**Mushroom Project**

***Problem statement***

**Data Set Information:**

This data set includes descriptions of hypothetical samples corresponding to 23 species of gilled mushrooms in the Agaricus and Lepiota Family (pp. 500-525). 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.

How Unique: Many folk traditions concern the defining features of poisonous mushrooms. However, there are no general identifiers for poisonous mushrooms, so such traditions are unreliable. Guidelines to identify particular mushrooms exist and will serve only if one knows which mushrooms are toxic

In this project, we will examine the data and build different **machine learning models** that will detect if the mushroom is **edible or poisonous** by its specifications like cap shape, cap color, gill color, etc. using different classifiers.

we will explore and analyze the data presented by "UCI Machine Learning" for mushroom classification. First, we shall analyze the data, by collecting its available characteristics, and then we will classify it to achieve maximum accuracy and precision

**TYPE OF PROBLEM:**

The above problem is a clear classification problem as we need to classify whether the mushroom is in which class ‘p’ or ‘e’ So this can be solved by any of the classification techniques like

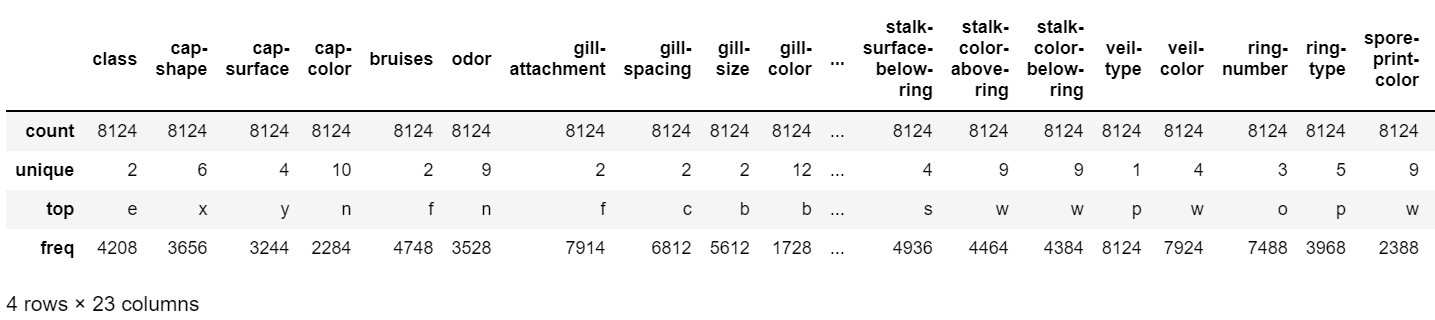
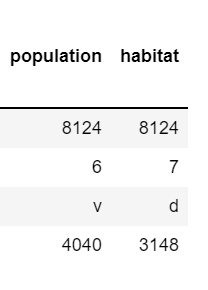
1. Logistic Regression .
2. Decision Tree Algorithm.
3. Random Forest Technique.
4. KNeighborsClassifier
5. Support Vector Classification

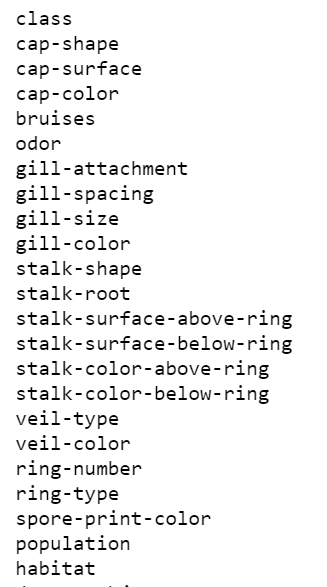
I have mentioned only few. We will be dealing with each of techniques later in this blog.

**Description about the Data Columns**

The dataset used in this project is [**mushrooms.csv**](https://www.kaggle.com/uciml/mushroom-classification) that contains 8124 instances of mushrooms with 23 features like cap-shape, cap-surface, cap-color, bruises, odor, etc.

We’ll use the specifications like cap shape, cap color, gill color, etc. to classify the mushrooms into edible and poisonous.By using df.describe() we got all the statistical values to compare the data in the columns



