random_state=9)
# fitting Gaussian Naive Bayes to the dataset
gnb = GaussianNB()
gnb = gnb.fit(X_train, y_train)

# validate the model

#Perform classification on an array of test vectors X.
y_pred = gnb.predict(X_test)
print(y_pred)

# Return log-probability estimates for the test vector X.
print(gnb.predict_log_proba(X_test))

# Return probability estimates for the test vector X.
print(gnb.predict_proba(X_test))

# Return the mean accuracy on the given test data and labels.
print(gnb.score(X_test,y_test))
```

References

Bonaccorso, G. (2018). *Machine learning algorithms: Popular algorithms for Data Science and Machine Learning*. Packt

Yeh, I.-C. (2008). *Blood Transfusion Service Center Data Set*. UCI Machine Learning Repository. Retrieved 2021, from https://archive.ics.uci.edu/ml/datasets/Blood+Transfusion+Service+Center.

1.9. naive Bayes. scikit-learn.org. (2021). Retrieved November 6, 2021, from https://scikit-learn.org/stable/modules/naive_bayes.html.

sklearn.naive_bayes.GaussianNB. scikit-learn.org. (2021). Retrieved November 6, 2021, from https://scikit-learn.org/stable/modules/generated/sklearn.naive_bayes.GaussianNB.html.