Pune Institute of Computer Technology

Dhankawadi,Pune

Maharashtra 411043

DEPARTMENT OF COMPUTER ENGINEERING

DA PROJECT REPORT

ON

"Predicting the edibility of mushrooms by analysing the dataset"

Submitted by:

Roll No. 41402 Aditi Kukde

Roll No. 41412 Ritika Deshpande

Roll No. 41414 Riya Disawal

Under the Guidance of

Prof. Hemant Shinde

1. Problem Statement

Mushroom edibility analysis: Analyze the dataset to determine which features indicate poisonous mushrooms. Also, which machine learning models perform the best on the given dataset?

2. Dataset description

- This dataset includes descriptions of hypothetical samples corresponding to 23 species of gilled mushrooms in the Agaricus and Lepiota Family Mushroom drawn from The Audubon Society Field Guide to North American Mushrooms (1981).
- Each species is identified as definitely edible, definitely poisonous, or of unknown edibility and not recommended. This latter class was combined with the poisonous one.
- The Guide clearly states that there is no simple rule for determining the edibility of a mushroom; no rule like "leaflets three, let it be" for Poisonous Oak and Ivy.
- This dataset was originally donated to the UCI Machine Learning repository.

3. H/W & S/W requirements

- Operating system: 64 bit linux operating system- ubuntu or a Windows operating system
- Ram: 4GB ram is recommended
- HDD: 50GB hard drive space is recommended
- Tools/Softwares: Python / Python2 / Python3(recommended)

4. Introduction

a. Classification Problem:
   i. We use the training dataset to get better boundary conditions which could be used to determine each target class.
   ii. Once the boundary conditions are determined, the next task is to predict the target class.
   iii. The whole process is known as classification.

b. Types of Classification Algorithms:
   i. Linear Classifiers
      1. Logistic regression
      2. Naive Bayes Classifier
      3. Fisher's Linear discriminant
   ii. Support vector machines
      1. Least squares support vector machines
   iii. Quadratic classifiers
   iv. Kernel Estimation

1. K Nearest Neighbour

v. Decision Trees
    1. Random forests

vi. Neural Networks

5. Objective

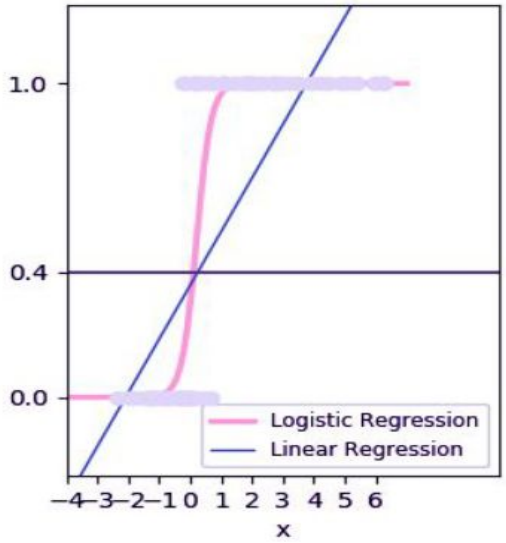
The main objectives of the project are -

- To learn various data preprocessing and techniques to get rid of noisy data.
- Learn to use various data visualization tools to draw necessary conclusions from analysis.
- Learn to implement various machine learning classification algorithms on the dataset and provide satisfactory results.

7. Machine Learning algorithms used:

1. <u>Logistic regression:</u>

- We use logistic regression for the binary classification of data-points. We perform categorical classification such that an output belongs to either of the two classes (1 or 0).
- For example – we can predict whether it will rain today or not, based on the current weather conditions.
- Two of the important parts of logistic regression are Hypothesis and Sigmoid Curve.With the help of this hypothesis, we can derive the likelihood of the event.
- The data generated from this hypothesis can fit into the log function that creates anS-shaped curve known as “sigmoid”. Using this log function, we can further predict the category of class.
- We can represent the Sigmoid as follows:



The produced graph is through this logistic function:

1 / (1 + e^-x)

The ‘e’ in the above equation represents the S-shaped curve that has values between 0 and 1. We write the equation for logistic regression as follows:

y = e^(b0 + b1*x) / (1 + e^(b0 + b1*x))

In the above equation, b0 and b1 are the two coefficients of the input x. We estimate these two coefficients using “maximum likelihood estimation”.