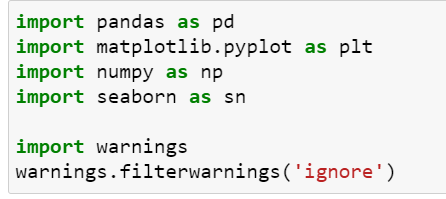
**Attribute Information:**

1. cap-shape: bell=b, conical=c, convex=x, flat=f, knobbed=k, sunken=s  
2. cap-surface: fibrous=f, grooves=g, scaly=y, smooth=s  
3. cap-color: brown=n,buff=b,cinnamon=c,gray=g,green=r, pink=p,purple=u,red=e,white=w,yellow=y  
4. bruises?: bruises=t,no=f  
5. odor: almond=a,anise=l,creosote=c,fishy=y,foul=f, musty=m,none=n,pungent=p,spicy=s  
6. gill-attachment: attached=a,descending=d,free=f,notched=n  
7. gill-spacing: close=c,crowded=w,distant=d  
8. gill-size: broad=b,narrow=n  
9. gill-color: black=k,brown=n,buff=b,chocolate=h,gray=g, green=r,orange=o,pink=p,purple=u,red=e, white=w,yellow=y  
10. stalk-shape: enlarging=e,tapering=t  
11. stalk-root: bulbous=b,club=c,cup=u,equal=e, rhizomorphs=z,rooted=r,missing=?  
12. stalk-surface-above-ring: fibrous=f,scaly=y,silky=k,smooth=s  
13. stalk-surface-below-ring: fibrous=f,scaly=y,silky=k,smooth=s  
14. stalk-color-above-ring: brown=n,buff=b,cinnamon=c,gray=g,orange=o, pink=p,red=e,white=w,yellow=y  
15. stalk-color-below-ring: brown=n,buff=b,cinnamon=c,gray=g,orange=o, pink=p,red=e,white=w,yellow=y  
16. veil-type: partial=p,universal=u  
17. veil-color: brown=n,orange=o,white=w,yellow=y  
18. ring-number: none=n,one=o,two=t  
19. ring-type: cobwebby=c,evanescent=e,flaring=f,large=l, none=n,pendant=p,sheathing=s,zone=z  
20. spore-print-color: black=k,brown=n,buff=b,chocolate=h,green=r, orange=o,purple=u,white=w,yellow=y  
21. population: abundant=a,clustered=c,numerous=n, scattered=s,several=v,solitary=y  
22. habitat: grasses=g,leaves=l,meadows=m,paths=p, urban=u,waste=w,woods=d

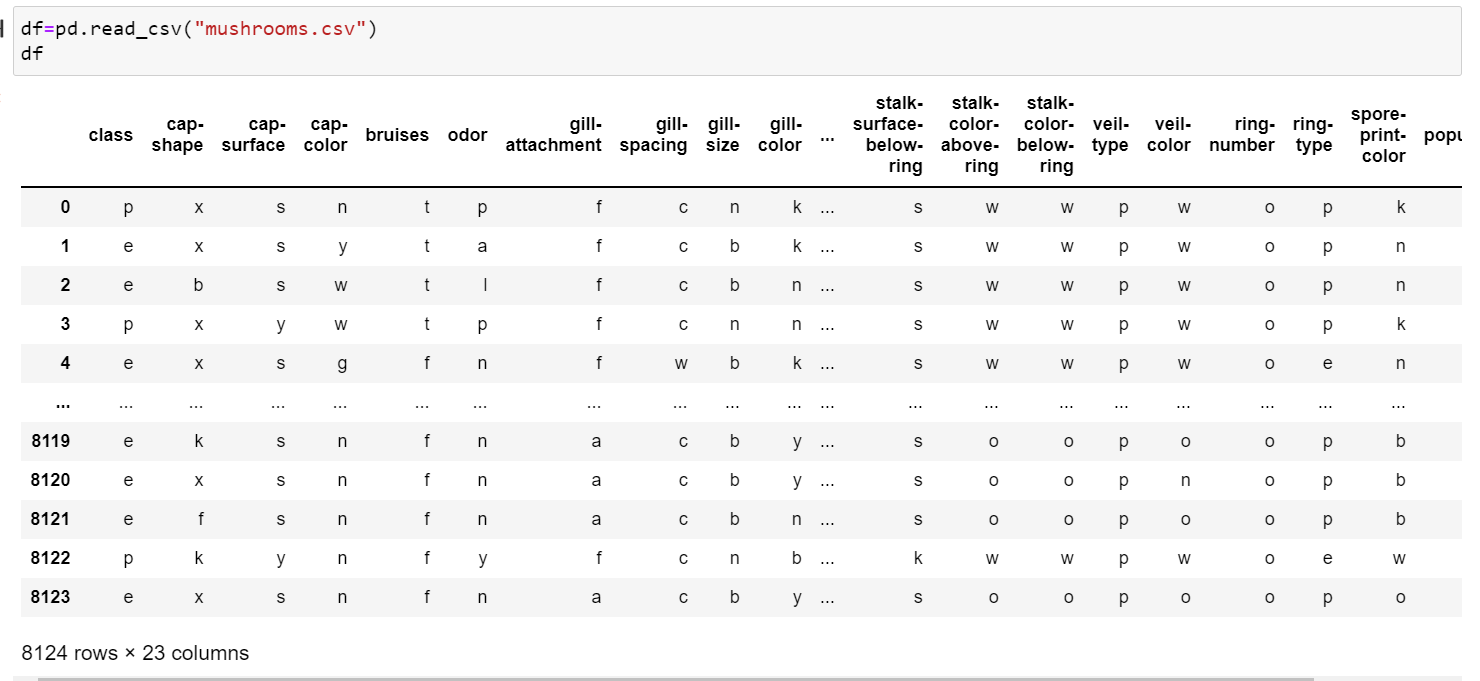
We’ll use the specifications like cap shape, cap color, gill color, etc. to classify the mushrooms into edible and poisonous.

