



Indira Gandhi National Open University
School of Computer & Information Sciences

MCSL-228
ARTIFICIAL
INTELLIGENCE
AND
MACHINE LEARNING
LAB

**ARTIFICIAL INTELLIGENCE AND
MACHINE LEARNING LAB**

Lab Sessions on Artificial Intelligence

5

Lab Sessions on Machine Learning

18

PROGRAMME DESIGN COMMITTEE

Prof. (Retd.) S.K. Gupta , IIT, Delhi	Sh. Shashi Bhushan Sharma, Associate Professor, SOCIS, IGNOU
Prof. Ela Kumar, IGDTUW, Delhi	Sh. Akshay Kumar, Associate Professor, SOCIS, IGNOU
Prof. T.V. Vijay Kumar JNU, New Delhi	Dr. P. Venkata Suresh, Associate Professor, SOCIS, IGNOU
Prof. Gayatri Dhingra, GVMITM, Sonipat	Dr. V.V. Subrahmanyam, Associate Professor, SOCIS, IGNOU
Mr. Milind Mahajan, Impressico Business Solutions	Sh. M.P. Mishra, Assistant Professor, SOCIS, IGNOU
	Dr. Sudhansh Sharma, Assistant Professor, SOCIS, IGNOU

COURSE DESIGN COMMITTEE

Prof. T.V. Vijay Kumar, JNU, New Delhi	Sh. Shashi Bhushan Sharma, Associate Professor, SOCIS, IGNOU
Prof. S.Balasundaram, JNU, New Delhi	Sh. Akshay Kumar, Associate Professor, SOCIS, IGNOU
Prof D.P. Vidyarthi, JNU, New Delhi	Dr. P. Venkata Suresh, Associate Professor, SOCIS, IGNOU
Prof. Anjana Gosain, USICT, GGSIPU, New Delhi	Dr. V.V. Subrahmanyam, Associate Professor, SOCIS, IGNOU
Dr. Ayesha Choudhary, JNU, New Delhi	Sh. M.P. Mishra, Assistant Professor, SOCIS, IGNOU
	Dr. Sudhansh Sharma, Assistant Professor, SOCIS, IGNOU

SOCIS FACULTY

Prof. P. Venkata Suresh, Director, SOCIS, IGNOU
Prof. V.V. Subrahmanyam, SOCIS, IGNOU
Dr. Akshay Kumar, Associate Professor, SOCIS, IGNOU
Dr. Naveen Kumar, Associate Professor, SOCIS, IGNOU (on EOL)
Dr. M.P. Mishra, Associate Professor, SOCIS, IGNOU
Dr. Sudhansh Sharma, Assistant Professor, SOCIS, IGNOU
Dr. Manish Kumar, Assistant Professor, SOCIS, IGNOU

PREPARATION TEAM

Dr.Sudhansh Sharma, Asst.Professor SOCIS, IGNOU (Unit Writer)	Prof. Ela Kumar (Content Editor) Department of Computers & Engg. IGDTUW, Delhi
	Prof.Parmod Kumar (Language Editor) SOH, IGNOU, New Delhi

Course Coordinator: Dr.Sudhansh Sharma,

Print Production

Sh. Tilak Raj, Assistant Registrar (Publication), MPDD

, 2022

©Indira Gandhi National Open University, 2022

ISBN-

All rights reserved. No part of this work may be reproduced in any form, by mimeograph or any other means, without permission in writing from the Indira Gandhi National Open University.

Further information on the Indira Gandhi National Open University courses may be obtained from the University's office at Maidan Garhi, New Delhi-110068.

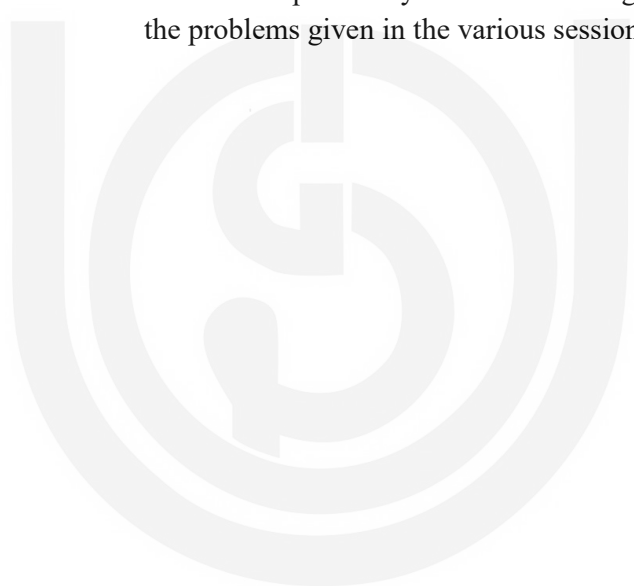
Printed and published on behalf of the Indira Gandhi National Open University, New Delhi by MPDD, IGNOU.

