**Mushroom Classification with Machine Learning**

Using Supervised Machine Learning Algorithms to identify if a particular mushroom is edible or poisonous.

**PROBLEM DEFINITION:**



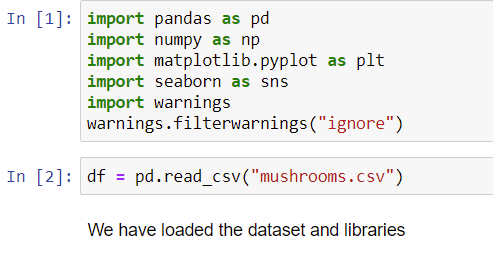
Many people think that mushrooms are vegetables. But this is not the case. All vegetables and fruits come from edible plants. The main characteristic of plants is that they contain chlorophyll, which is used to convert energy from sunlight into carbohydrates. Mushrooms contain no chlorophyll which means they cannot photosynthesize; they steal the carbohydrates they need from plants. Mushrooms are a type of **Fungus** and not all of them are edible.

There are variety of Mushrooms present in nature, this becomes impossible to confirm if the mushroom is edible or poisonous. We will use Machine Learning to help classify the Mushrooms.

**DATA ANALYSIS:**

**Overview of Data**

First, we need to import the necessary libraries and data.



Information of imported libraries:

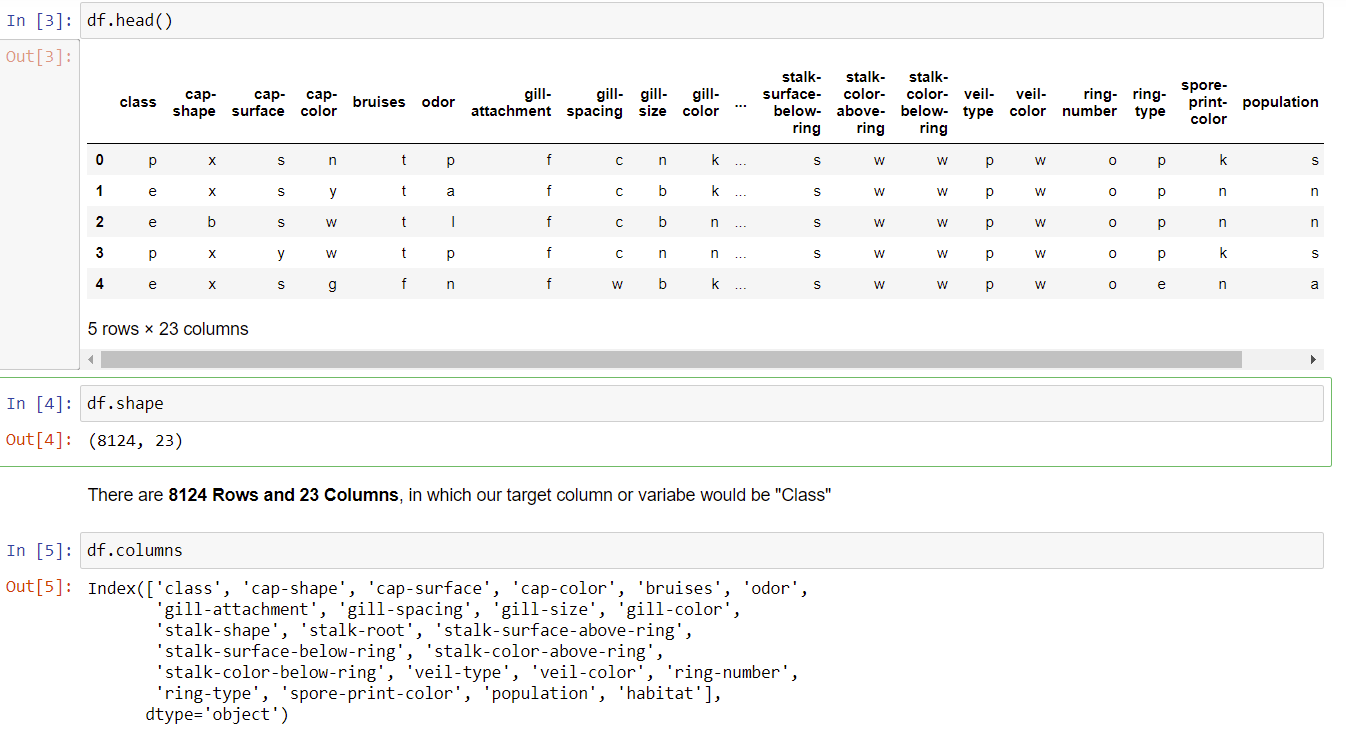
**pandas** to handle tabular data,

**numpy** to support large, multi-dimensional arrays and matrices, help to perform mathematical functions on those arrays.