***Data Analysis:***

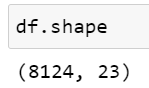
Now let me walk through the code. Firstly I just imported the necessary packages like pandas, numpy, seaborn etc. so that i can carry the necessary operations further



Now I am going to upload or read the files/data-sets using pandas. For this we used read\_csv

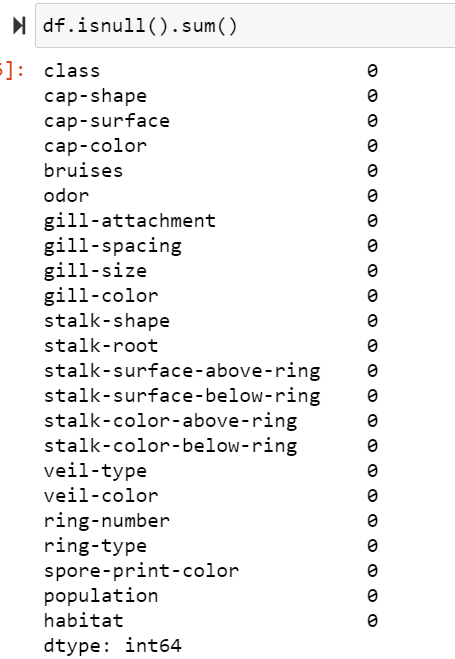


By using df.shape I understood that there are 8124 rows and 23 columns

****

By using df.dtypes we understood about the types of each columns. We have maximum columns as object

**Dealing with null values**

****

There are no null values present in the dataset. We have checked this with isnull().sum()

**Examining the Data**

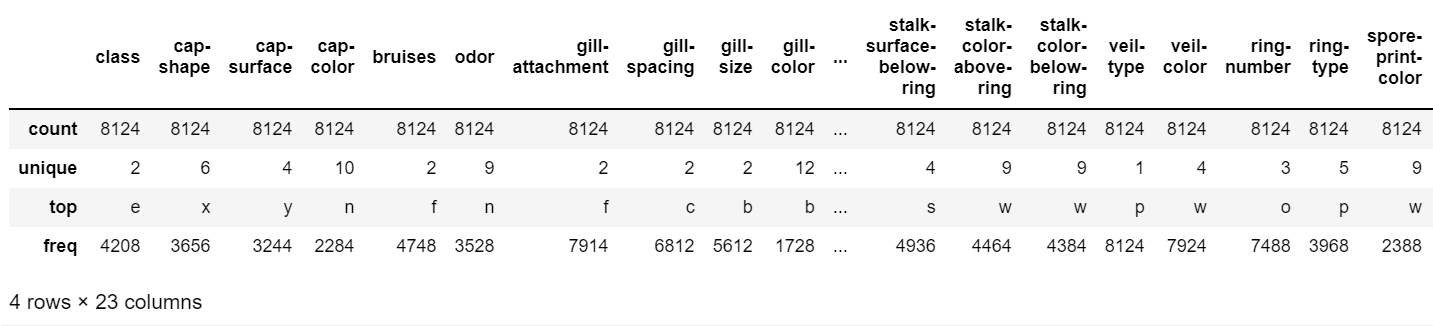
After importing the data, to learn more about the dataset,we will use **.head()** **.info()** and **.describe()** methods.



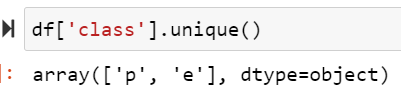
The **.describe()** method will give you the **statistics** of the columns.

* ***count*** shows the number of responses.
* ***unique*** shows the number of unique categorical values.
* ***top*** shows the highest-occurring categorical value.
* ***freq*** shows the frequency/count of the highest-occurring categorical value.

Here is the output:



## Unique occurrences of ‘class’ column



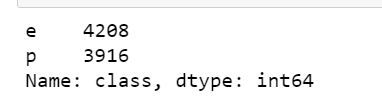
The **.unique()** method will give you the unique occurrences in the ‘class’ column of the dataset.

As we can see, there are two unique values in the ‘class’ column of the dataset namely

## Count of the unique occurrences of ‘class’ column



The **.value\_counts()** method will give you the count of the unique occurrences. Here is the output:



As we can see, there are **4208 occurrences of edible mushrooms** and **3916 occurrences of poisonous mushrooms** in the dataset.

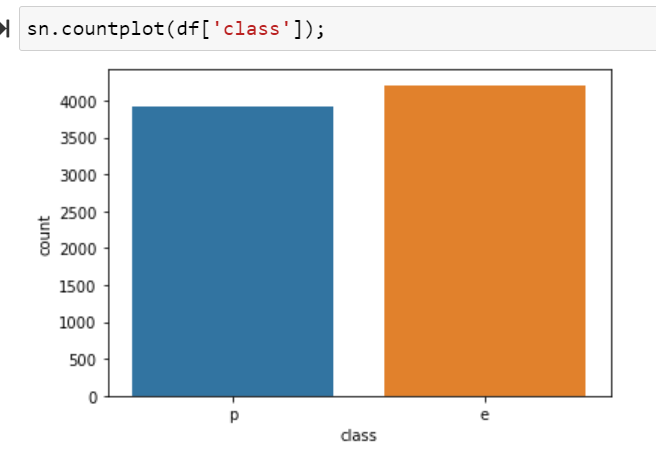
## Exploratory Data Analysis

Why are we doing EDA?

