|  |
| --- |
|  |
| **Mushrooms Classification**  **Data Mining Project**  **Prof. Walaa Khaled**  **20201701607**  **20201701656**  **20201701653**  **20201701663**  **20201701662**  **Team: Emtinan Tarek Sadek**  **Mennatullah Emad Abdelsamie**  **Mariam Khaled Ahmed**  **Hana Tarek Mahmoud**  **Heba Magdy Moustafa** |

***Content***

* Data Preprocessing
* Classification Methods
  + Naive Bayes Classification
  + SVM Classification
  + KNN Classification
  + Logistic Regression Classification
  + Decision Tree Classification
  + Random Forest Classification
* Observation

***Introduction***

We’re representing the documentation of our python project performed on a dataset which contains features of edible and poisonous Mushrooms. We’ll examine the data and use classification algorithms which detect whether any mushroom is edible or poisonous regarding its specifications like cap shape-color or gill color-size. Let's examine the data.

***Data Preprocessing***

Preprocessing is an important step in data-mining as it cleans the data before it is processed to further improve accuracy. In this project, preprocessing methods are used to manipulate data in order to enhance the performance when executing classification models.

1. Importing the required Libraries to visualize and perform needed functionalities.

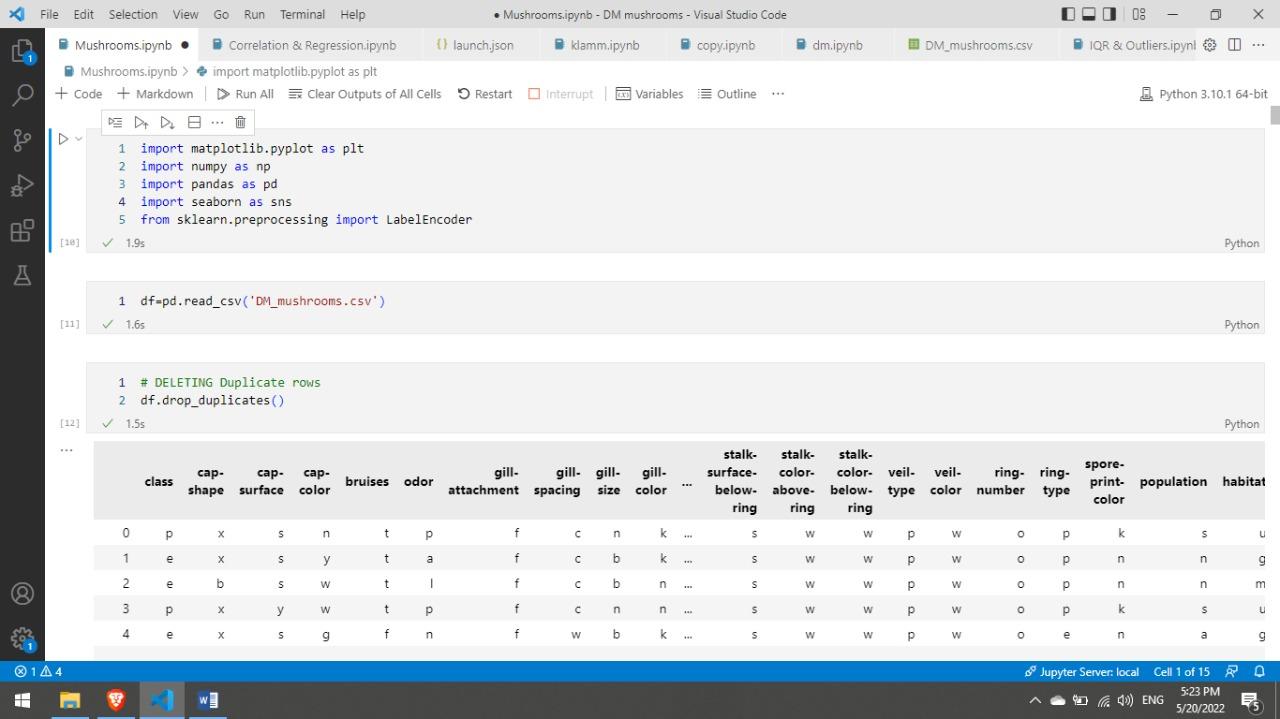
**matplotlib** for creating static, animated, and interactive visualizations in Python.