**Matplotlib** and **seaborn** for data visualizations.

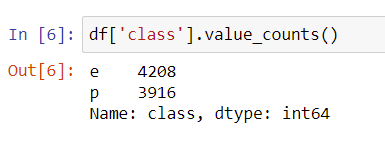
**warnings** help us not show unnecessary warning while we continue working on the project.

Now, we will look at our data.

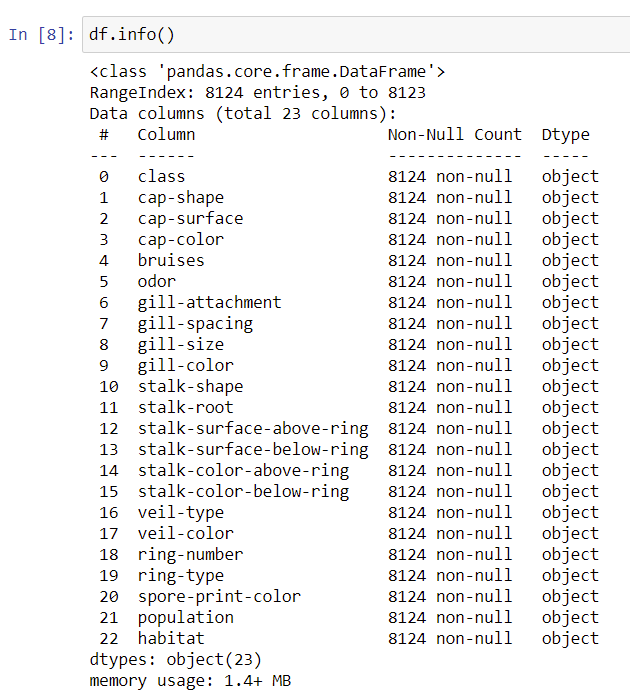


* There are 8124 rows and 23 columns. Columns are also displayed.
* All the feature or columns are object type. “class” would be our target feature.
* We would need to encode all the data into integer data type to pass the data into a Machine Learning Model.

Looking at target feature:



* **e - Edible and p- Poisonous.**
* This would be a **Bi-Classification** problem and by the value counts this would be a **Balanced dataset Problem**.
* We can use **Accuracy Score** while building a model



df.info() function shows that there are no null values in dataset. All the features are Object datatype.

**Preprocessing**

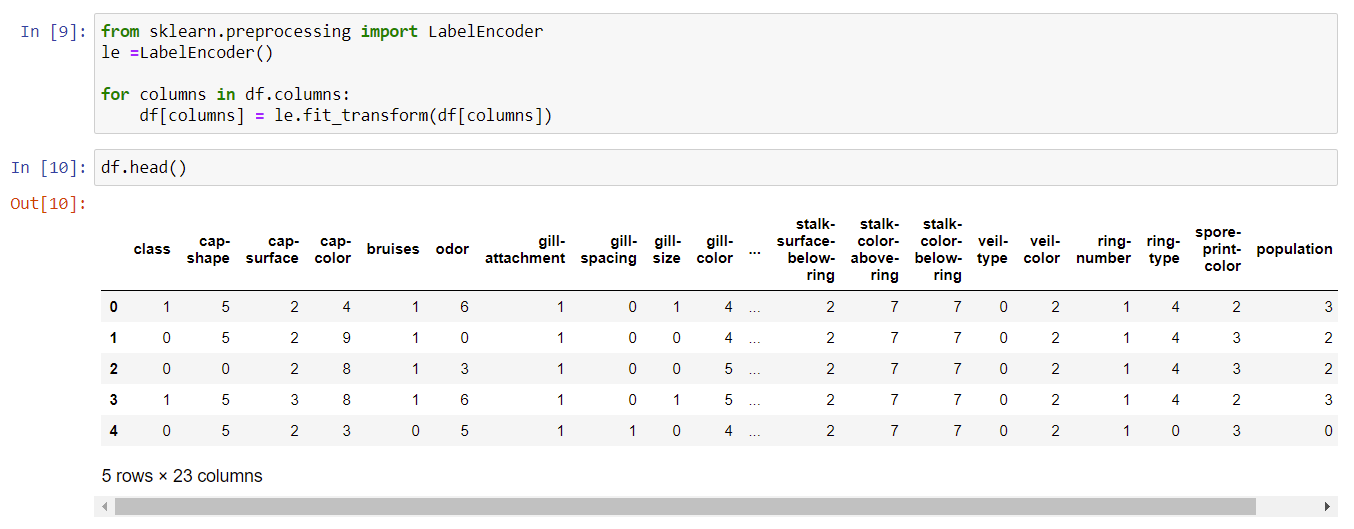
All the columns are categorical and we can use LabelEncoder or OneHotEncoder to encode the values of the columns. LabelEncoder is used for ordinal values and OneHotEncoder for nominal values.

