Let's chase Python

You must have come across many blogs and code repositories leveraging the simplified syntax of python along with its vast capabilities. To learn similar python programming skills, let us have a look at some related basic concepts and operations in Python. Follow the links below:

https://www.tutorialspoint.com/python/index.htm
https://www.educba.com/python-programming-beginners-tutorial/

After going through the links above, you will get familiar with the basic syntax of python and will get the answer to the question: **Why is Python so Popular?**

We have summarized the answer to the above question for you.

1. Easy to Learn and Use
Python language is one of the most accessible programming languages available because it has simplified syntax and is not complicated, which gives more emphasis on natural language. Due to its ease of learning and usage, python codes can be easily written and executed much faster than other programming languages.

2. Mature and Supportive Python Community
Python was created more than 30 years ago, which is a lot of time for any community of programming language to grow and mature adequately to support developers ranging from beginner to expert levels. There are plenty of documentation, guides and Video Tutorials for Python language.

3. Big data, Machine Learning and Cloud Computing
Cloud Computing, Machine Learning, and Big Data are some of the trending topics in the computer science world right now, which helps lots of organizations to transform and improve their processes and workflows. Hundreds of python libraries are being used in thousands of machine learning projects every day, such as TensorFlow for neural networks and OpenCV for computer vision, etc.

Let us have hands-on practice using python and apply it to solve a problem using machine learning-based solutions.

Before that, we need to know the following things:
What is machine learning (ML)?
Machine learning is about learning to predict something or extracting knowledge from data. ML is a part of artificial intelligence. ML algorithms build a model based on sample data or known as training data and based upon the training data, the algorithm can predict something on new data.

Some examples of daily life applications of ML are given below:

- **Virtual personal assistants**: Siri and Alexa are some popular examples of virtual personal assistants. As the name suggests, they assist in finding information when asked over voice. All you need to do is activate them and ask, “What is my schedule for today?”, “What are the flights from Germany to London” or similar questions.
  Machine learning is an important part of these personal assistants as they collect and refine the information on the basis of your previous involvement with them. Later, this set of data is utilized to render results that are tailored to your preferences.

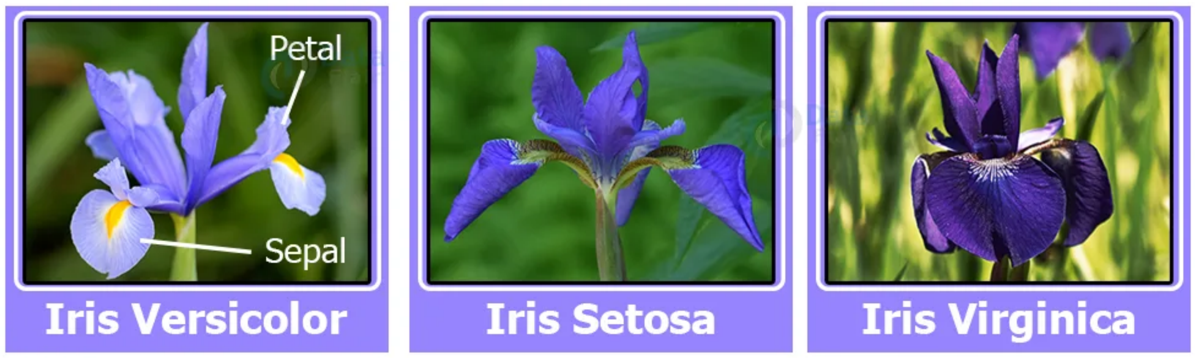
- **Traffic Predictions:** We all have been using GPS navigation services. While we do that, our current locations and velocities are being saved at a central server for managing traffic. This data is then used to build a map of the current traffic. Machine learning in such scenarios helps to estimate the regions where congestion can be found on the basis of daily experiences.

Similar to the concept of the above application examples, we have with us a problem where our task is to predict flower species.

Iris flower species prediction

Here, we will deal with three types of flower species (classes), **Versicolor, Setosa, Virginica,** which can be recognized on the basis of their characteristics. These characteristics include **‘Sepal length’, ‘Sepal width’, ‘Petal length’, ‘Petal width’**

The aim is to predict flowers based on their specific features.



You can download the dataset using this link:
https://drive.google.com/file/d/1RiF1P-tg439Cxfs-h8ugOm1UnL9pMEhu/view?usp=sharing

The dataset consists of 50 samples from each of the three species of Iris.

The basic libraries that you will need for this assignment include:
Numpy: https://cs231n.github.io/python-numpy-tutorial/
https://numpy.org/doc/stable/user/quickstart.html
https://www.w3schools.com/python/numpy/default.asp

Pandas: https://www.w3schools.com/python/pandas/default.asp
Sklearn: https://scikit-learn.org/stable/tutorial/statistical_inference/unsupervised_learning.html
(The link covers the concept required for the assignment. You can explore more over the web if you want.)
Matplotlib: https://www.w3schools.com/python/matplotlib_intro.asp
Seaborn (optional, but don't miss it if you are interested in visualizing your results using elegant plots): https://www.mygreatlearning.com/blog/seaborn-tutorial/

Now let us implement the following interesting algorithm to predict iris flower species.