**pandas** for data manipulation and analysis.

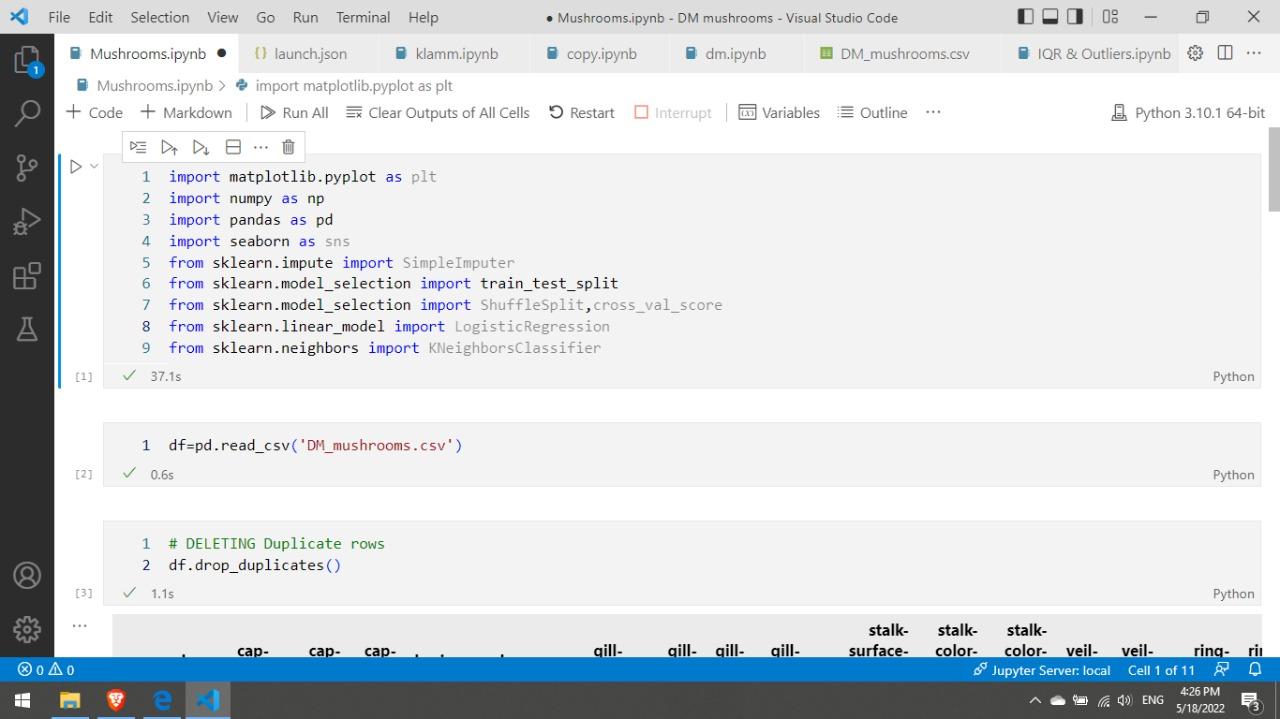
**numpy** for array-processing, along with a large collection of high-level mathematical functions to operate on these arrays.

**seaborn** for making statistical graphics in Python. It builds on top of matplotlib and integrates closely with pandas data structures.