COURSE INTRODUCTION

This course introduces you to practically work with the two most powerful concepts i.e. Artificial Intelligence (AI) and Machine Learning (ML). After completing this course you will be able to perform programming for various techniques of AI and ML, you learned in MCS-224(Artificial intelligence & Machine Learning) course of this programme.

The pre-requisites of the course includes the understanding of Python programming, you may refer to the concepts of Python programming given in the course MCS-201(Programmng in C & Python) and also try to attempt the lab exercises given in the section-2 of MCSL-205(C and Python Lab.).

In order to have better understanding you are advised to firstly get the conceptual clarity of the various algorithms presented in the MCS-224(Artificial intelligence & Machine Learning) course of this programme, and then implement your understanding in Python programming, to solve the problems given in the various sessions of this course.



ignou
THE PEOPLE'S
UNIVERSITY

BLOCK INTRODUCTION

This is the lab course, wherein you will have the hands on experience for programming various algorithms of Artificial Intelligence (AI) and Machine Learning (ML) in Python. You have studied the concepts of both Artificial Intelligence (AI) and Machine Learning (ML) in the course (MCS-224 i.e. Artificial Intelligence and Machine Learning).

In Lab sessions on AI, you will implement the concepts of AI learned in MCS 224, by using Python programming

In Lab sessions on Machine Learning, you will implement the concepts of Machine Learning, learned in MCS-224, by using Python programming. You may use Google Colabs or ANACONDA or any other environment for Python programming.

A list of programming problems for both Artificial Intelligence (AI) and Machine Learning (ML) is provided, you need to attempt the problems given in respective sessions.

Please go through the general guidelines and the program documentation guidelines carefully.

This block consists of Lab sessions on AI and Machine learning and they are organized as follows:

- **Lab sessions on AI** : Comprises of 10 sessions where in 20 programming tasks are framed to be practiced in lab
- **Lab sessions on Machine learning** : Comprises of 10 sessions where in 20 programming tasks are framed to be practiced in lab

Happy Programming!!

AI AND MACHINE LEARNING LAB

Structure No.		Page
2.0	Introduction	18
2.1	Objectives	18
2.2	General Guidelines	18
2.3	Python Libraries for AI and Machine Learning	19
2.4	Working with Python	21
2.5	Running Python Programs	24
2.6	List of Lab Assignments – Session wise	25

2.0 INTRODUCTION

This is the lab course, wherein you will have the hands on experience towards various Artificial intelligence & Machine Learning algorithms, You have studied in MCS-224(Artificial intelligence & Machine Learning) course of this programme.

This course explores the implementation of various Artificial intelligence & Machine Learning algorithms, using Python programming under Google Colabs or ANACONDA or any other Python programming environment. A list of programming problems is also provided at the end of each session. Please go through the general guidelines and the program documentation guidelines carefully.

The pre-requisites of the course includes the understanding of Python programming, you may refer to the concepts of Python programming given in the course MCS-201(Programming in C & Python) and also try to attempt the lab exercises given in the section-2 of MCSL-205(C and Python Lab.).

In order to have better understanding you are advised to firstly get the conceptual clarity of the various algorithms presented in the MCS-224(Artificial intelligence & Machine Learning) course of this programme, and then implement your understanding in Python programming, to solve the problems given in the various sessions of this course.

2.1 OBJECTIVES

After completing this lab course you will be able to:

- Understand the Algorithms for Artificial Intelligence (AI), from the perspective of programming
- Understand the Algorithms for Machine Learning(ML), from the perspective of programming
- Understand the real power Python programming

2.2 GENERAL GUIDELINES

- You should attempt all problems/assignments given in the list session wise.
- You may seek assistance in doing the lab exercises from the concerned lab instructor. Since the assignments have credits, the lab instructor is obviously not expected to tell you how to solve these, but you may ask questions concerning the algorithmic implementation of the concepts of AI and ML in Python programming language or any technical problem.