**LabelEncoder**: We will use label encoder to encode the values into integers as we have multiple columns and due to problem requirement to classify into two classes. If you need help with LabelEncoder please follow the link below:

<https://scikit-learn.org/stable/modules/generated/sklearn.preprocessing.LabelEncoder.html>

There are many types of encoding categorical data. To know more please follow below link:

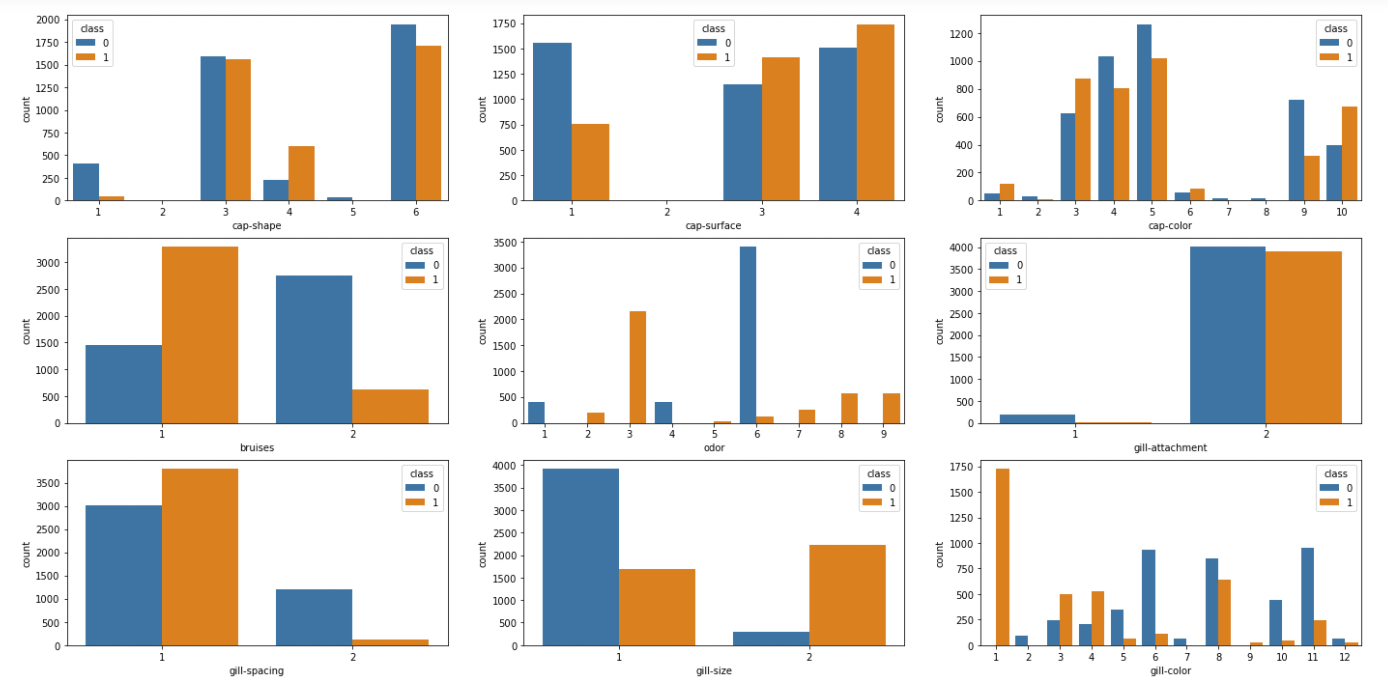
<https://www.analyticsvidhya.com/blog/2020/08/types-of-categorical-data-encoding/>