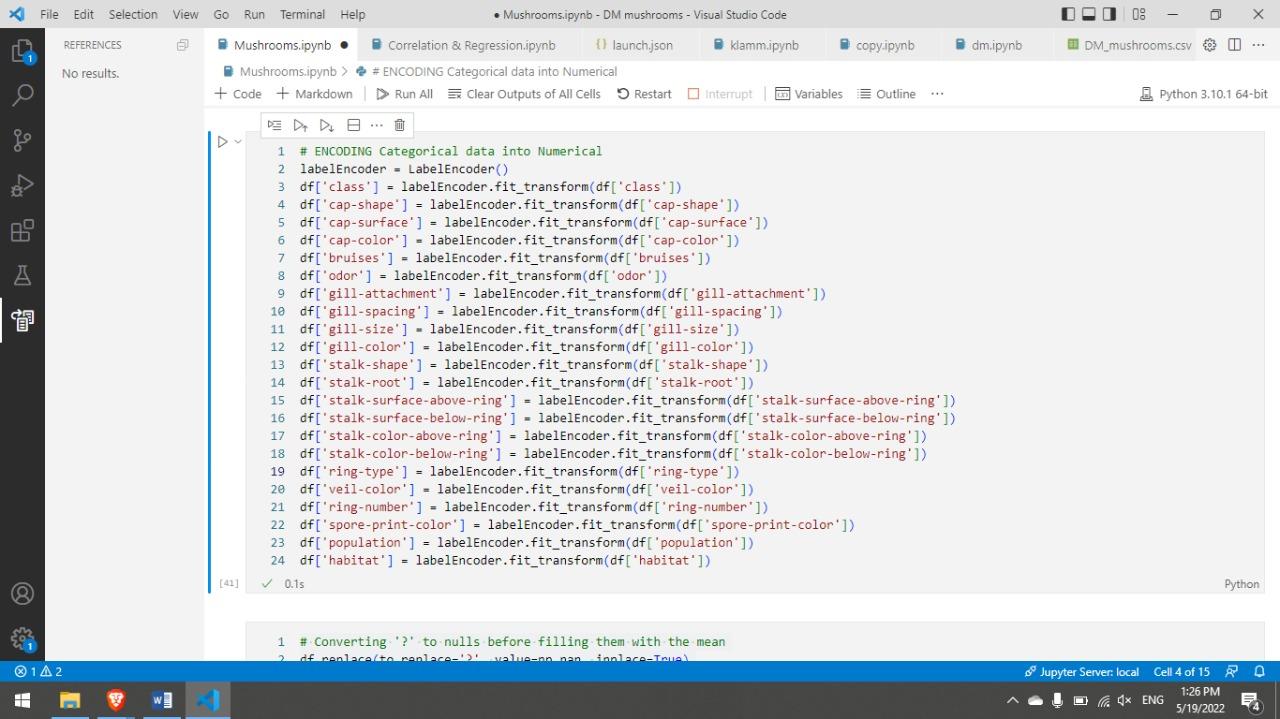
**sklearn.preprocessing** provides several functions and transformer classes to change raw features into suitable representations to be used.



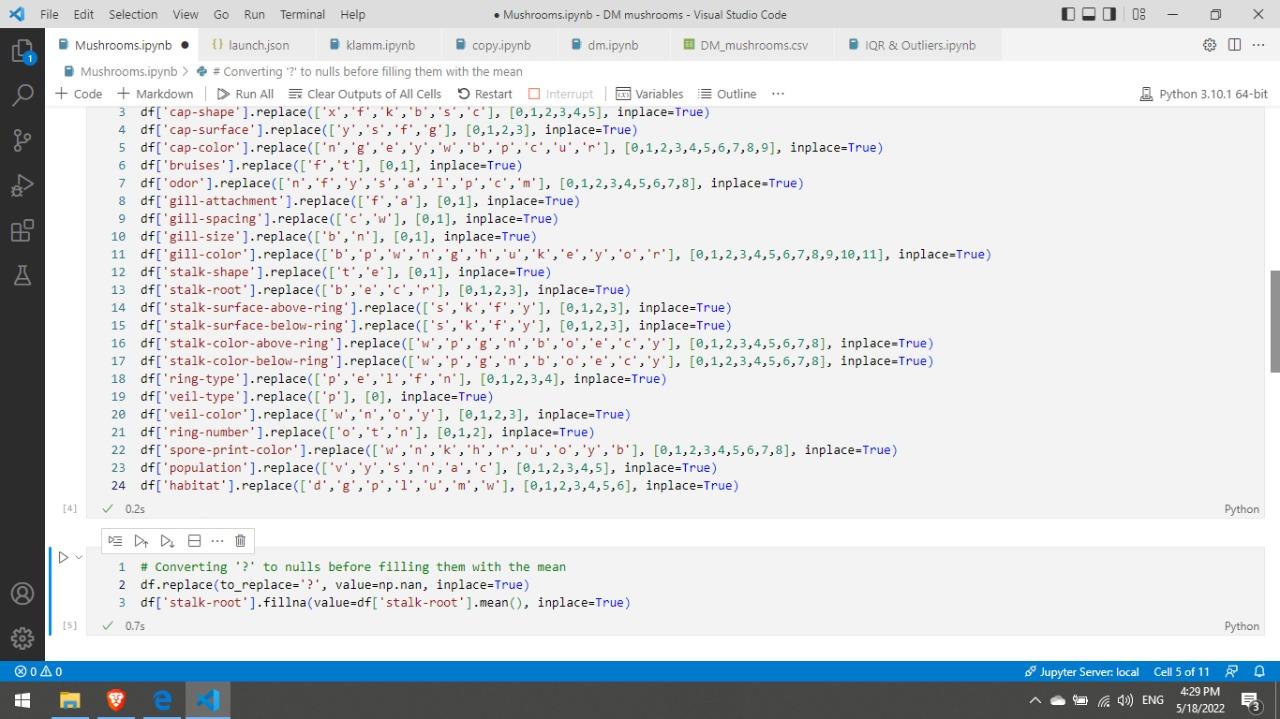
1. Starting with reading the csv file that holds Mushrooms data & Removing any existing duplicate rows to remove such noise.



