**Major Project Synopsis**

On

**Mashrooms Classification**

*In partial fulfillment of requirements for the degree*

Of

BACHELOR OF TECHNOLOGY

IN

COMPUTER SCIENCE ENGINEERING

***Submitted by:***

Yashi Naik (20100BTBDAI07226)

Manthan Vyas(20100BTBDAI07216)

Sujal Rathod(20100BTBDAI07221)

Sumit Dubey (20100BTBDAI07222)

Dipanshu Barfa(20100BTBDAI0705)

***Under the guidance of:***

Mr. Om Kant Sharma



**DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING**

**SHRI VAISHNAV INSTITUTE OF INFORMATION TECHNOLOGY**

**SHRI VAISHNAV VIDHYAPEETH VISHVAVIDHYALAYA, INDORE**

**JUL-DEC-2022**

**SHRI VAISHNAV INSTITUTE OF INFORMATION TECHNOLOGY**

**Introduction**

**Mushroom**, the [conspicuous](https://www.merriam-webster.com/dictionary/conspicuous) umbrella-shaped fruiting body (sporophore) of certain fungi, typically of the order [Agaricales](https://www.britannica.com/science/Agaricales) in the phylum [Basidiomycota](https://www.britannica.com/science/Basidiomycota) but also of some other groups. 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 colour, gill colour, etc. using different classified.The dataset used in this project is mushrooms.csv that contains 8124 instances of mushrooms with 23 features like cap-shape, cap-surface, cap-colour, bruises, odour, etc.

**Problem domain**

Mushrooms are edible fungus that can provide several important nutrients. The many kinds of mushroom have varying compositions and nutritional profile but some of them contains poisonous substances so it becomes very important to understand different variants and their properties .

**Objectives**

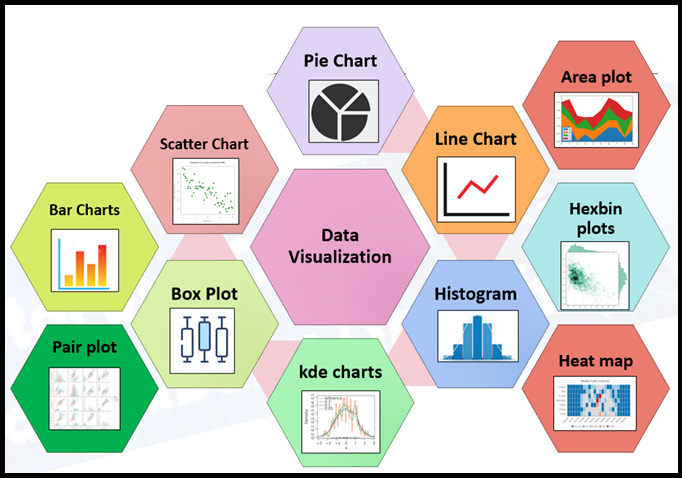
The project involves understanding different types of mushrooms ,here the large scale data is provided about mushrooms therefore we are given two models for greater objectivity and robust analysis of data and, pick other external data sources to add explanatory value and analysing data using various visualizations which will help in discovering various trends in the data, provides a perspective of data, will increase the predictivity accuracy of the model, and puts the data into a visual context.

**Methodology**

In this project we have performed data analysis using data visualizations, data pre-processing and models such as decision tree model and random forest models for predictions of data

**Data analysis** is the process of examining, cleansing, transforming, and modeling data with the objective of extracting useful information for decision-making. It is often used in different domains, such as business, science, and the humanities. Data analysis is of many types but here in this project we have performed predictive type f data analysis.Predictive analysis is used to predict what is likely to happen given the previous data. It makes predictions of future outcomes; however, it is important to note that it is just an estimate. Other factors may be needed to be taken into account, such as industry trends or [macroeconomic developments](https://www.imf.org/en/Publications/Policy-Papers/Issues/2019/12/11/Macroeconomic-Developments-and-Prospects-in-Low-Income-Developing-Countries-2019-48872) in the economy or society as a whole. Following methods are performed-:

**Data visualization**- Data Visualization techniques involve the generation of graphical or pictorial representation of DATA**,**form which leads you to understand the insight of a given data set. This visualisation technique aims to identify the Patterns, Trends, Correlations, and Outliers of data sets.