2. Naive Bayes Classifier

Naive Bayes is one of the powerful machine learning algorithms that is used for classification. It is an extension of the Bayes theorem wherein each feature assumes independence. It is used for a variety of tasks such as spam filtering and other areas of text classification.

Naive Bayes algorithm is useful for:

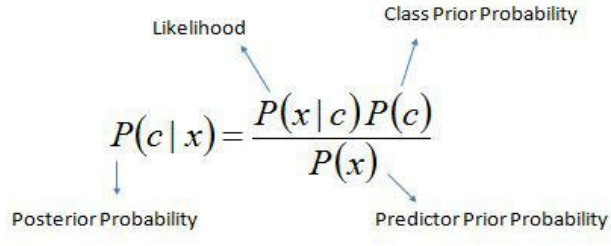
- Naive Bayes is an easy and quick way to predict the class of the dataset. Using this, one can perform a multi-class prediction.
- When the assumption of independence is valid, Naive Bayes is much more capable than the other algorithms like logistic regression. Furthermore, you will require less training data.

This is a classification technique based on an assumption of independence between predictors or what’s known as *Bayes’ theorem*. In simple terms, a Naive Bayes classifier assumes that the presence of a particular feature in a class is unrelated to the presence of any other feature.

For example, a fruit may be considered to be an apple if it is red, round, and about 3 inches in diameter. Even if these features depend on each other or upon the existence of the other features, a Naive Bayes Classifier would consider all of these properties to independently contribute to the probability that this fruit is an apple.

To build a Bayesian model is simple and particularly functional in case of enormous data sets. Along with simplicity, Naive Bayes is known to outperform sophisticated classification methods as well.

Bayes theorem provides a way of calculating posterior probability **P(c|x)**from **P(c)**, **P(x)**and **P(x|c)**. The expression for Posterior Probability is as follows.



$$P(c \mid X) = P(x_1 \mid c) \times P(x_2 \mid c) \times \cdots \times P(x_n \mid c) \times P(c)$$

Here,

- P(c|x) is the posterior probability of class(target) given predictor(attribute).
- P(c) is the prior probability of class.
- P(c) is the prior probability of class.
- P(x|c) is the likelihood which is the probability of predictor given class.
- P(x) is the prior probability of predictor.