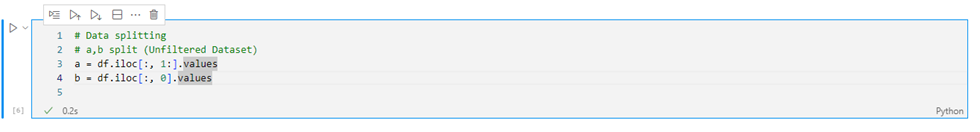
1. Encoding the Categorical data into Numerical using LabelEncoder to be able to make further enhancements & use them in further models.



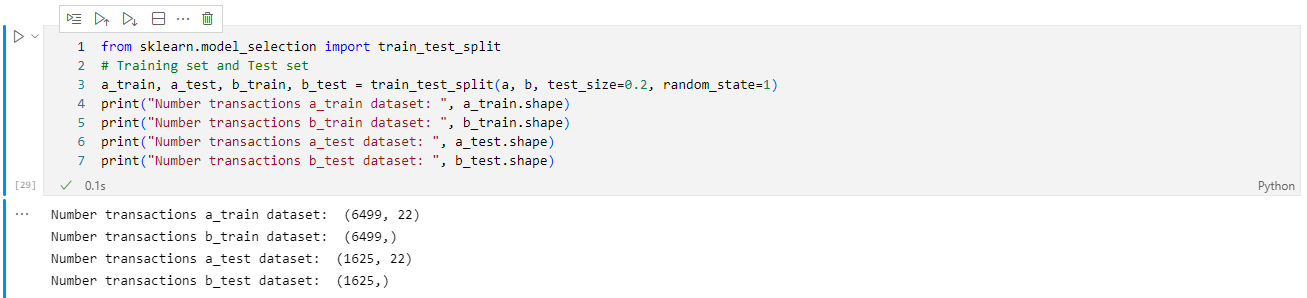
1. Converting missing values in the dataset marked as ‘?’ with their mean values. Stalk-Root is the only column observed containing missing values.



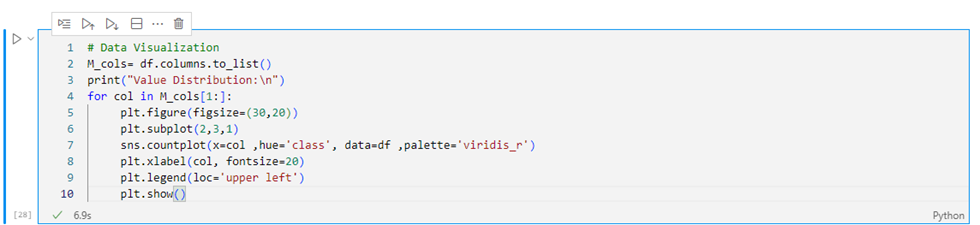
1. Here, the full, unfiltered dataset is split into A and B. Where A carries the independent columns, while B carries the dependent column “the Label”.