Data visualization techniques most important part of Data Science, there won’t be any doubt about it. And even in the Data Analytics space as well the Data visualization do a major role.

**In our project we have used catplot and countplot for data visualization.**

**Catplot()** function is used to work on categorical data.

As we know that there are a number of axes-level functions for plotting categorical data in different ways and a figure-level interface, catplot(), that gives unified higher-level access to them.

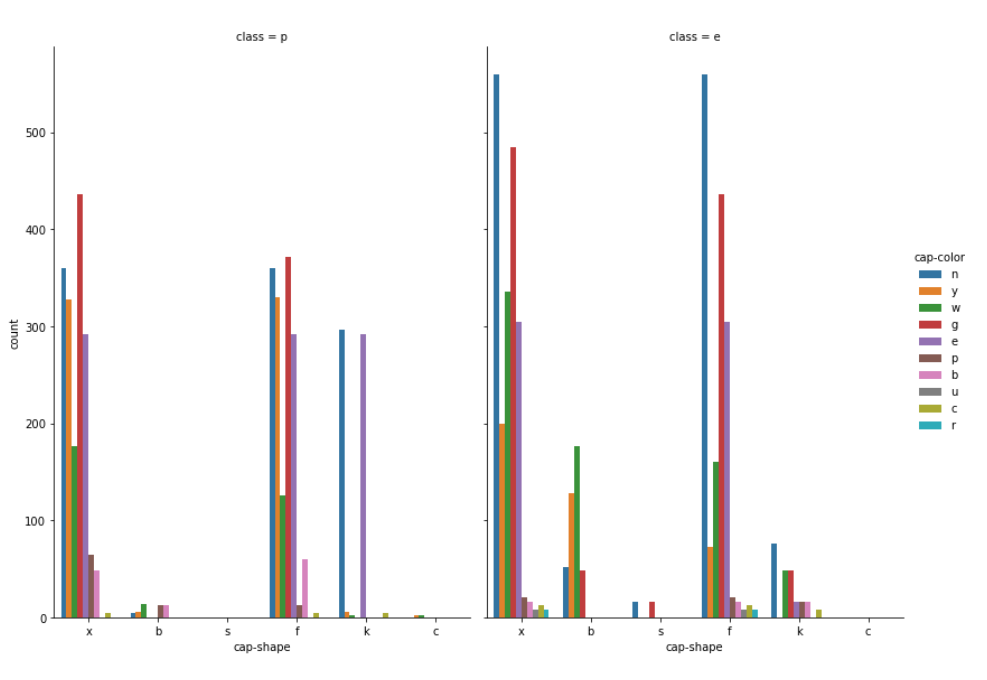
By default for data representation catplot() uses a scatterplot. There are two types of different categorical plots that take different approaches to represent categorical data with a scatter plot in which all the points which belong to the same category will fall in the same position along the axis corresponding to the categorical variable.

**Syntax**

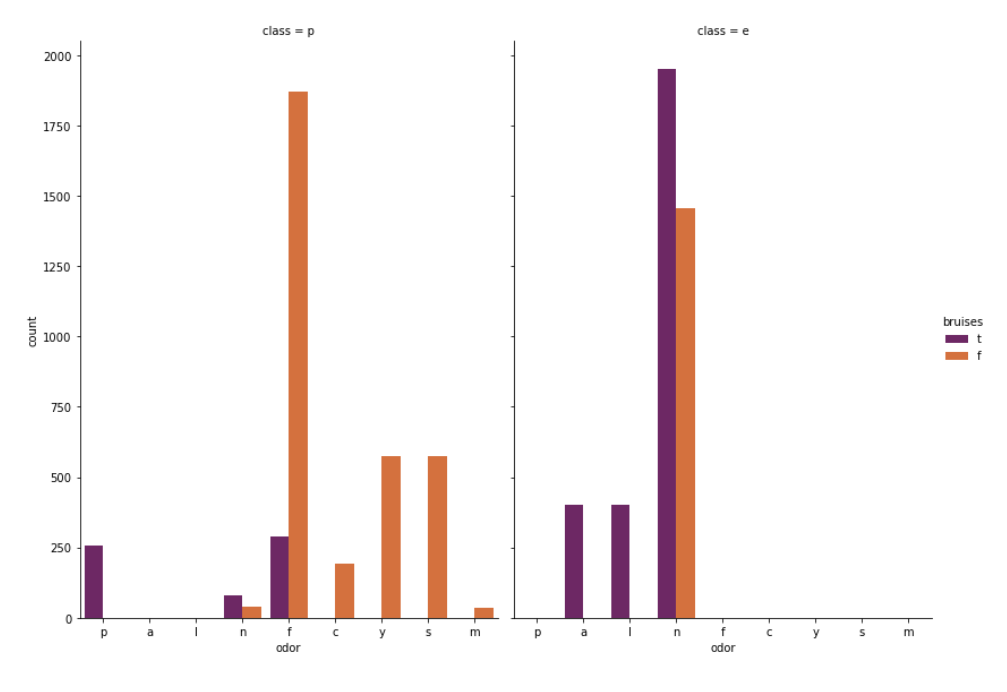
seaborn.catplot(\*, x=None, y=None, hue=None, data=None, row=None, col=None, kind=’strip’,

