

LP1 Assignment DA R2

Naive Bayes algorithm for classification on Pima Indians Diabetes dataset.

Date - 7th September, 2020.

Assignment Number - DA R2

Title

Naive Bayes algorithm for classification on Pima Indians Diabetes dataset.

Problem Definition

Download Pima Indians Diabetes dataset. Use Naive Bayes Algorithm for classification

- Load the data from the CSV file and split it into training and test datasets.
- Summarize the properties in the training dataset so that we can calculate probabilities and make predictions.
- Classify samples from a test dataset and a summarized training dataset.

Learning Objectives

- Learn Naive Bayes algorithm
- Learn to summarize the properties in the training dataset.
- Learn to split the dataset into training and test datasets.
- Learn to classify samples from a test dataset and a summarized training dataset.

Learning Outcomes

I will be able to summarize the properties of the dataset, split the dataset into training and test data, apply Naive Bayes algorithm for classification application.

Software Packages and Hardware Apparatus Used

- Operating System : 64-bit Ubuntu 18.04

- Programming Language : Python 3
- Jupyter Notebook Environment : Google Colaboratory
- Python Libraries : Sklearn, Pandas, Matplotlib

Related Mathematics

Mathematical Model

Let S be the system set:

$$S = \{s; e; X; Y; F_{me}; DD; NDD; F_c; S_c\}$$

s =start state

e =end state i.e. classification of samples from the test dataset

X =set of inputs

$$X = \{X_1\}$$

where X_1 = Pima Indians Diabetes dataset (768 records, 8 columns)

Y =set of outputs

$$Y = \{Y_1, Y_2\}$$

- Y_1 = Confusion Matrix
- Y_2 = Accuracy Score

F_{me} is the set of main functions

$F_{me} = f_0$ where

- f_0 = Display Function

F_f is the set of friend functions

$F_f = \{f_1, f_2, f_3, f_4\}$ where

- f_1 = function to load dataset into dataframe
- f_2 = function to split dataset into training and test datasets
- f_3 = function to Normalize dataset
- f_4 = function to invoke Naive Bayes classifier

DD = Deterministic Data

- PIMA Indians diabetes dataset

NDD = Non-deterministic data

- null values in the dataset

Fc = failure case

- Failed to classify the record into correct class

Concepts related Theory

The Bayesian Classification represents a supervised learning method as well as a statistical method for classification. Assumes an underlying probabilistic model and it allows us to capture uncertainty about the model in a principled way by determining probabilities of the outcomes. It can solve diagnostic and predictive problems. This Classification is named after Thomas Bayes, who proposed the Bayes Theorem. Bayesian classification provides practical learning algorithms and prior knowledge and observed data can be combined. Bayesian Classification provides a useful perspective for understanding and evaluating many learning algorithms. It calculates explicit probabilities for hypotheses and it is robust to noise in input data.

The fundamental Naive Bayes assumption is that each feature makes an independent and equal contribution to the outcome

Bayes' Theorem Bayes' Theorem finds the probability of an event occurring given the probability of another event that has already occurred. Bayes' theorem is stated mathematically as the following equation:

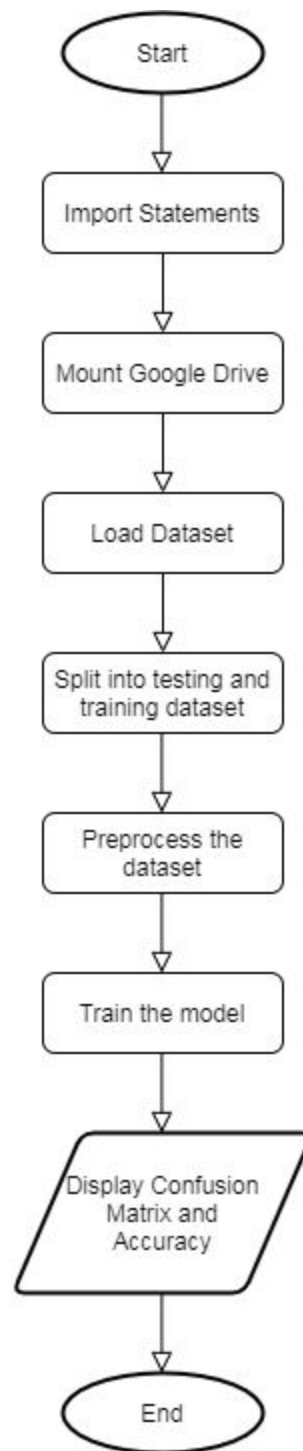
$$P(A|B) = \frac{P(B|A)P(A)}{P(B)}$$

Now, it's time to put a **naive assumption** to the Bayes' theorem, which is, independence among the features. So now, we split evidence into the independent parts. Now, if any two events A and B are independent, then,

$$P(A,B) = P(A)P(B)$$

Gaussian Naive Bayes classifier In Gaussian Naive Bayes, continuous values associated with each feature are assumed to be distributed according to a Gaussian distribution. A Gaussian distribution is also called Normal distribution. When plotted, it gives a bell shaped curve which is symmetric about the mean of the feature values

Flowchart



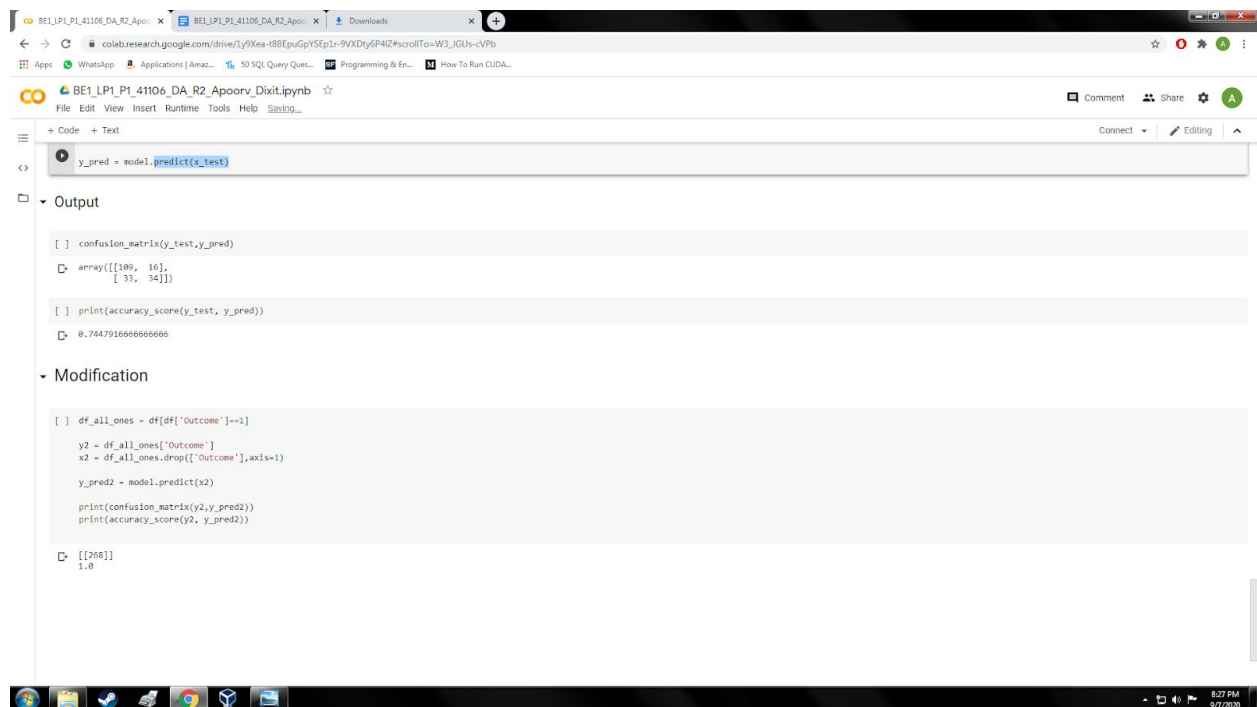
Dataset Description

This dataset describes the medical records for Pima Indians and whether or not each patient will have an onset of diabetes within five years.