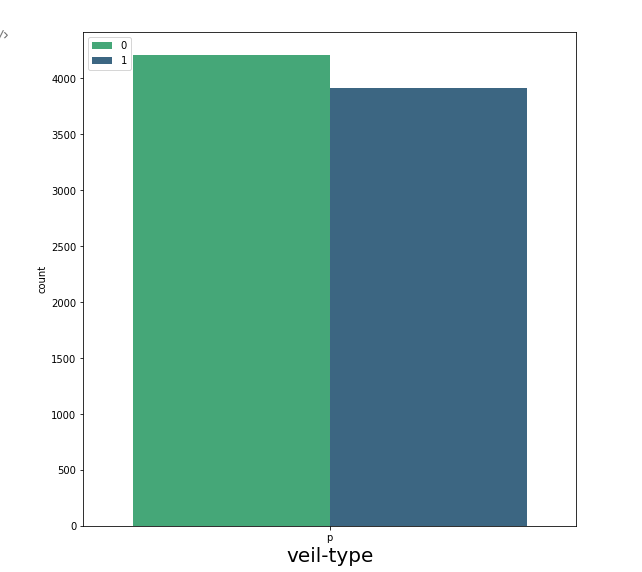


1. Next, the needed library is imported. Then, each of A and B are divided into training and testing sets. Noting that the training and testing percentages are 80% and 20% respectively. Providing that random state = 1, meaning it is guaranteed that the split will always be the same.

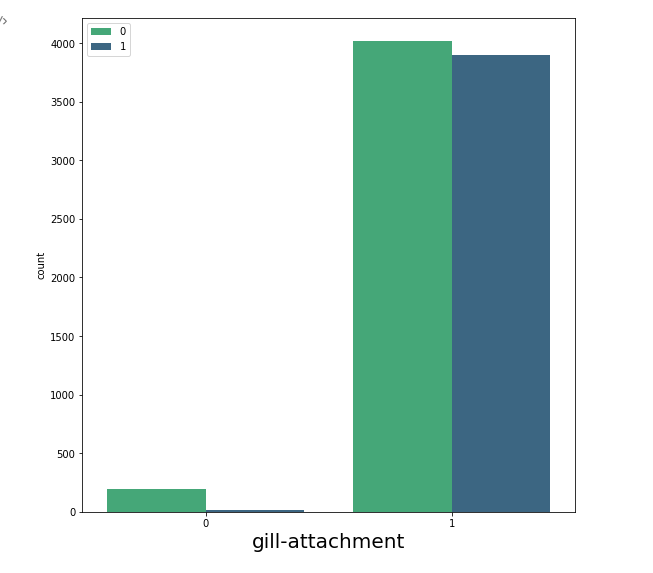


1. Next, the variables are plotted to get a clear look at all the attributes’ distribution in order to assist us in feature selection. After analyzing each graph, it was concluded that two rows should be dropped.





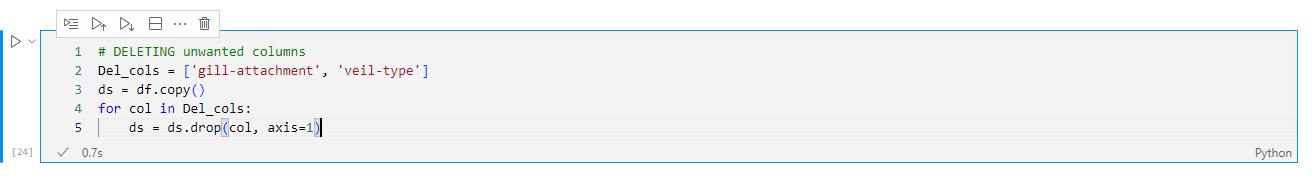
* From the ‘veil-type’ plot, it is observed that veil-type has only one unique value and hence won't contribute anything to the data. So it can be safely removed.