color=None, palette=None, \*\*kwargs)









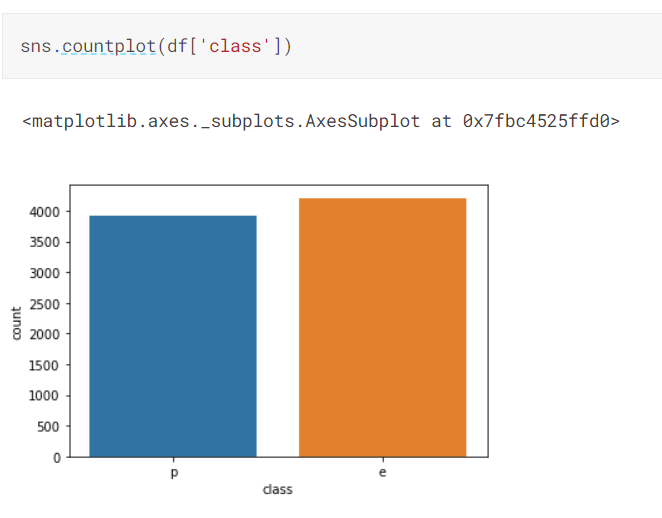
**seaborn.countplot()** method is used to Show the counts of observations in each categorical bin using bars.

**Syntax :** seaborn.countplot(x=None, y=None, hue=None, data=None, order=None, hue\_order=None, orient=None, color=None, palette=None, saturation=0.75, dodge=True, ax=None, \*\*kwargs)  
Parameters : This method is accepting the following parameters that are described below: 

* x, y: This parameter take names of variables in data or vector data, optional, Inputs for plotting long-form data.
* hue : (optional) This parameter take column name for colour encoding.
* data : (optional) This parameter take DataFrame, array, or list of arrays, Dataset for plotting. If x and y are absent, this is interpreted as wide-form. Otherwise it is expected to be long-form.
* order, hue\_order : (optional) This parameter take lists of strings. Order to plot the categorical levels in, otherwise the levels are inferred from the data objects.
* orient : (optional)This parameter take “v” | “h”, Orientation of the plot (vertical or horizontal). This is usually inferred from the dtype of the input variables but can be used to specify when the “categorical” variable is a numeric or when plotting wide-form data.
* color : (optional) This parameter take matplotlib color, Color for all of the elements, or seed for a gradient palette.
* palette : (optional) This parameter take palette name, list, or dict, Colors to use for the different levels of the hue variable. Should be something that can be interpreted by color\_palette(), or a dictionary mapping hue levels to matplotlib colors.
* saturation : (optional) This parameter take float value, Proportion of the original saturation to draw colors at. Large patches often look better with slightly desaturated colors, but set this to 1 if you want the plot colors to perfectly match the input color spec.
* dodge : (optional) This parameter take bool value, When hue nesting is used, whether elements should be shifted along the categorical axis.
* ax : (optional) This parameter take matplotlib Axes, Axes object to draw the plot onto, otherwise uses the current Axes.
* kwargs : This parameter take key, value mappings, Other keyword arguments are passed through to matplotlib.axes.Axes.bar().

Returns: Returns the Axes object with the plot drawn onto it.





**Data pre-processing** is a process of preparing the raw data and making it suitable for a machine learning model. It is the first and crucial step while creating a machine learning model.When creating a machine learning project, it is not always a case that we come across the clean and formatted data. And while doing any operation with data, it is mandatory to clean it and put in a formatted way. So for this, we use data pre-processing task.