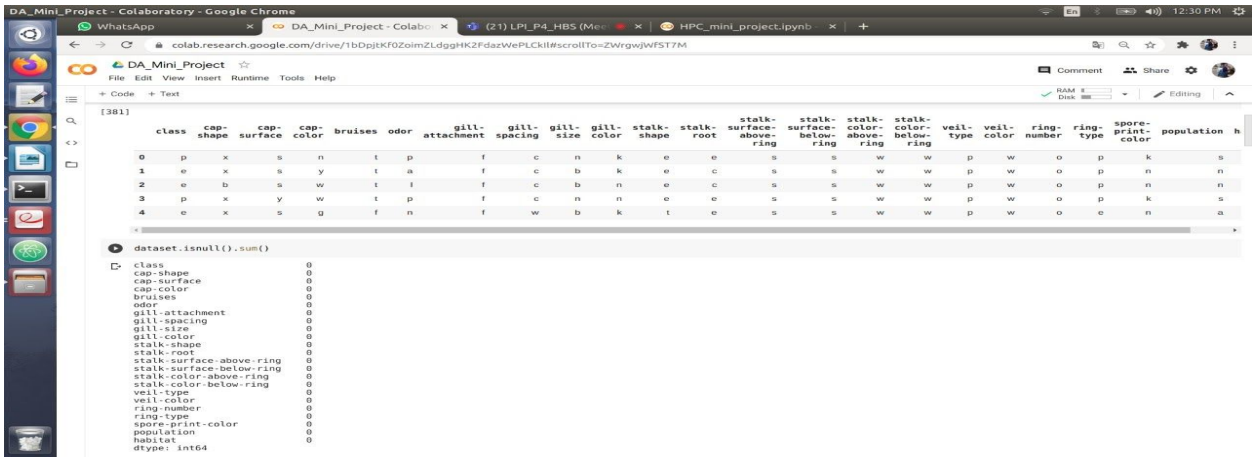
8. Test Cases and analysis

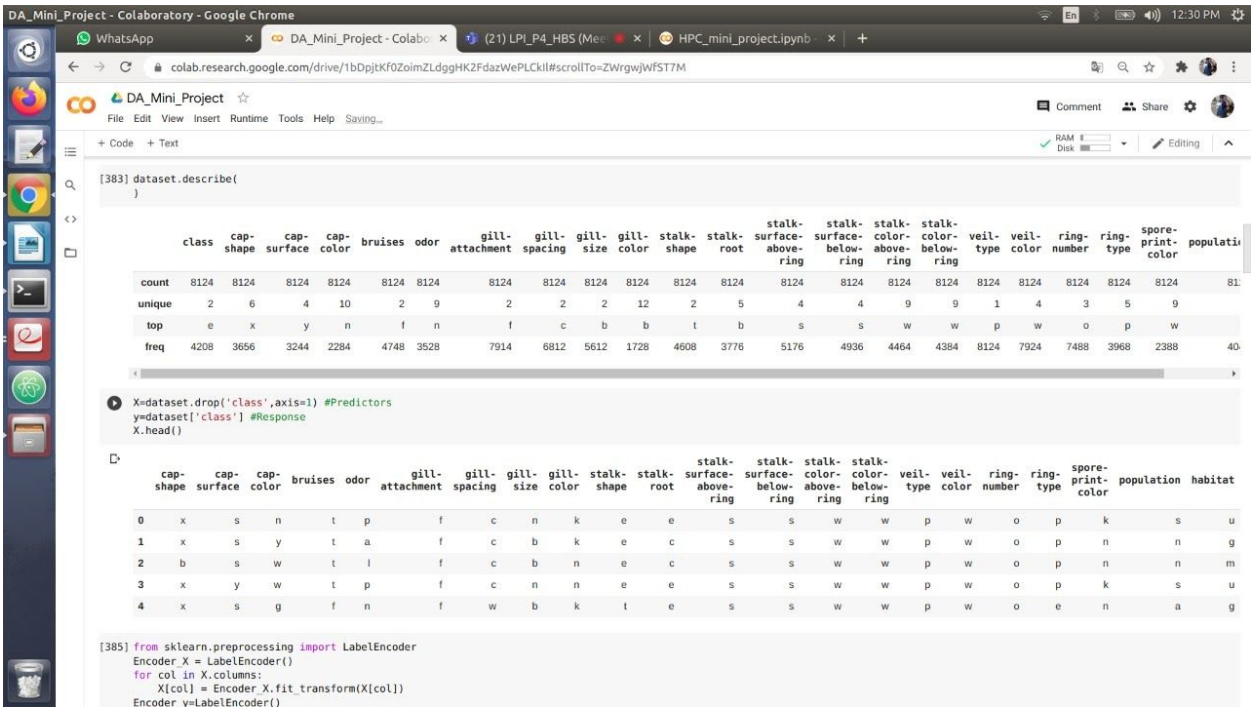
1. Logistic regression:
    a. Accuracy Score: 0.9057
    b. Confusion Matrix: [[2849, 102], [434, 2301]]
    c. Average accuracy: 0.9057
    d. Standard deviation: 0.0098