Like these there are many more we can assume. But one basic question you may get it …”Why are we doing all these ? Why can’t we do directly modelling the data instead of knowing all these…..” Well in some cases we can easily come to conclusion if we just to do EDA. Then there is no necessary for going through next models

## Now let’s visualize the count of edible and poisonous mushrooms using Seaborn

‘**e**’ and ‘**p**’, are **“count.values”** represents the count of those unique values i.e. **4208** and **3916** respectively. Here is the output of the bar graph:



Well don’t get to worry about the fancy names like exploratory data analysis and all. By looking at the columns description in the above paragraph, we can make many assumptions like

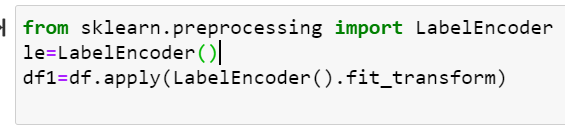
All 2 classes are equally distributed so there is no imbalance isssue exists

# Data Manipulation

The data is categorical so we’ll use **LabelEncoder** to convert it to ordinal. **LabelEncoder** converts each value in a column to a number.

This approach requires the category column to be of ‘**category**’ datatype. By default, a non-numerical column is of ‘**object**’ datatype. From the **df.describe()**method, we saw that our columns are of ‘object’ datatype. So we will have to change the type to ‘category’ before using this approach.

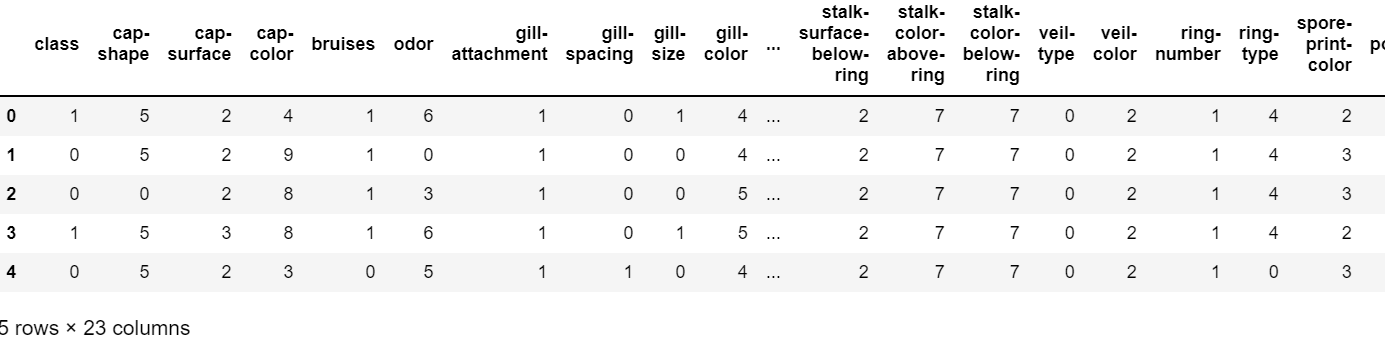
By using LabelEncoder converting all categorical values to numerical values.