It involves below steps:

* **Getting the dataset**
* **Importing libraries**
* **Importing datasets**
* **Encoding Categorical Data**
* **Splitting dataset into training and test set**
* **Checking accuracy**

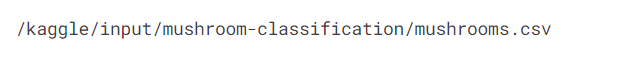
1) Get the Dataset

To create a machine learning model, the first thing we required is a dataset as a machine learning model completely works on data. The collected data for a particular problem in a proper format is known as the **dataset**.

Dataset may be of different formats for different purposes, such as, if we want to create a machine learning model for business purpose, then dataset will be different with the dataset required for a liver patient. So each dataset is different from another dataset. To use the dataset in our code, we usually put it into a CSV **file**. However, sometimes, we may also need to use an HTML or xlsx file.

What is a CSV File?

CSV stands for "**Comma-Separated Values**" files; it is a file format which allows us to save the tabular data, such as spreadsheets. It is useful for huge datasets and can use these datasets in programs.



2) Importing Libraries

In order to perform data preprocessing using Python, we need to import some predefined Python libraries. These libraries are used to perform some specific jobs. There are three specific libraries that we will use for data preprocessing, which are:

**Numpy:** Numpy Python library is used for including any type of mathematical operation in the code. It is the fundamental package for scientific calculation in Python. It also supports to add large, multidimensional arrays and matrices. So, in Python, we can import it as:

1. import numpy as nm

Here we have used **nm**, which is a short name for Numpy, and it will be used in the whole program.

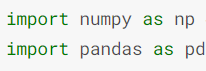
**Matplotlib:** The second library is **matplotlib**, which is a Python 2D plotting library, and with this library, we need to import a sub-library **pyplot**. This library is used to plot any type of charts in Python for the code. It will be imported as below:

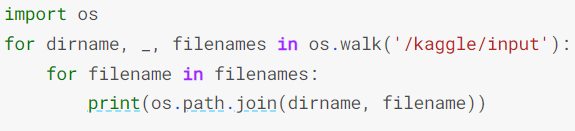
1. import matplotlib.pyplot as mpt

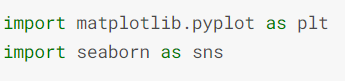
Here we have used mpt as a short name for this library.

**Pandas:** The Pandas library is one of the most famous Python libraries and used for importing and managing the datasets. It is an open-source data manipulation and analysis library. It will be imported as below:

Here, we have used pd as a short name for this library. Consider the below image:







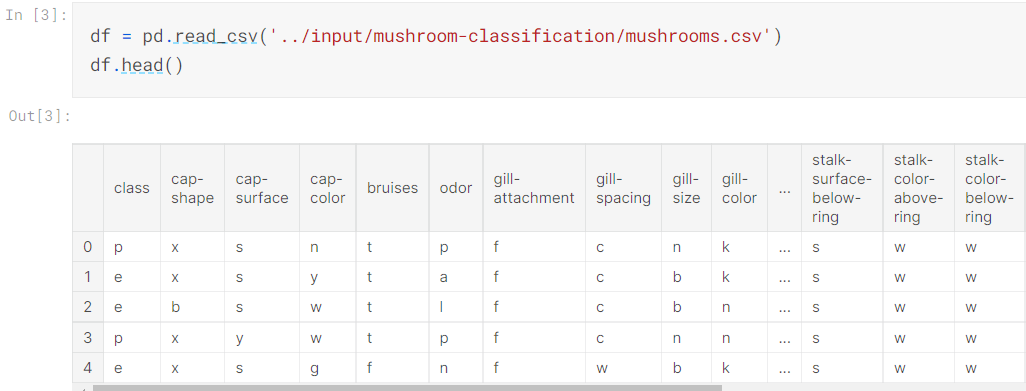
3) Importing the Datasets

Now we need to import the datasets which we have collected for our machine learning project. But before importing a dataset, we need to set the current directory as a working directory. To set a working directory in Spyder IDE, we need to follow the below steps:

1. Save your Python file in the directory which contains dataset.
2. Go to File explorer option in your IDE, and select the required directory.
3. Click on run option to execute the file.

read\_csv() function:

Now to import the dataset, we will use read\_csv() function of pandas library, which is used to read a csv file and performs various operations on it. Using this function, we can read a csv file locally as well as through an URL.