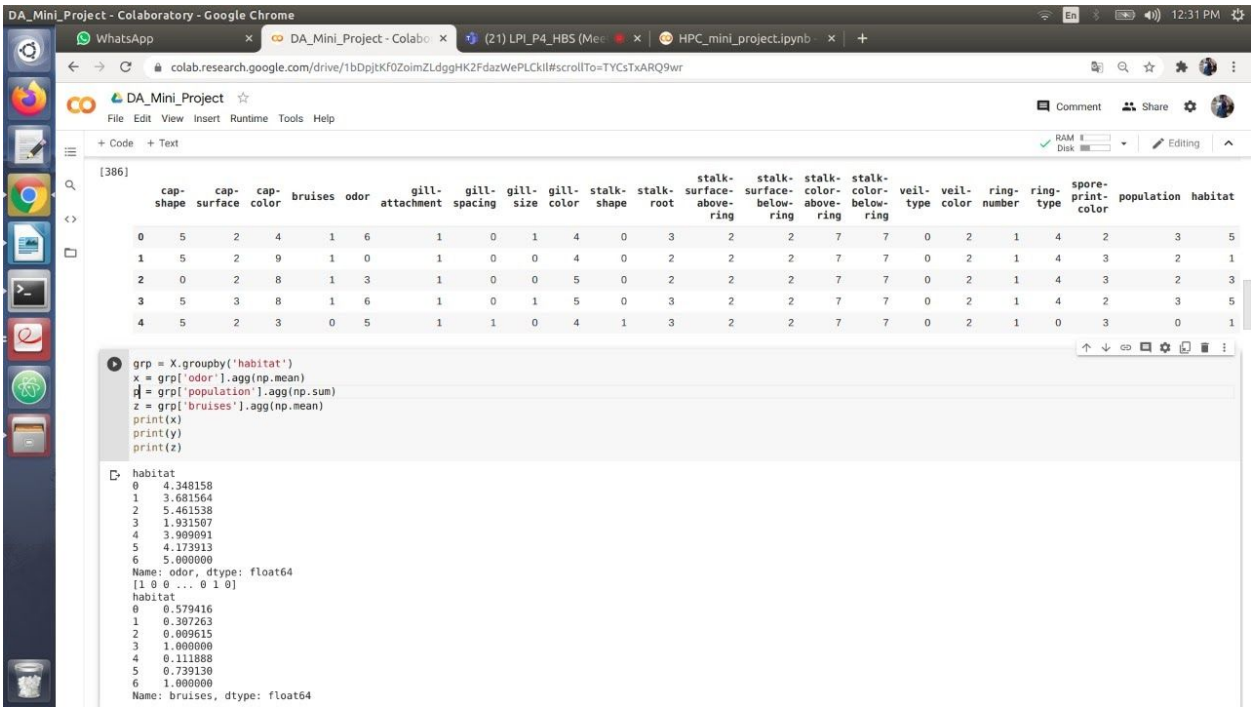
2. Naive Bayes :
    a. Accuracy Score: 0.8966
    b. Confusion Matrix: [[1215, 42], [210, 971]]
    c. Average accuracy: 0.8995
    d. Standard deviation: 0.0098

Thus, it is observed that the Logistic regression algorithm provides the best result.

9. Output Screenshots




DA_Mini_Project - Colaboratory - Google Chrome
12:30 PM
WhatsApp
DA_Mini_Project - Colabo
(21) LPI_P4_HBS (Mee
HPC_mini_project.ipynb
colab.research.google.com/drive/1bDpjtKf0ZoimZLdggHK2FdazWePLCkII#scrollTo=ZWrgwjWfST7M
DA_Mini_Project
File Edit View Insert Runtime Tools Help Saving...
Comment
Share
+ Code + Text
RAM
Disk
Editing
[383] dataset.describe(
)
class cap-shape cap-surface cap-color bruises odor gill-attachment gill-spacing gill-size gill-color stalk-shape stalk-root stalk-surface-above-ring stalk-surface-below-ring stalk-color-above-ring stalk-color-below-ring veil-type veil-color ring-number ring-type spore-print-color populati
count 8124 8124 8124 8124 8124 8124 8124 8124 8124 8124 8124 8124 8124 8124 8124 8124 8124 8124 8124 8124 8124 81
unique 2 6 4 10 2 9 2 2 2 12 2 5 4 4 9 9 1 4 3 5 9
top e x y n f n f c b b t b s s w w p w o p w
freq 4208 3656 3244 2284 4748 3528 7914 6812 5612 1728 4608 3776 5176 4936 4464 4384 8124 7924 7488 3968 2388 40
X=dataset.drop('class',axis=1) #Predictors
y=dataset['class'] #Response
X.head()
cap-shape cap-surface cap-color bruises odor gill-attachment gill-spacing gill-size gill-color stalk-shape stalk-root stalk-surface-above-ring stalk-surface-below-ring stalk-color-above-ring stalk-color-below-ring veil-type veil-color ring-number ring-type spore-print-color population habitat
0 x s n t p f c n k e e s s w w p w o p k s u
1 x s y t a f c b k e c s s w w p w o p n n g
2 b s w t l f c b n e c s s w w p w o p n n m
3 x y w t p f c n n e e s s w w p w o p k s u
4 x s g f n f w b k t e s s w w p w o e n a g
[385] from sklearn.preprocessing import LabelEncoder
Encoder_X = LabelEncoder()
for col in X.columns:
X[col] = Encoder_X.fit_transform(X[col])
Encoder_y=LabelEncoder()