Checking the dataset again

******

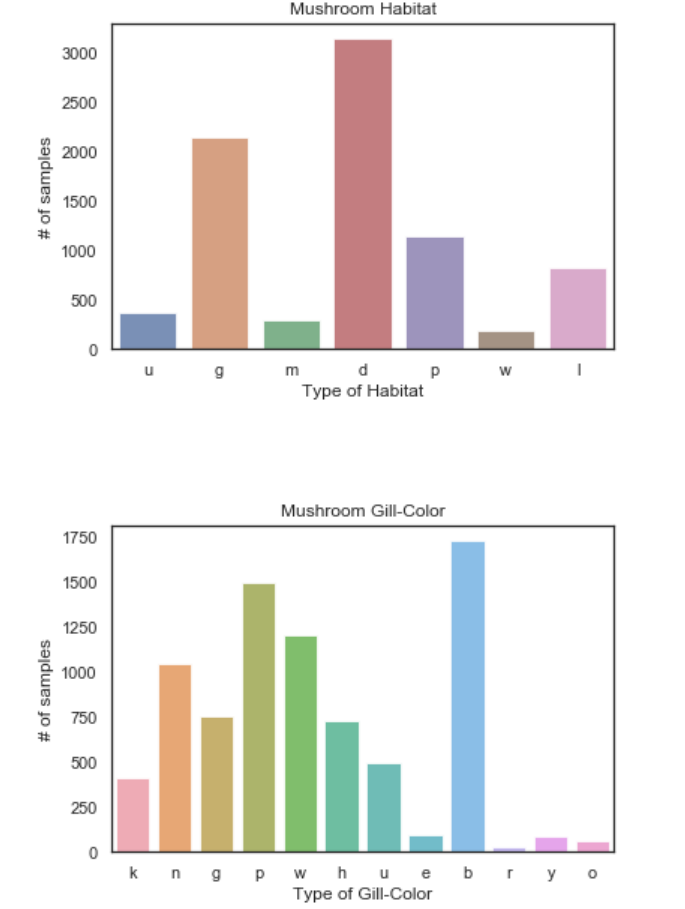
Here is the output



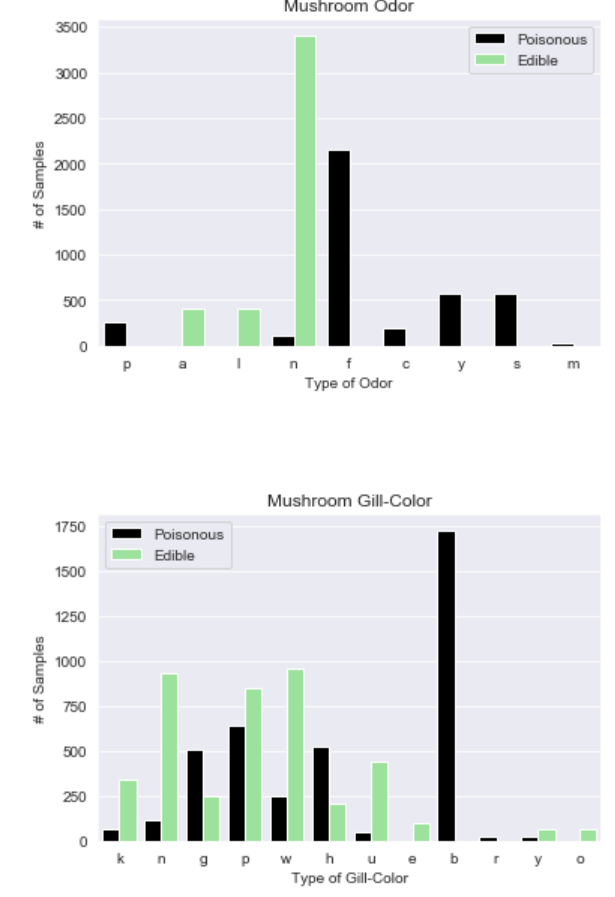
Now we see that all the column values are converted to ordinal and there are no categorical values left.

The **explanatory variables** covered a range of descriptive and visual characteristics on the structure of each observed mushroom — such as, cap color, odor, ring number and stalk shape. There were originally 21 explanatory variables in my dataset, all were categorical. A few feature visualizations are plotted with Seaborn below

By using df.head we will get the first 5 rows of the dataset.



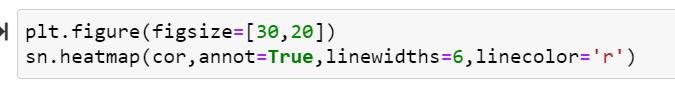
I also utilized EDA to visualize the relationship between the features and the classes of the dataset. There were definitely instances that showed a clear separation

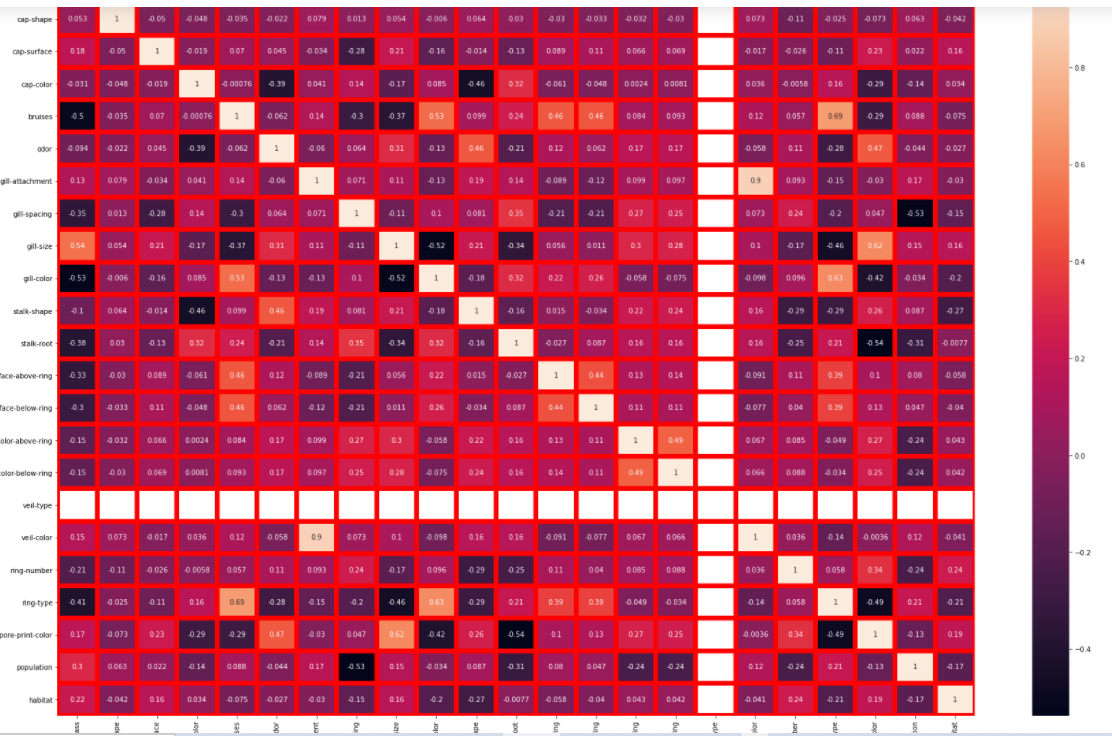


From the above two graph we came to know that type of odor is more edible in n and type of gill color is more poisonous in b.

**Checking the correlation between variables**

****

****

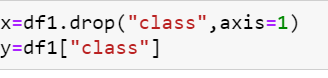
****

## Action Taken based on observation:

* We have seen how categorical & continuous data is distributed & correlated with the target variable. Also, I have mentioned the findings below each plot
* gill\_size is more relation with the target and here no multicollinearity occurs. veil-type is not having any correlation with other variables.
* The feature, veil-type, was dropped since all values were the same and would carry no influence in the model.