- For each program you should add comments above each function in the code, including the main function. This should also include a description of the function written, the purpose of the function, meaning of the argument used in the function and the meaning of the return value (if any).
- These descriptions should be placed in the comment block immediately above the relevant function source code.
- The comment block above the main function should describe the purpose of the program. Proper comments are to be provide where and when necessary in the programming.
- The program should be interactive, general and properly documented with real Input/ Output data.
- If two or more submissions from different students appear to be of the same origin (i.e. are variants of essentially the same program), none of them will be counted. You are strongly advised not to copy somebody else's work.
- It is your responsibility to create a separate directory to store all the programs, so that nobody else can read or copy.
- Observation book and Lab record are compulsory
- The list of the programs (list of programs given at the end, session-wise) is available to you in this lab manual. For each session, you must come prepare with the algorithms and the programs written in the Observation Book. You should utilize the lab hours for executing the programs, testing for various desired outputs and enhancements of the programs.
- As soon as you have finished a lab exercise, contact one of the lab instructor / in-charge in order to get the exercise evaluated and also get the signature from him/her on the Observation book.
- Completed lab assignments should be submitted in the form of a Lab Record in which you have to write the algorithm, program code along with comments and output for various inputs given.
- The total no. of lab sessions (3 hours each) are 20 and the list of assignments is provided session-wise. It is important to observe the deadline given for each assignment.

2.3 PYTHON LIBRARIES FOR AI AND MACHINE LEARNING

Python programming is widely used in Artificial Intelligence, Machine Learning, Neural Networks and many other advanced fields of Computer Science. Ideally, It is designed for rapid prototyping of complex

applications. Python has interfaces with various Operating system calls and libraries, which are extensible to C, C++ or Java. Many large companies like NASA, Google, YouTube, Bit Torrent, etc. uses the Python programming language for the execution of their valuable projects.

We learned from the comparison of C and Python, given in MCS – 201 (Programming in C & Python), that Python is a high-level, general-purpose, interpreted high level programming language. It is dynamically typed and garbage-collected, and supports multiple programming paradigms like structured (particularly, procedural,) object-oriented, and functional programming, and due to its comprehensive standard library Python is often described as a "batteries included" language

Python is the most popular programming language, and it marked its presence by contributing actively to every emerging field in computer science. It has vast set of libraries for almost all fields such as Machine Learning (Numpy, Pandas, Matplotlib), Artificial intelligence (Pytorch, TensorFlow), and Game development (Pygame,Pyglet), and many more.

We are already aware that the development of different type of applications require different types of packages, libraries, modules, frameworks etc. Now you might be confused that what is the difference between these terms, are they same or different. Let's Clear your confusion first and then we will briefly discuss about the functionality of different packages, used for the development of various Python applications.

Python: Framework Vs Library Vs Package Vs Module

Framework is a collection of libraries. This is the architecture of the program.

Library is a collection of packages. (*We may understand that, Python library or framework is a pre-written program that is ready to use on common coding tasks.*)

Package is a collection of modules. It must contains an `__init__.py` file as a flag so that the python interpreter processes it as such. The `__init__.py` could be an empty file without causing issues. A package, in essence, is like a directory holding subpackages and modules. While we can create our own packages, we can also use one from the Python Package Index (PyPI) to use for our projects. To import a package, we type `import Game.Sound.load` Or We can also import it giving it an alias: `import Game.Sound.load as load game`.

Module is a file which contains various Python functions and global variables. It is simply just .py extension file which has python executable code. We put similar code together in one module. This helps us modularize our code, and make it much easier to deal with. And not only that, a module grants us reusability. With a module, we don't need to write the same code again for a new project that we take up.

let's see how Module and Package differ:

- 1) A module is a file containing Python code. A package, however, is like a directory that holds sub-packages and modules.
- 2) A package must hold the file `__init__.py`. This does not apply to modules.
- 3) To import everything from a module, we use the wildcard `*`. But this does not work with packages.

We will learn more about them, as we proceed in this course, don't worry. To start with we will discuss about various Frameworks and Libraries first, you will be learning about their usage and also the usage of methods and packages, later.