DA_Mini_Project - Colaboratory - Google Chrome
12:31 PM
WhatsApp
DA_Mini_Project - Colabo
(21) LPI_P4_HBS (Mee
HPC_mini_project.ipynb
colab.research.google.com/drive/1bDpjtKf0ZoimZLdggHK2FdazWePLCkII#scrollTo=TYCsTxARQ9wr
DA_Mini_Project
File Edit View Insert Runtime Tools Help
Comment
Share
+ Code + Text
RAM
Disk
Editing
[386]
cap-shape cap-surface cap-color bruises odor gill-attachment gill-spacing gill-size gill-color stalk-shape stalk-root stalk-surface-above-ring stalk-surface-below-ring stalk-color-above-ring stalk-color-below-ring veil-type veil-color ring-number ring-type spore-print-color population habitat
0 5 2 4 1 6 1 0 1 4 0 3 2 2 7 7 0 2 1 4 2 3 5
1 5 2 9 1 0 1 0 0 4 0 2 2 2 7 7 0 2 1 4 3 2 1
2 0 2 8 1 3 1 0 0 5 0 2 2 2 7 7 0 2 1 4 3 2 3
3 5 3 8 1 6 1 0 1 5 0 3 2 2 7 7 0 2 1 4 2 3 5
4 5 2 3 0 5 1 1 0 4 1 3 2 2 7 7 0 2 1 0 3 0 1
grp = X.groupby('habitat')
x = grp['odor'].agg(np.mean)
y = grp['population'].agg(np.sum)
z = grp['bruises'].agg(np.mean)
print(x)
print(y)
print(z)
habitat
0 4.348158
1 3.681564
2 5.461538
3 1.931507
4 3.909091
5 4.173913
6 5.000000
Name: odor, dtype: float64
[1 0 0 ... 0 1 0]
habitat
0 0.579416
1 0.307263
2 0.009615
3 1.000000
4 0.111888
5 0.739130
6 1.000000
Name: bruises, dtype: float64

DA_Mini_Project - Colaboratory - Google Chrome

12:31 PM

WhatsApp | DA_Mini_Project - Colabo | (21) LPI_P4_HBS (Mee | HPC_mini_project.ipynb

colab.research.google.com/drive/1bDpjtKf0ZoimZLdggHK2FdazWePLCkIl#scrollTo=TYCsTxARQ9wr

DA_Mini_Project

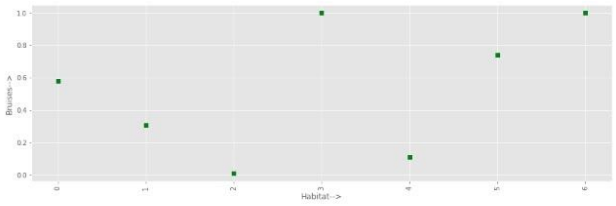
File Edit View Insert Runtime Tools Help All changes saved

Comment Share

+ Code + Text

RAM Disk Editing

[392]




```
from sklearn.preprocessing import LabelEncoder
Encoder_X = LabelEncoder()
for col in X.columns:
    X[col] = Encoder_X.fit_transform(X[col])
Encoder_y=LabelEncoder()
y = Encoder_y.fit_transform(y)
X.head()
```

	cap-shape	cap-surface	cap-color	bruises	odor	gill-attachment	gill-spacing	gill-size	gill-color	stalk-shape	stalk-root	stalk-surface-above-ring	stalk-surface-below-ring	stalk-color-above-ring	stalk-color-below-ring	veil-type	veil-color	ring-number	ring-type	spore-print-color	population	habitat
0	5	2	4	1	6	1	0	1	4	0	3	2	2	7	7	0	2	1	4	2	3	5
1	5	2	9	1	0	1	0	0	4	0	2	2	2	7	7	0	2	1	4	3	2	1
2	0	2	8	1	3	1	0	0	5	0	2	2	2	7	7	0	2	1	4	3	2	3
3	5	3	8	1	6	1	0	1	5	0	3	2	2	7	7	0	2	1	4	2	3	5
4	5	2	3	0	5	1	1	0	4	1	3	2	2	7	7	0	2	1	0	3	0	1

```
[394] X=pd.get_dummies(X,columns=X.columns,drop_first=True)
      X.head()
```

DA_Mini_Project - Colaboratory - Google Chrome

12:32 PM

WhatsApp | DA_Mini_Project - Colabo | (21) LPI_P4_HBS (Mee | HPC_mini_project.ipynb

colab.research.google.com/drive/1bDpjtKf0ZoimZLdggHK2FdazWePLCkIl#scrollTo=TYCsTxARQ9wr