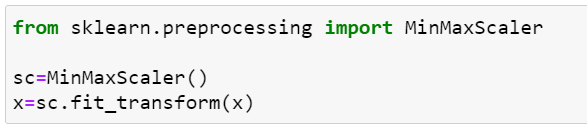
# *Pre-Processing Pipeline*

Dividing Data into x and y

****

Divided the data into two parts features (x) and target (y) for model

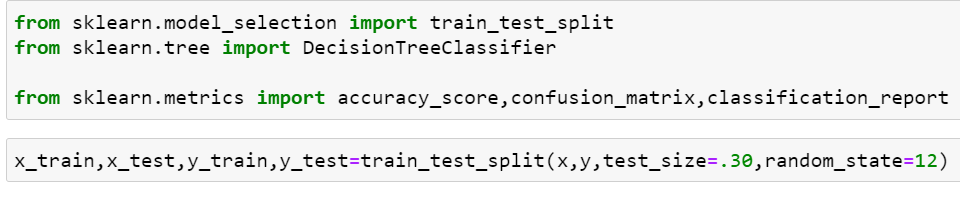
# Scaling



Scaled all the x-data by using the min-max scaler method

# Splitting the Data for Training and Testing

# Find the best random states

****

Splitted the data in the dataset for testing and training to get the best random states

**Modeling**

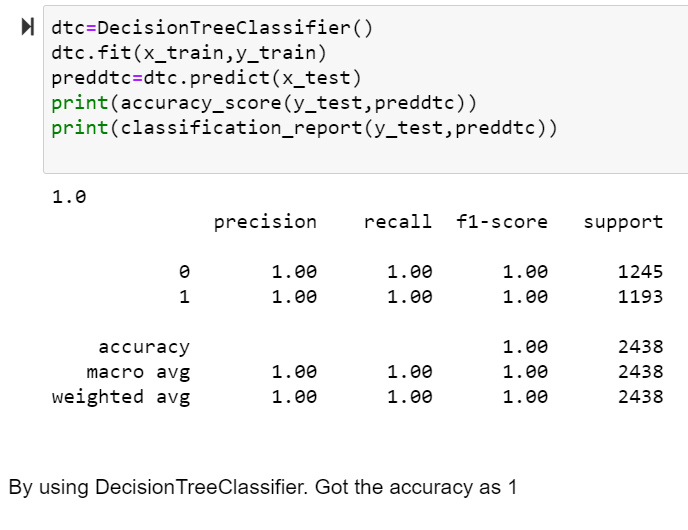
After getting a bit of insight into my dataset through EDA, I dug into the modeling. It should be noted that my dataset was mostly clean from the start — but there were still several areas for cleaning. The main areas of pre-processing my data for modeling were:

* All binary categorical features were encoded to 0 and 1 values (including the target).
* One-hot encoding was applied to all features that were not binary or ordinal/numeric.
* Train/Test split applied to dataset.

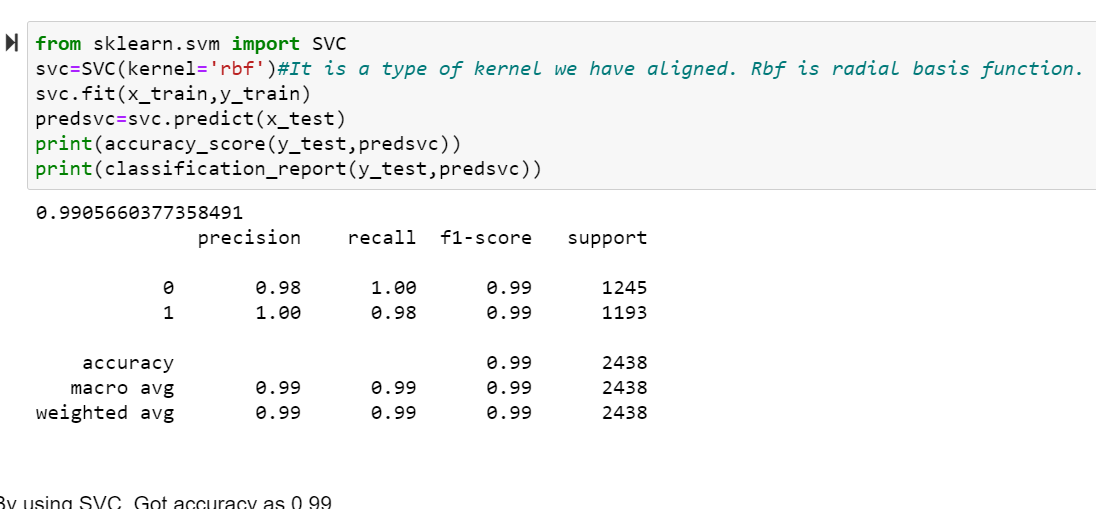
So what models did I run?