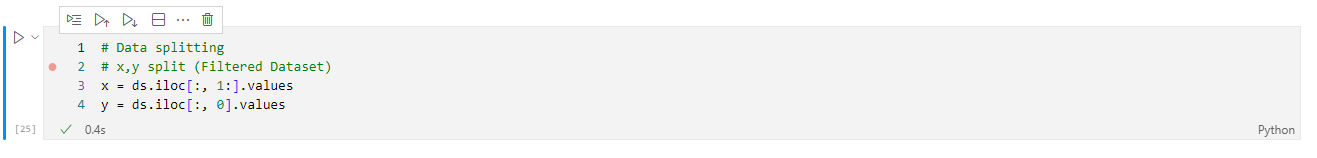
* From the ‘gill-attachment’ plot,

Almost our entire data ‘1’ gill attachment, therefore this column is pretty much useless in our analysis.

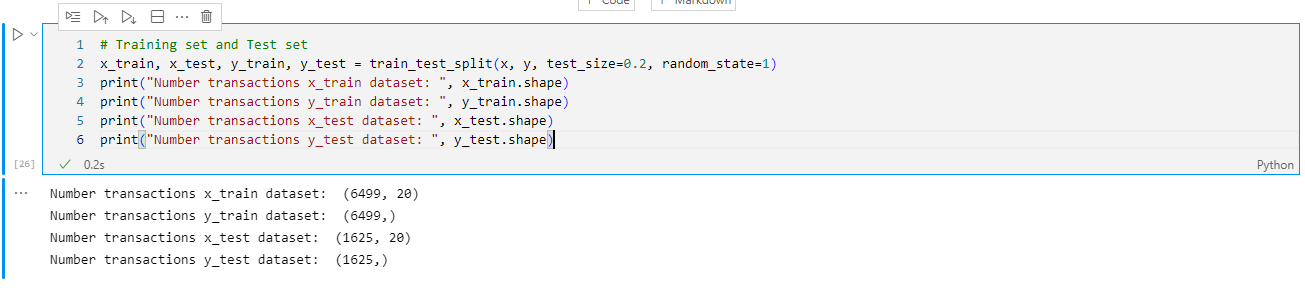
1. In this cell, a new, filtered dataframe is created where the mentioned attributes above were dropped to provide better, more accurate results.



1. Here, the unfiltered dataset is split into X and Y. Where X carries the independent columns, while Y carries the dependent column “the Label”.



1. Then, each of X and Y are divided into training and testing sets. Noting that the training and testing percentages are 80% and 20% respectively. Providing that random state = 1, meaning it is guaranteed that the split will always be the same.

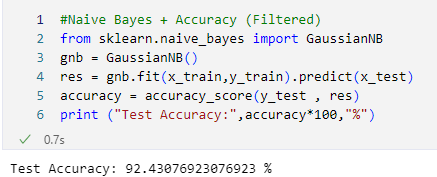


***Classification Methods***

* **Naive Bayes Classification**

It is a classification technique based on Bayes' Theorem with an assumption of independence among predictors. 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.

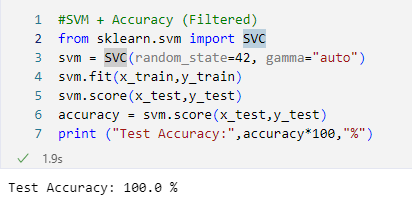
Importing the required library and applying the algorithm to the data set



* **SVM Classification**

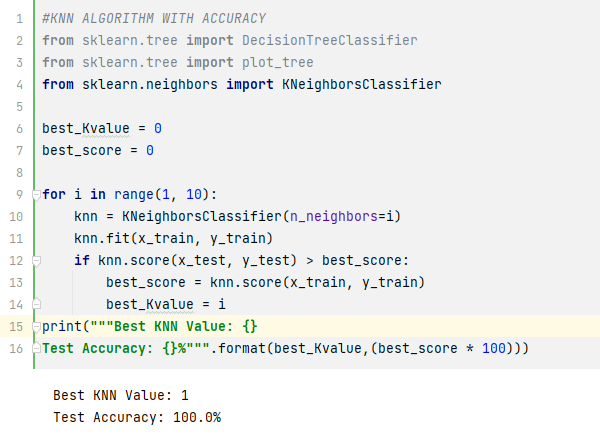
SVM is a linear model for classification and regression problems and works by creating a line or a hyperplane which separates the data into classes.

Importing the required library and applying the algorithm to the data set.

.