An introduction towards the functionality of some of the libraries, which are frequently used in the areas of Artificial Intelligence and Machine Learning, is given below :

- TensorFlow: TensorFlow is an end-to-end python machine learning library for performing high-end numerical computations. TensorFlow can handle deep neural networks for image recognition, handwritten digit classification, recurrent neural networks, NLP (Natural Language Processing)
- Keras : is a leading open-source Python library used for development of neural networks and machine learning projects. It simplifies the process of designing and development of neural networks for the beginners of machine learning. Keras also deals with convolution neural networks(CNN) and Recurrent Neural Networks (RNN), highly required in the field of Deep Learning. It includes algorithms for normalization, optimization, and activation layers. Instead of being an end-to-end Python machine learning library, Keras acts as a user-friendly, extensible interface that enhances modularity & total expressiveness.
- Theano : is a library for scientific computation, it allows you to define, optimize as well as evaluate the complex mathematical expressions, which deals with multidimensional arrays. The repetitive computation of a tricky mathematical expression is the basis of several ML and AI applications. Theano quickly performs the data-intensive calculation, the rate of execution is almost hundred times faster than when executing on our CPU alone. It aims to reduce the development and execution time of ML apps, particularly in deep learning algorithms. Only one drawback of

Theano in front of TensorFlow is that its syntax is quite hard for the beginners.

- Scikit-learn : it is a prominent open-source library for machine learning through Python, it includes a wide range of algorithms like DBSCAN, gradient boosting, random forests, vector machines, and k-means etc., which are generally used to implement various concepts of Machine Learning i.e. Classification, Clustering, Regression etc. It can interoperate with numeric and scientific libraries of Python like NumPy and SciPy. Scikit-learn supports both supervised as well as unsupervised ML.
- PyTorch : it is a production-ready library for machine-learning, supported with excellent examples, applications and use cases, also supported by a strong community. It supports GPU and CPU computations, thus provides performance optimization and scalability in research as well as production. The two high-end features of the PyTorch are Deep neural networks and Tensor computation, which are boosted because of the machine learning compiler “Glow”, specially designed to improve the performance of deep learning frameworks.
- NumPy : also known as Numerical Python, a library to perform scientific computations. Almost all Python machine-learning tools like Matplotlib, SciPy, Scikit-learn, etc rely on this library to a reasonable extent. It comes with functions for dealing with complex mathematical operations like linear algebra, Fourier transformation, random number and features that work with matrices and n-arrays in Python. It is widely used in handling sound waves, images, and other binary functions.
- SciPy : It is library, that works for all type of scientific programming projects. Its main functionality is built upon NumPy, and it includes modules for linear algebra, Integration, optimization, and statistics too. SciPy is supported with extensive documentation, which makes its working really easy.
- Pandas : it is an open-source library of python that offers a wide range of tools for data analysis & manipulation, with Pandas, the data from a broad range of sources like CSV, SQL databases, JSON files, and Excel, can be read. The data analysis & manipulation is a pre-requisite for most of the machine learning projects, where a significant amount of time is spent to analyse the trends and patterns hidden in the datasets. Using Pandas one can manage

complex data operations with just one or two commands, it comes with several inbuilt methods for data handling, and it also serves as the starting point to create more focused and powerful data tools.

- Matplotlib : this library is specifically meant for Data Science, it helps to generate data visualization like 2D diagrams and graphs like histograms, scatter plots, and even graphs for non-cartesian coordinates. It is equipped with the object oriented API to embed plots into the applications, directly. It is because of this library, the Python is able to compete with scientific tools like MATLAB or Mathematica. It is quite compatible with the popular plotting libraries, but developers are required to write more code than usual, while using this library for generating advanced visualizations.
- Seaborn : it is an unparalleled visualization library, based on Matplotlib's foundations. Seaborn offers high-level dataset based interface to make amazing statistical graphics. With Seaborn, it is simple to create certain types of plots like time series, heat maps, and violin plots. The functionalities of Seaborn go beyond Pandas and matplotlib with the features to perform statistical estimation at the time of combining data across observations, plotting and visualizing the suitability of statistical models to strengthen dataset patterns.
- Scrapy : It is the most widely used library of python, for the purpose of data sciences, specifically data mining. It is used to build crawling programs i.e spider bots, which are used to retrieve structured data (example URLs or Contact Information) from the web. Developers use it for gathering data from APIs.
- Pygame : is a python programming language library for making multimedia applications like video games, animations etc. It includes libraries related to computer graphics and sound, which are designed to be used with the Python programming language. Pygame is suitable to create client-side applications that can be potentially wrapped in a standalone executable.
- Django : it a web framework build using Python, it encourages rapid development of web applications. Developers can focus on writing app only, It's primary goal is to ease the creation of complex, database-driven websites. Django emphasizes on code

reusability and pluggability of components, thus it supports less code, low coupling, and rapid development of websites.