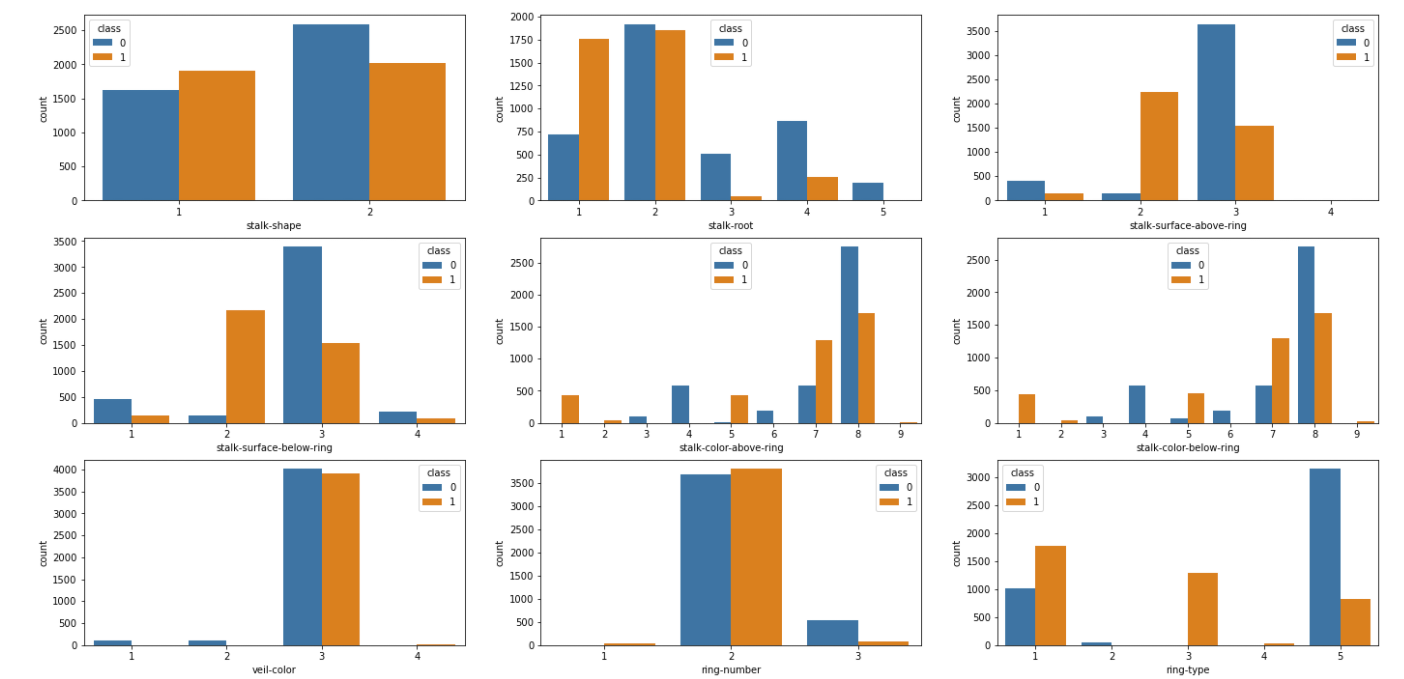
We have transformed the dataset using LabelEncoder where **1 -Poisonous and 0 - Edible.**

**EDA:**

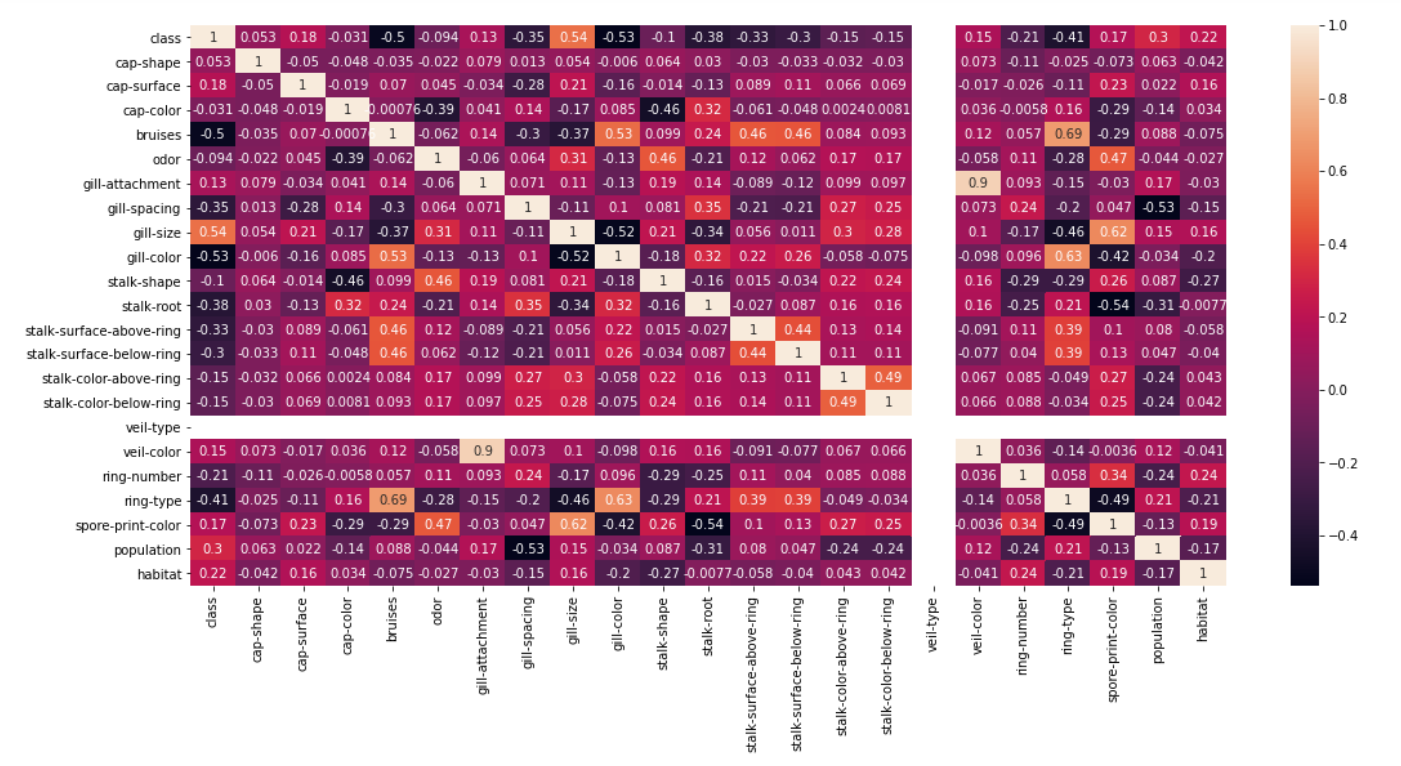
There are 22 columns and we will try to understand the data more by visualizing. We will plot all the columns in countplot with hue as “class”.