**KNN (K nearest neighbor) Classification:**

K nearest neighbor is an algorithm that classifies a data point based on how its neighbors are classified



* **Logistic Regression Classification:**

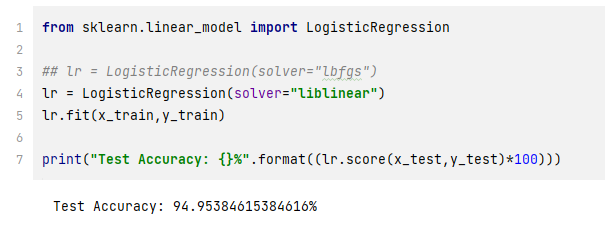
Logistic Regression is used for Classification to predict Binomial data either (0 , 1), (Yes , No) , (Success , Fail) , or many more (Binary Classification) and to predict Ordinal data which represents more than 2 discrete variables for example (A , B , C) or more (Multiclass Classification).

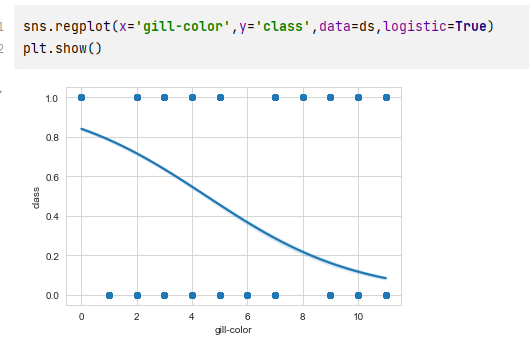
The main reason behind the “S - Shaped curve” of the Logistic Regression is because of being calculated using a Sigmoid Function. The value of the Sigmoid Function always lies between 0 and 1, which is why it’s used to solve categorical problems having two possible values.

Diagram

Description automatically generatedA picture containing chart

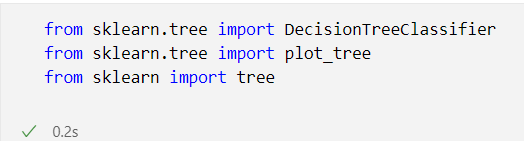
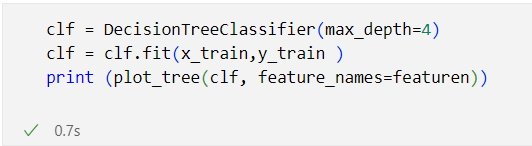
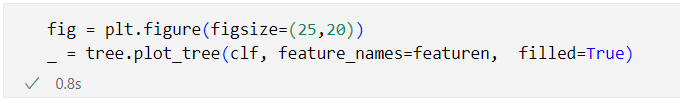
Description automatically generated