4) Encoding Categorical data:

Categorical data is data which has some categories such as, in our dataset; there are two categorical variable, Country, and Purchased.

Since machine learning model completely works on mathematics and numbers, but if our dataset would have a categorical variable, then it may create trouble while building the model. So it is necessary to encode these categorical variables into numbers.

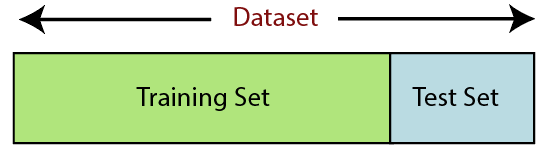


5) Splitting the Dataset into the Training set and Test set

In machine learning data pre-processing, we divide our dataset into a training set and test set. This is one of the crucial steps of data pre-processing as by doing this, we can enhance the performance of our machine learning model.

Suppose, if we have given training to our machine learning model by a dataset and we test it by a completely different dataset. Then, it will create difficulties for our model to understand the correlations between the models.

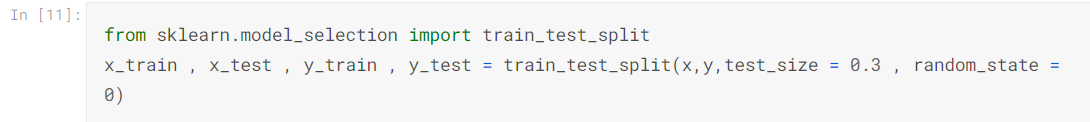
If we train our model very well and its training accuracy is also very high, but we provide a new dataset to it, then it will decrease the performance. So we always try to make a machine learning model which performs well with the training set and also with the test dataset. Here, we can define these datasets as:



Training Set: A subset of dataset to train the machine learning model, and we already know the output.

Test set: A subset of dataset to test the machine learning model, and by using the test set, model predicts the output.

* In the above code, the first line is used for splitting arrays of the dataset into random train and test subsets.
* In the second line, we have used four variables for our output that are
  + x\_train: features for the training data
  + x\_test: features for testing data
  + y\_train: Dependent variables for training data
  + y\_test: Independent variable for testing data
* In train\_test\_split() function, we have passed four parameters in which first two are for arrays of data, and test\_size is for specifying the size of the test set. The test\_size maybe .5, .3, or .2, which tells the dividing ratio of training and testing sets.



6) checking accuracy

Accuracy is checked using different models we have used decision tree model and random forest model

Decision tree model- A decision tree is a flowchart-like structure in which each internal node represents a test on a feature (e.g. whether a coin flip comes up heads or tails) , each leaf node represents a class label (decision taken after computing all features) and branches represent conjunctions of features that lead to those class labels. The paths from root to leaf represent classification rules. Below diagram illustrate the basic flow of decision tree for decision making with labels (Rain(Yes), No Rain(No)). Decision trees are constructed via an algorithmic approach that identifies ways to split a data set based on different conditions. It is one of the most widely used and practical methods for supervised learning. Decision Trees are a [non-parametric](https://machinelearningmastery.com/parametric-and-nonparametric-machine-learning-algorithms/) supervised learning method used for both classification and regression tasks.



Random forest model-

Random Forest is a popular machine learning algorithm that belongs to the supervised learning technique. It can be used for both Classification and Regression problems in ML. It is based on the concept of ensemble learning, which is a process of combining multiple classifiers to solve a complex problem and to improve the performance of the model.

As the name suggests, "Random Forest is a classifier that contains a number of decision trees on various subsets of the given dataset and takes the average to improve the predictive accuracy of that dataset." Instead of relying on one decision tree, the random forest takes the prediction from each tree and based on the majority votes of predictions, and it predicts the final output.

The greater number of trees in the forest leads to higher accuracy and prevents the problem of overfitting.



**Result**

On implementing both the models following results are being found:-

1. Decision Tree Accuracy - 98.2%

2. Random Forest Accuracy - 99%

Hence, Random Forest model is comparitively better than decision tree model.

3.Now analysing and modelling whole dataset - Odourless and Bruised Mushrooms are highly recommended for eating. Fishy Odour with No Bruises Mushrooms are poisonous.

**Conclusion**

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. Therefore here after analysing this data set we get a clearer picture of different classifications of mushrooms