1. Import libraries

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.cluster import KMeans
from sklearn.metrics import silhouette_score
from sklearn.preprocessing import MinMaxScaler
```

The `as` keyword is used to create an alias.

Are you wondering about the usage of these libraries??
Just wait. You will be using the functionalities associated with these libraries in the following coding steps.

2. Read the dataset

```
iris = pd.read_csv("/content/drive/MyDrive/data_p/Iris.csv")
x = iris.iloc[:, [0, 1, 2, 3]].values
```

`pd.read_csv` Reads a comma-separated values (csv) file into DataFrame. A Pandas DataFrame is a 2-dimensional data structure, like a 2-dimensional array or a table with rows and columns. For more info:
https://pandas.pydata.org/docs/reference/api/pandas.read_csv.html
Great! You have just used the pandas library that you imported with alias “pd”.

iris.iloc The iloc() function in python is one of the functions defined in the Pandas module that helps us to select a specific row or column from the data set. Using the iloc() function in python, we can easily retrieve any particular value from a row or column using index values.
For more info: https://www.geeksforgeeks.org/python-extracting-rows-using-pandas-iloc/
Play with iloc parameters and see what different outputs you get.

```
iris.info()
iris[0:10]
```

The info() method prints information about the DataFrame.
https://pandas.pydata.org/docs/reference/api/pandas.DataFrame.info.html

Do you remember printf in C to print your output on the screen? Here, in python, you will be simply using “print()”. https://www.w3schools.com/python/ref_func_print.asp

3. Check distribution frequency

In this step, we will check the frequency distribution of species. It means that we are going to print the number of samples of each flower species present in the dataset.

```
iris_outcome = pd.crosstab(index=iris["Species"],
                              columns="count")     # Name the count
column
```

```
print(iris_outcome)
```

Note: # is used to write comments as you must have used // in C programming.

pd.crosstab Compute a simple cross-tabulation of two (or more) factors.
By default, it computes a frequency table of the factors unless an array of values and an aggregation function are passed. For more info:
https://www.w3resource.com/pandas/crosstab.php#:~:text=The%20crosstab()%20function%20is,an%20aggregation%20function%20are%20passed.

You can explore more by plotting elegant plots (box plots, scatter plots, etc.) using the seaborn library. You are provided with the link for the related code in the subsequent steps. Just wait till then!

Now that we know about the dataset, we need to design an algorithm to predict flower species. Here, we are going to use an unsupervised algorithm, k-means clustering.

Unsupervised machine learning uses machine learning algorithms to analyze and cluster unlabeled datasets. These algorithms discover hidden patterns or data groupings without the need for human intervention.
Example: Suppose the unsupervised learning algorithm is given an input dataset containing images of different types of cats and dogs. The algorithm is never trained upon the given dataset, which means it does not have any idea about the features of the dataset. The task of the unsupervised learning algorithm is to identify the image features on their own. An unsupervised learning algorithm will perform this task by clustering the image dataset into groups according to similarities between images.

If still confused, let us take a general example of how we humans differentiate. Suppose we are asked to group 10 animals (cats and dogs) on the basis of their similarities, then we can easily group them into two classes without even knowing the different types of dogs and cats. We can simply do it by learning to differentiate between the two categories of animals by learning the similarity features (all cats' ears and sizes will be similar and different from those of dogs).

K-means clustering is a common example of an exclusive clustering method where data points are assigned into K groups, where K represents the number of clusters based on the distance from each group's centroid.

The data points closest to a given centroid will be clustered under the same category. A larger K value will be indicative of smaller groupings with more granularity, whereas a smaller K value will have larger groupings and less granularity.

K-means clustering is commonly used in market segmentation, document clustering, image segmentation, and image compression.

How to Implementing K-Means Clustering?

1. Choose the number of clusters k.
2. Select k random points from the data as centroids.
3. Assign all the points to the closest cluster centroid.
4. Recompute the centroids of newly formed clusters.
5. Repeat steps 3 and 4.

Here, we are using sklearn library to use k-means clustering.

4. **Finding the optimum number of clusters for k-means classification**

```
from sklearn.cluster import KMeans
wcss = []


for i in range(1, 11):
    kmeans = KMeans(n_clusters = i, init = 'k-means++', max_iter =
300, n_init = 10, random_state = 0)
    kmeans.fit(x)
    wcss.append(kmeans.inertia_)
```

5. **Using the elbow method to determine the optimal number of clusters for k-means clustering (https://www.geeksforgeeks.org/elbow-method-for-optimal-value-of-k-in-kmeans/)**

```
plt.plot(range(1, 11), wcss)
plt.title('The elbow method')
plt.xlabel('Number of clusters')
plt.ylabel('WCSS') #within cluster sum of squares
plt.show()
```

6. **Implementing K-means clustering**

```
kmeans = KMeans(n_clusters = 3, init = 'k-means++', max_iter = 300,
n_init = 10, random_state = 0)
y_kmeans = kmeans.fit_predict(x)
```

7. **Visualising the clusters**

```
plt.scatter(x[y_kmeans == 0, 0], x[y_kmeans == 0, 1], s = 100, c =
'purple', label = 'Iris-setosa')
plt.scatter(x[y_kmeans == 1, 0], x[y_kmeans == 1, 1], s = 100, c =
'orange', label = 'Iris-versicolour')
plt.scatter(x[y_kmeans == 2, 0], x[y_kmeans == 2, 1], s = 100, c = 'green',
label = 'Iris-virginica')

#Plotting the centroids of the clusters
```

```
plt.scatter(kmeans.cluster_centers_[:, 0], kmeans.cluster_centers_[:,1], s
= 100, c = 'red', label = 'Centroids')

plt.legend()
```

Do you want to plot a 3D scatter plot??

Use the link below to get access to the above code. Try visualizing the different plots provided in the link.

https://colab.research.google.com/drive/1EA4N_q027SC8a1DdVR1fUpDD5woCxGg4?usp=sharing

Implement the code explained above and also the remaining part of the code given at

https://colab.research.google.com/drive/1EA4N_q027SC8a1DdVR1fUpDD5woCxGg4?usp=sharing