Fields description follow:

1. Pregnancies = Number of times pregnant
2. Glucose = Plasma glucose concentration
3. BloodPressure = Diastolic blood pressure (mm Hg)
4. SkinThickness = Triceps skinfold thickness (mm)
5. Insulin = 2-Hour serum insulin (mu U/ml)
6. BMI = Body mass index (weight in kg/(height in m)^2)
7. DiabetesPedigreeFunction = Diabetes pedigree function
8. Age = Age (years)
9. Outcome = Class variable (1:tested positive for diabetes, 0: tested negative for diabetes)

Output Screenshots



The screenshot displays a Jupyter Notebook interface with a browser window at the top showing the Google Colab URL. The notebook has a single code cell with the following code:

```
y_pred = model.predict(x_test)
```

The output section shows the results of the code execution:

```
[ ] confusion_matrix(y_test, y_pred)
array([[109, 16],
       [ 33, 34]])

[ ] print(accuracy_score(y_test, y_pred))
0.7447916666666666
```

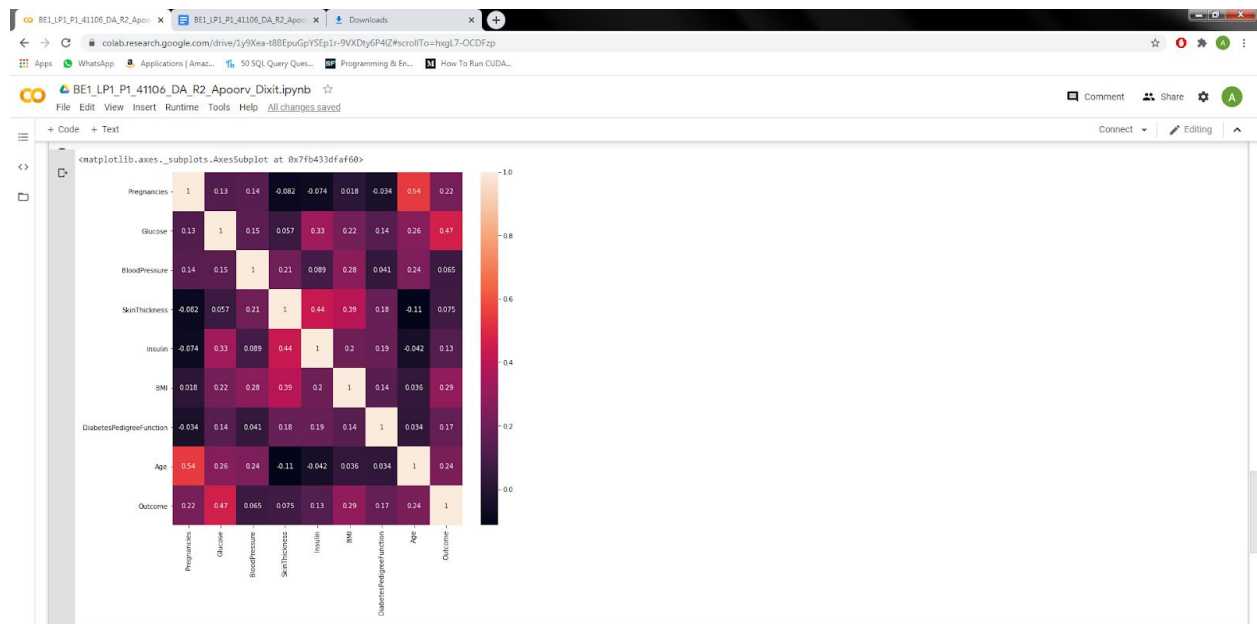
The modification section shows the code used to generate the output:

```
[ ] df_all_ones = df[df['Outcome']==1]
y2 = df_all_ones['Outcome']
x2 = df_all_ones.drop(['Outcome'], axis=1)
y_pred2 = model.predict(x2)
print(confusion_matrix(y2, y_pred2))
print(accuracy_score(y2, y_pred2))
```