I ran a baseline dummy classifier (predicts the majority class),SVC, KNN, Decision Tree and Random Forest Classifier. I utilized grid search to tune the hyper parameters of each (not including the baseline model). My best estimators and their test scores are shown below:

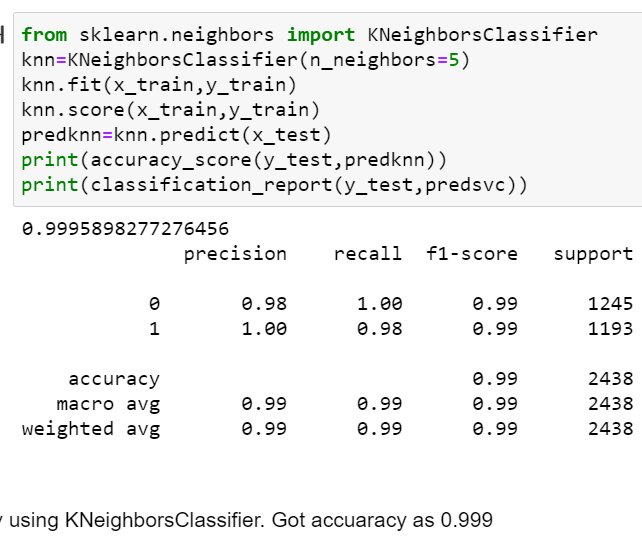
**DecisionTree classifier**

****

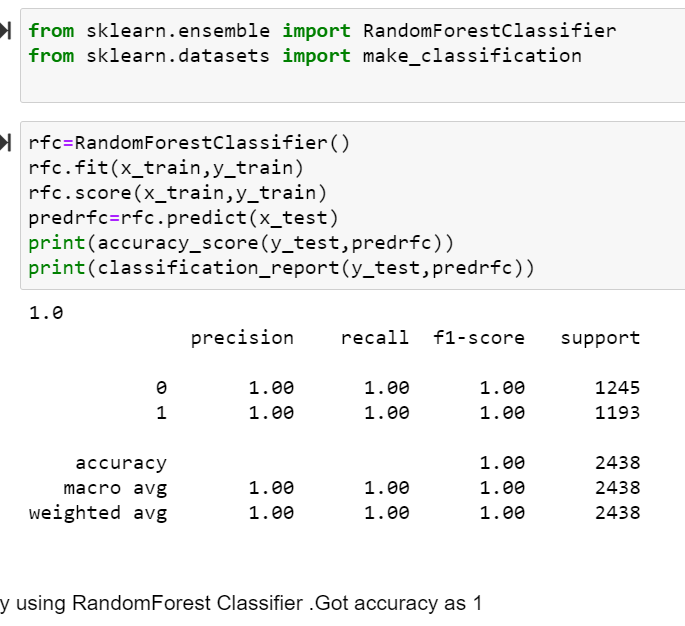
**SVC**

****

**KNN CLASSIFIER**

****

**RandomForest Classifier**

****

# cross validation score

# Observation**:**

For every model, we have train the data (x\_train, y\_train) & predict with the help of x\_test. Now, with the help of the y\_test & prediction value, we got the accuracy score. So, out of all models, we are getting the highest accuracy for some algorithms with less type-I&II error, but it could be due to overfitting, so we need to check the cross-validation if the model is overfitted or not.

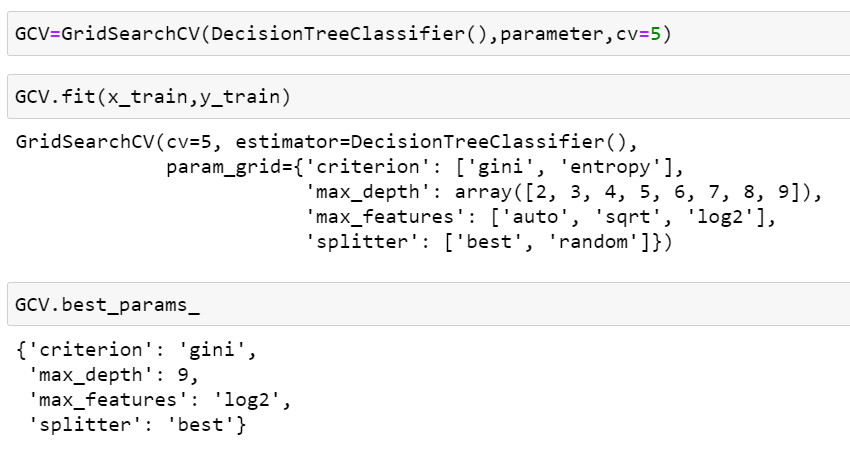
We would perform cross-validation for every model by using **sklearn.model\_selection import cross\_val\_score** & compare with the accuracy score, whichever model gives the less difference between cross-validation score & accuracy score is the best fit model.

|  |  |  |
| --- | --- | --- |
| MODEL | ACCURACY | CROSS VALIDATION |
| KNeighborsClassifier | 100 | 88 |
| DecisionTreeClassifier | 100 | 91 |
| SVC | 99 | 84 |
| RandomForestClassifier | 100 | 90 |

As we have seen, DecisionTreeClassifier,RandomForest classifier is giving less difference as compare to other models. dtc, is giving the lowest difference .So we can choose dtc for HyperParameterTunning.