****

* Bruises, gill-attachment, gill-spacing and gill-size has only 2 classes.
* There are 8 types of odor for Mushrooms and looking at the plot we can easily say that, only 3 categories (1, 4, 6) are Edible and all the other Categories are Poisonous.
* With **cap-surface** only category 1 has more edible Mushrooms registered, all the other categories have a slightly more influence of Poison Mushrooms.
* With **gill-attachment** Category 1 has less Mushroom values and 2 has the most of the values. Both the categories are slightly influenced by Edible Mushrooms.
* With **gill-spacing:** Two categories for this feature. Category 1 has most of the Mushroom values and influenced by Poison Mushrooms. category 2 has more Edible Mushrooms.
* With **gill-size:** There are only two sizes for a Mushroom in terms of gill-size. Category 1 has most Edible Mushrooms and 2 most poisonous.
* With **gill-color:** This feature or Mushrooms in general have 11 gill colours. In which category 1 is always poisonous.



* With stalk-shape: There only two type of stalk-shape and both the categories seems to have a slight imbalance in terms of value influence.
* With **stalk-root:** This column has 5 categories and category 2 and 3 has most of the Mushrooms.
* With **stalk-surface-above-ring:** There are 4 categories, 2 and 3 have most of the Mushrooms, with 2 influenced by Poisonous Mushrooms and 3 with Edible Mushrooms.
* With **stalk-surface-below-ring:** Same is the case with this column. Both "stalk-surface-above-ring" and "stalk-surface-below-ring" has same trends.
* With **stalk-color-above-ring:** There are 8 categories, in which 3, 4, 6 and 8 are influence by Edible Mushrooms. All the other categories are influence by Poison Mushrooms.
* With **stalk-color-below-ring:** Same is the case here as "stalk-color-above-ring".



We aren't able to make out all the relations easily, however, 'veil-type' has only one category and has no variations all together.



I looked at the correlation of the columns with our target feature and ‘veil-type’ column has only 1 class for poisonous and edible.

We will drop this column as this has no use in model building.

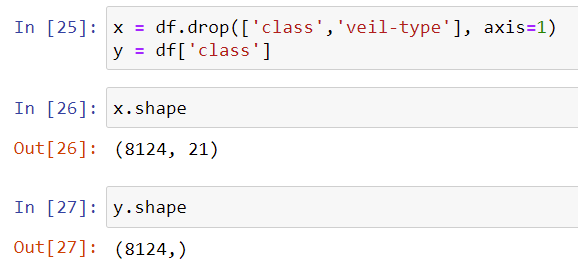
**OUTLIERS AND SKEWNESS:**

As all the columns are categorical, we will not remove outliers and treat skewness. We will drop ‘veil-type’ and move forward with model creation.

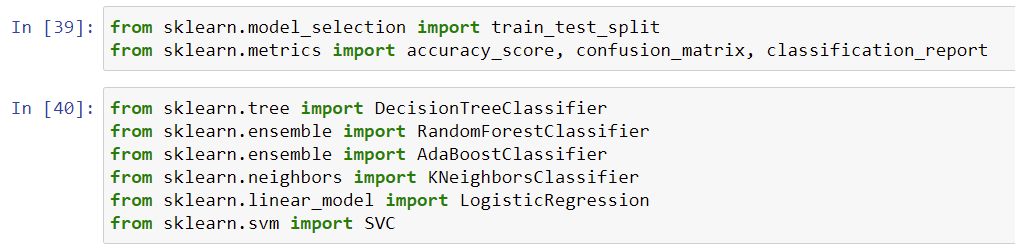
**Pre-processing steps** till now:

1. using labelencoder to encode the values in the data
2. removing ‘veil-type’ columns as it has 1 value for both the target classes.