DA_Mini_Project

File Edit View Insert Runtime Tools Help All changes saved

Comment Share

+ Code + Text

RAM Disk Editing

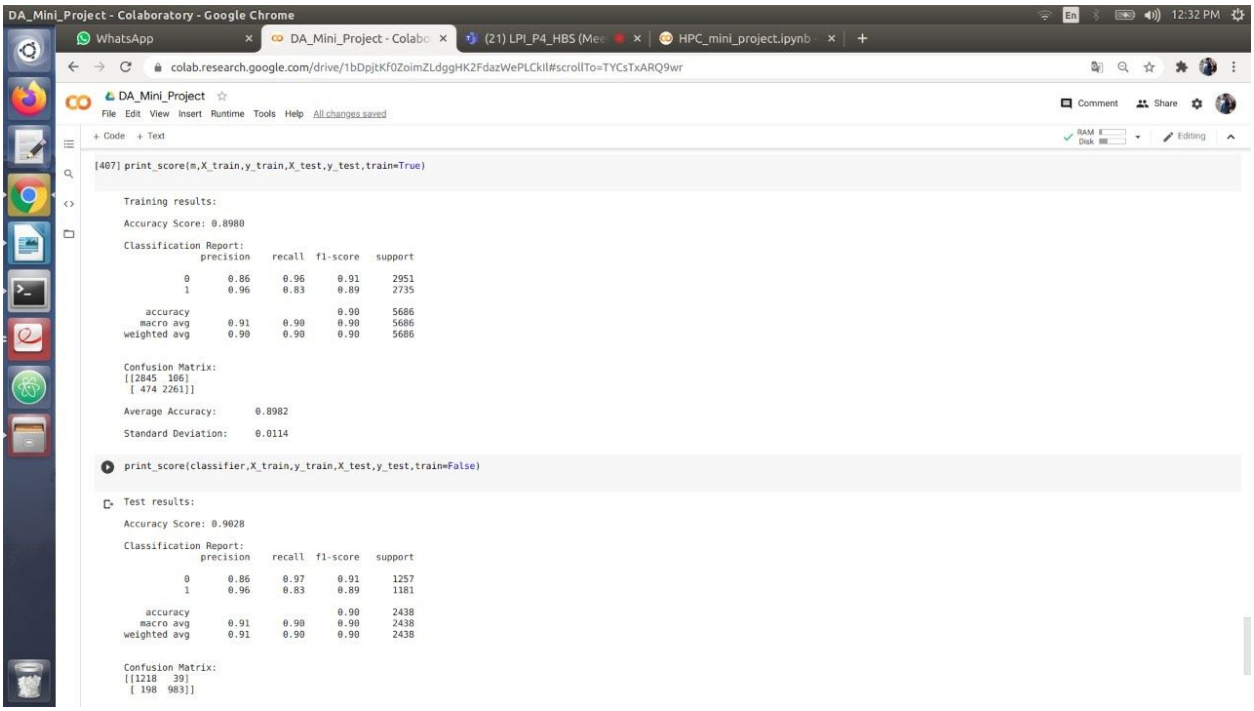
[403]

```
    Training results:

    Accuracy Score: 0.9057

    Classification Report:
                  precision    recall  f1-score   support

               0       0.87      0.97      0.91      2951
               1       0.96      0.84      0.90      2735

        accuracy                           0.91      5686
       macro avg       0.91      0.90      0.90      5686
    weighted avg       0.91      0.91      0.91      5686


    Confusion Matrix:
    [[2849  102]
     [ 434 2301]]

    Average Accuracy:       0.9057

    Standard Deviation:     0.0098
```

```
print_score(classifier,X_train,y_train,X_test,y_test,train=False)
```

```
    Test results:

    Accuracy Score: 0.9028

    Classification Report:
                  precision    recall  f1-score   support

               0       0.86      0.97      0.91      1257
               1       0.96      0.83      0.89      1181

        accuracy                           0.90      2438
       macro avg       0.91      0.90      0.90      2438
    weighted avg       0.91      0.90      0.90      2438


    Confusion Matrix:
    [[1218   39]
     [ 190  983]]
```

```
[405] from sklearn.naive_bayes import GaussianNB
```



10. Conclusion

We have successfully implemented classification algorithms to determine whether mushrooms will be deadly or safe to consume. The classification algorithms were analysed and compared to determine the best one and the results generated have been reported.