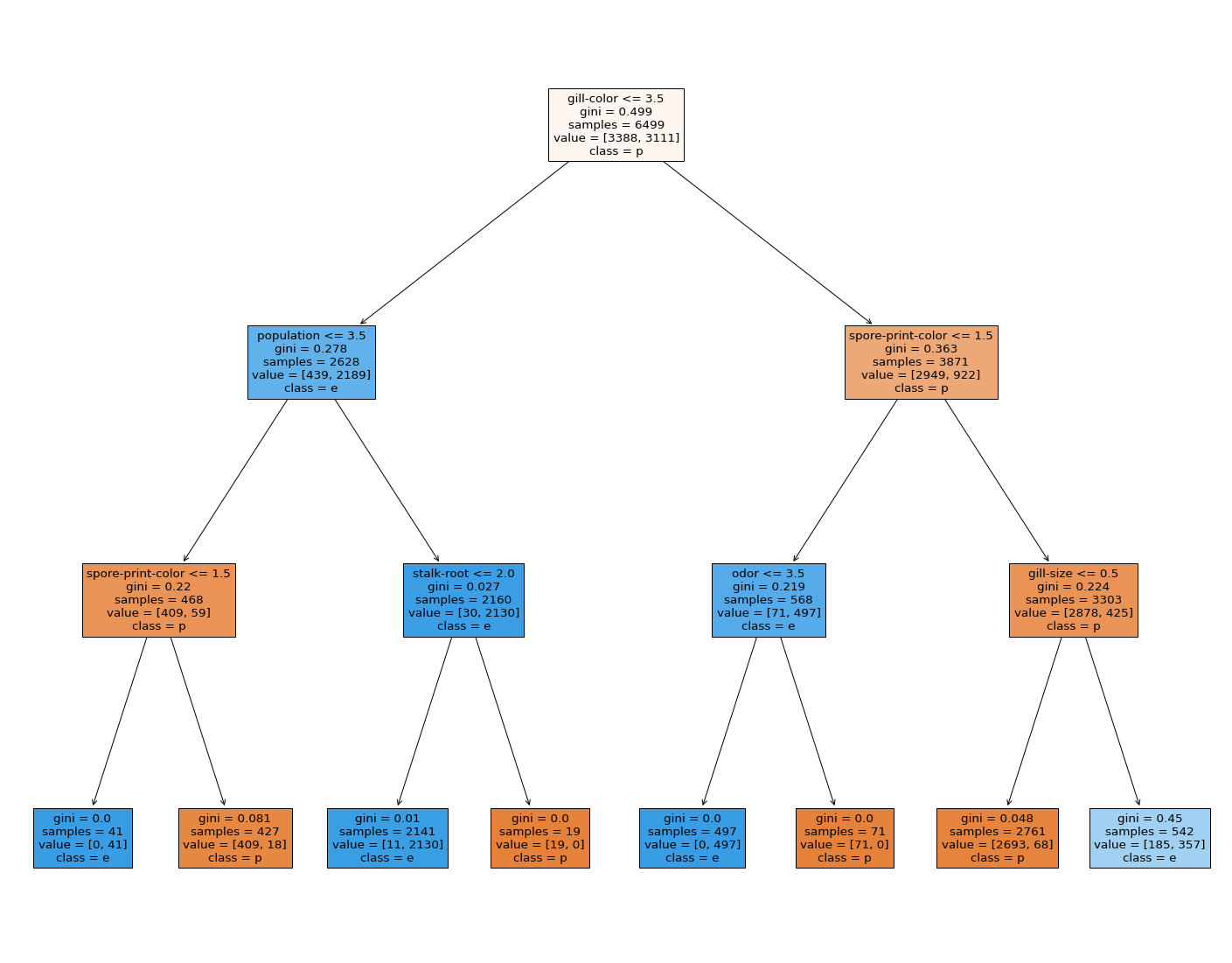


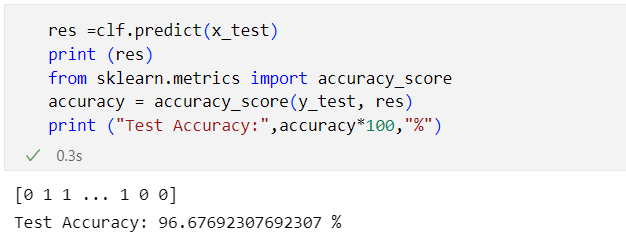


* **Decision Tree:**

We applied decision tree on our data by

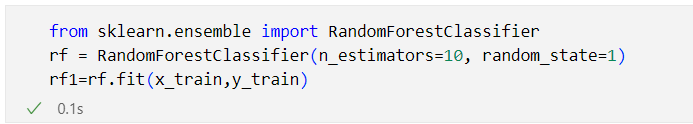
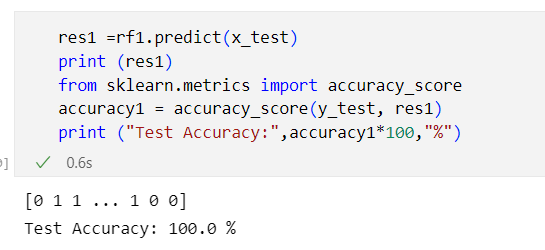
1. Importing the required libraries****
2. Create the actual decision tree and fit it with our data****
3. Then plot the tree ****

****

1. Check the accuracy****

* **Random Forest Classifier :**

**contains *number of decision trees on various subsets of the dataset and takes the average of all the decision trees results to improve the predictive accuracy of that dataset.***

1. Import **RandomForestClassifier** library and fit it to the data****
2. Checking the accuracy****

* **Observation :**

Finally, we conclude that the most suitable classification models are

SVM, KNN and Random Forest, since they have the highest percentages (100%) in accuracy.

Checking the results with a confusion matrix.