**BUILDING MACHINE LEARNING MODEL**



We dropped ‘veil-type’ column and separated the independent columns(x) and dependent column (y target feature)

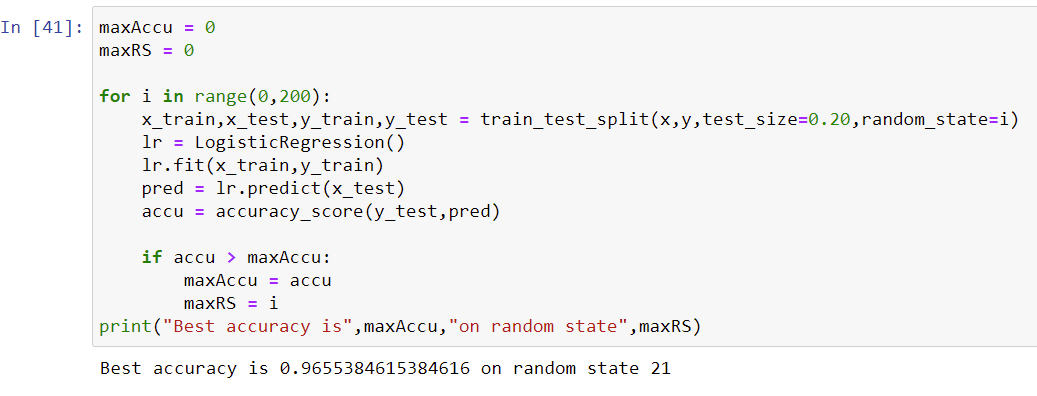
****

We imported:

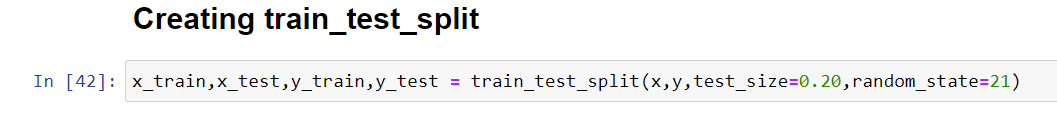
1. train\_test\_split to split our data to train and test.
2. accuracy\_score, confusion\_matrix and classification report to understand our model performance.
3. Different Classifier algorithms.

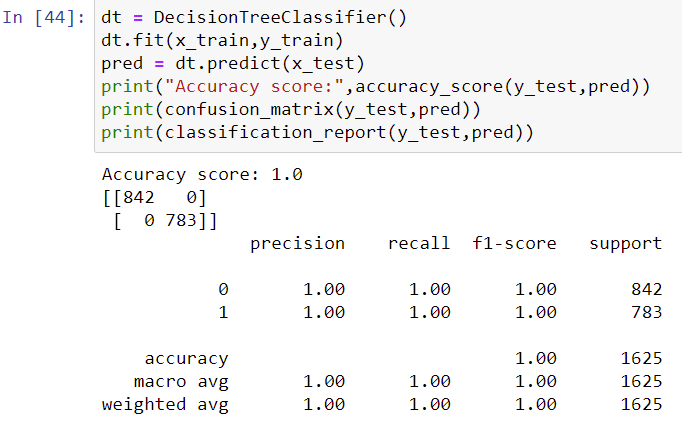
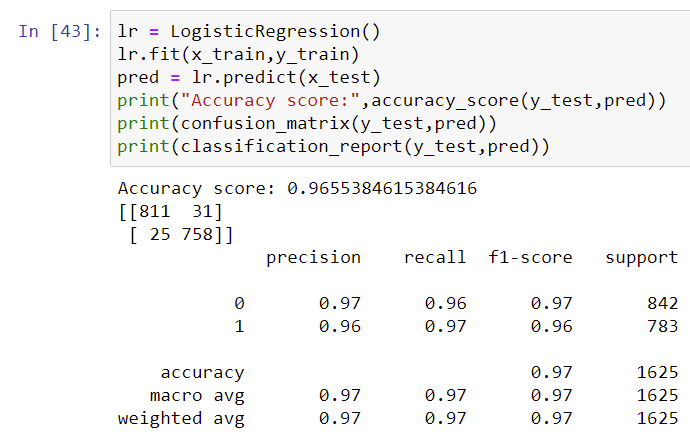
**Finding the best random state for the model**