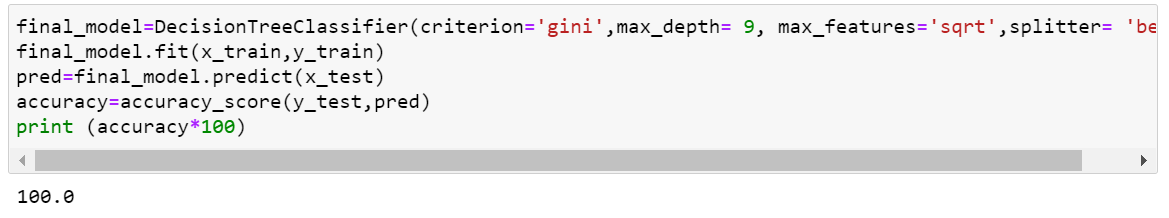
# *HYPER PARAMETER TUNING*:

To enhance the accuracy tuning the model .I have used the **GridSearchCV** grid for tuning the model so we can get the best parameters



After tuning the model we got the best parameters for model

Lets train the model with best parameters

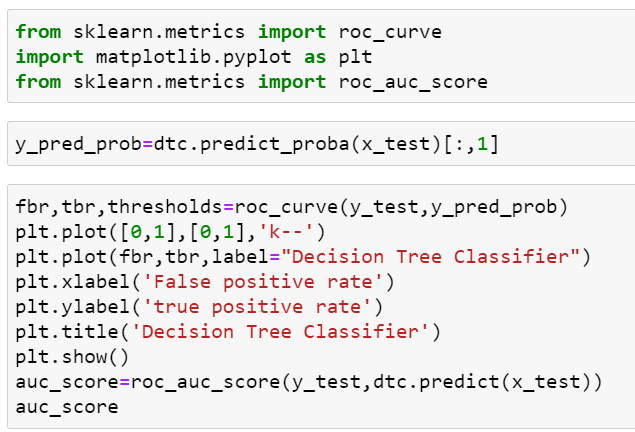
****

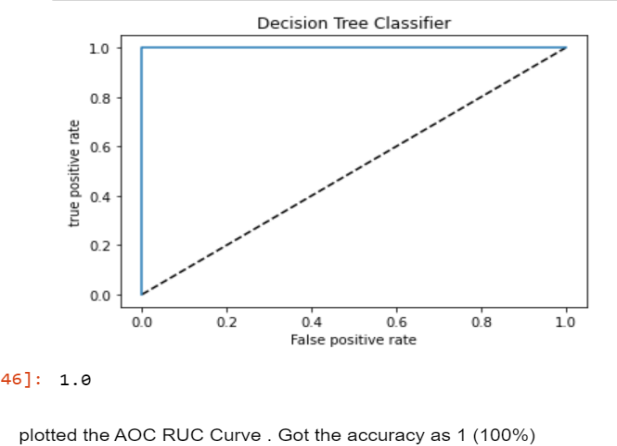
So here got the best model with DecisionTreeClassifier with accuracy 100. We used the parmeters criterion='gini',max\_depth=9, max\_features='sqrt',splitter= 'best

***roc\_auc\_score*:**

An **ROC curve** (**receiver operating characteristic curve**) is a graph showing the performance of a classification model at all classification thresholds. This curve plots two parameters:

* True Positive Rate
* False Positive Rate



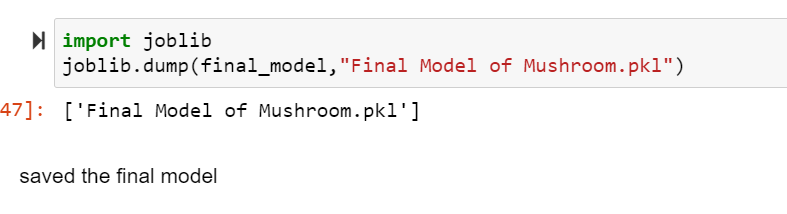


We plotted the roc-auc curve with 100% accuracy. So we can conclude that we got the best model.

# Concluding Remarks

**Saving the model**

Saving the model: The model is ready & we have saved the model in ‘pkl’ format by using “joblib”

*Final conclusion*

Since my models performed so well, it was clear to me that they were able to identify specific traits that greatly influenced the classification of an edible versus poisonous mushroom. And that was exactly what I was hoping for!

To expand on this project and perhaps pick a ‘final’ model, I would like to see if there is similar data available for future analysis. This way I can test my models further and perhaps find some insight into what will work best in a real world situation. A few models performed perfectly on my dataset, so its worth questioning if the dataset was perhaps compromised and therefore, testing my models on another dataset could provide further validation.

Additionally, using my conclusions paired with image classification could also be a worthwhile experiment.

I was able to draw some conclusions which are worth sharing for the purpose of this project.

1. **Odor**: If the Mushroom has an odor, especially if the odor is not pleasant, it is likely to be poisonous.

**2. Gill Size:** If the gills are narrow, it is likely to be poisonous.

**3. Spore Print Color:** White, red, ‘chocolate’ are likely to be poisonous.

**4. Bruises:** If the mushroom does not have bruises, it is likely to be poisonous.

**5. Stalk Surface Above Ring**: If it appears silky, it is likely to be poisonous.

We can now use our model to decide if an ‘poisonous’ or ‘edible’ mushroom. Evidently, this problem was a cakewalk for our machine learning model. Actually, I was hoping to find a dataset that would put up more of a challenge and allow me to troubleshoot the model, but it was still great for practicing with Scikit-learn. Nevertheless, this was a phenomenal example to showcase the power of machine learning—while experts concluded there is no rule that can be used to determine if these genera of mushrooms are edible, a machine with no domain knowledge figured it out without breaking a sweat!

Nevertheless, **this model serves to assist experts, not replace them**. False negatives in this model may be life-threatening; as such, the final word about consuming a wild mushroom should always come from an expert.