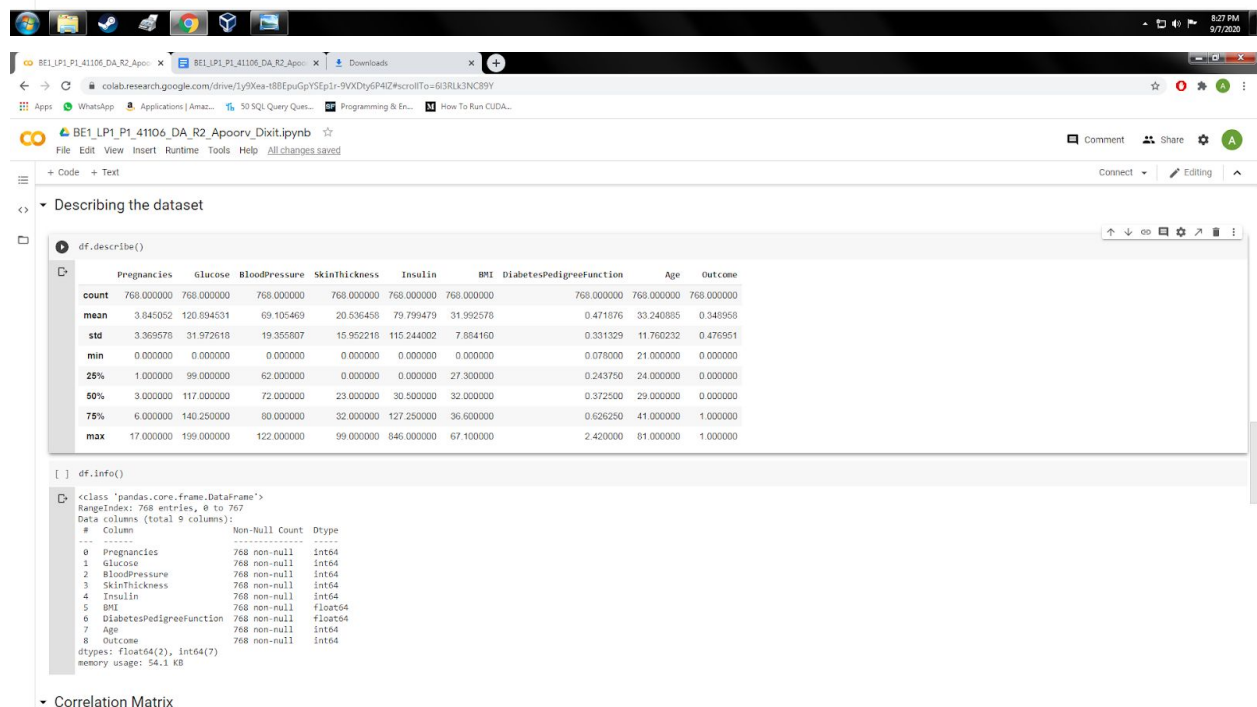
The final output of the modification code is:

```
[[208]]
1.0
```

The bottom of the screenshot shows the Windows taskbar with various application icons and the system clock indicating 8:27 PM on 9/7/2020.



Splitting the dataset



Correlation Matrix

Conclusion

I have successfully summarized the properties of the dataset, split the dataset into training and test data and applied Naive Bayes algorithm for classification application.