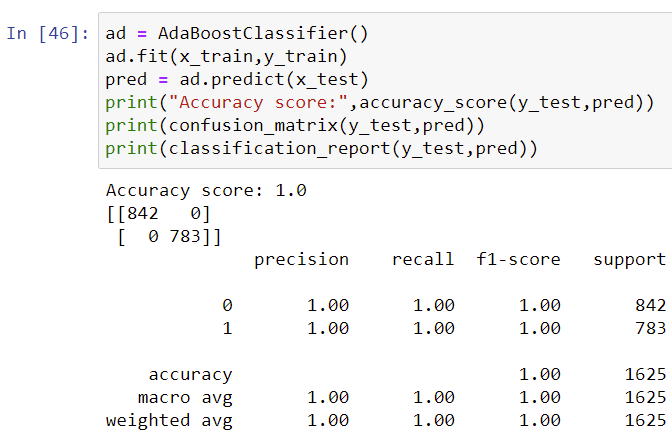
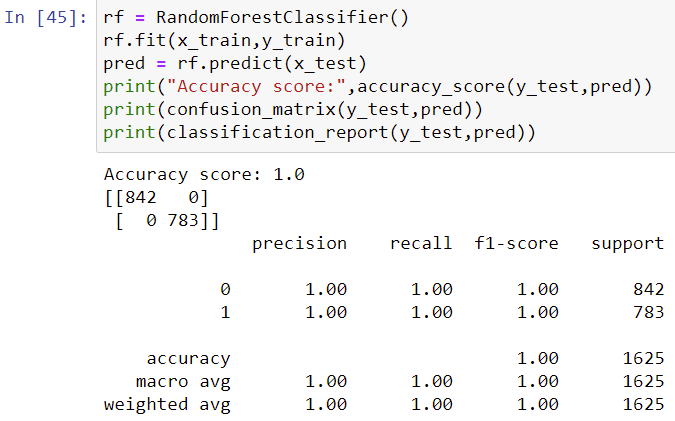
Finding a good random state for our model is important as this may increase our model performance.

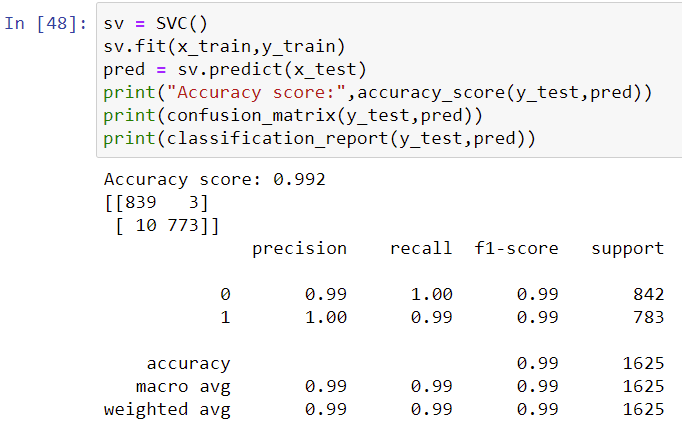
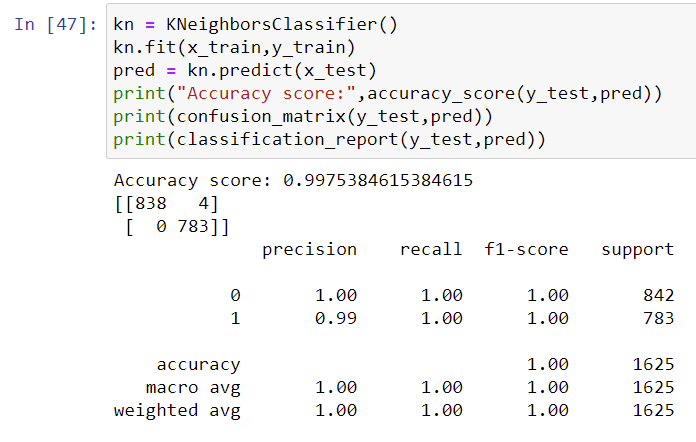


Now we will use 21 as the random state and train all the algorithm we imported earlier.









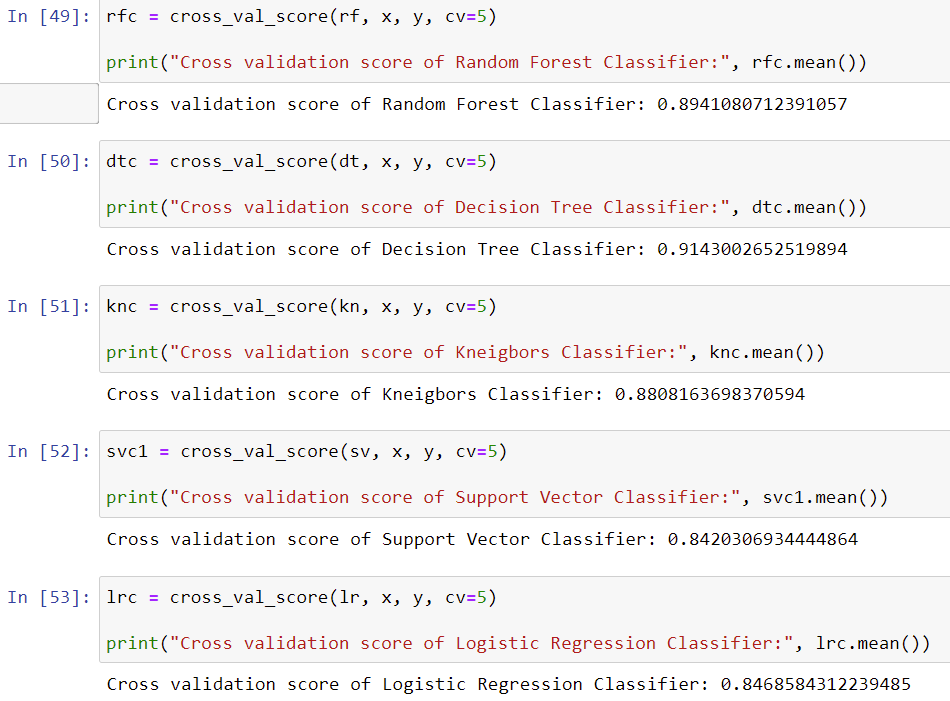
We are getting **100% accuracy** for **RandomForest, DecisionTree, AdaBoost** and with **SVC and Kneighbors 99.20% and 99.75%**. This might be due to Overfitting (Highly unlikely).  
**Logistic regression** is at **96.55%. We will cross validate to make sure we are these results were not due to overfitting.**

**Crossvalidation**

**We will import** cross\_val\_score from sklearn to cross validate our model performances. We will set the cv to 5 so that our model runs 5 times and we will take the mean of the five model performances to understand the results. For more details:

<https://scikit-learn.org/stable/modules/generated/sklearn.model_selection.cross_val_score.html>

****

****

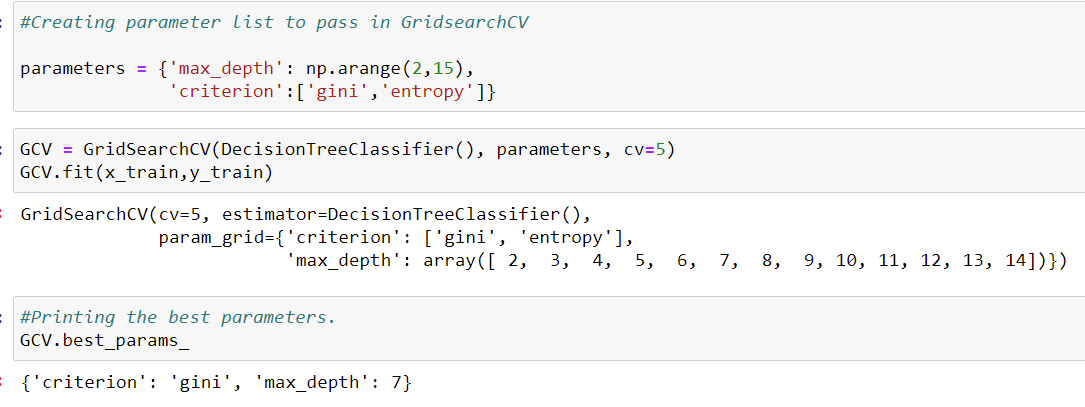
Out of all the Algorithms we tried, **Decision Tree Classifiers** gives us an Accuracy of **90.69%** with a difference of around **10%** with the crossvalidation accuracy. We will choose this classifier and to improve the accuracy we will perform Hyper Parameter Tuning.

**Hyperparameter Tuning**

To do this we will need to import gridsearchcv to perform hyperparameter tuning. To know more please visit:

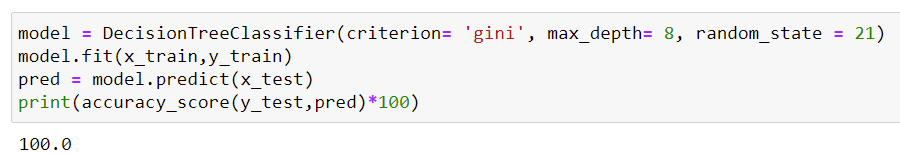
<https://scikit-learn.org/stable/modules/generated/sklearn.model_selection.GridSearchCV.html>



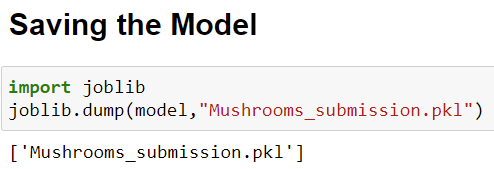


We give in parameters for the Decision Tree Classifier and fit GridSearchCv and retrieve the best parameters for the classifier. Using the output classifier parameters, we will build our final model.

**Model**



We have now created our final model and we can save this file for submission or to classify the mushrooms. We can use joblib to do that.



**CONCLUSION**

We can conclude that veil-type has no importance in our problem. All the features are important and by using Tree Algorithm we can easily create a model which can accurately classify the mushrooms.

I’m amazed by the simplicity and low bar of technical knowledge that is required to get a functioning machine learning model. Machine learning will not only simply and ease our lives but also broaden our scope of understanding.