- MicroPython : is a software implementation of Python 3, written in C. Infact, MicroPython is a Python compiler that runs on the hardware of micro-controller, it includes modules to provide access to the low-level hardware. One of the renowned embedded application is Raspberry Pi which uses Python for its computing. It can be used as a computer or like a simple embedded board to perform high-level computations, you can use it for IoT or Mobile applications, thus may produce your smart gadgets.
- BeeWare – is a Python GUI and mobile development framework, to develop native Python Mobile Apps. “BeeWare” project, provides a set of tools and an abstraction layer, which can be used to write native-looking mobile and desktop applications using Python.
- Kivy – is an opensource Python Framework used as a Cross-platform Python GUI for Mobile App development, it allows you to write pure-Python graphical applications that run on both i.e. the main desktop platforms (Windows, Linux, macOS and Raspberry Pi) and also on on iOS & Android.Both Kivy and BeeWare are worth considering. So far as maturity goes, Kivy seems to be the more mature platform right now

In this course you are given exposure to implement the various algorithms of Artificial intelligence & Machine Learning (You have studied in MCS-224) by using various libraries of Python as programming language.

2.4 WORKING WITH PYTHON

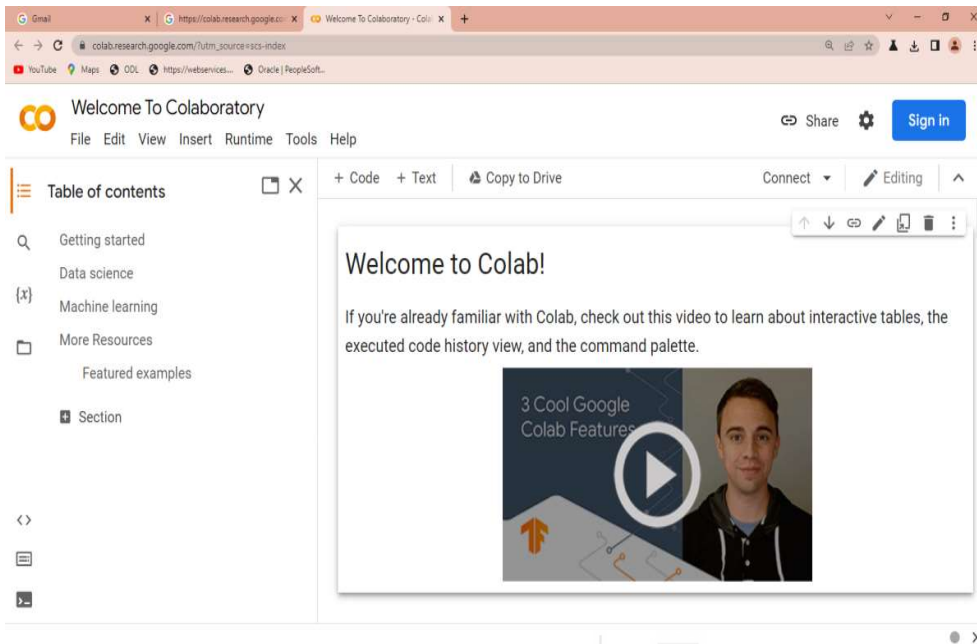
Now, we are having a bit of clarity about the Python as a programming language, from the past sections we learned about the various libraries and frameworks of Python. Now, we need to work on the IDEs (Integrated Development Environments) of Python, there are many IDEs like Jupyter Notebook, Spyder, VS Code, R Programming etc., all are collectively available in Anaconda (Anaconda is a free and open-source distribution of the Python and R programming languages for scientific computing (data science, machine learning applications, large-scale data processing, predictive analytics, etc.)), or you may also go for the cloud versions Like Google Colab Notebook, where you need not to have high configuration hardwares, only internet is required and through your gmail account you can work on Python using Jupyter notebook.

We will discuss in brief, some of the ways to work with Python, you may choose any or try all and other options too.

- 1) Just browse for <https://www.python.org> and perform following steps :
 - a) Download the latest version of Python for the operating system installed on your computer, from <https://www.python.org/downloads/>
 - b) Now Run this exe file and install the Python, just click next and go ahead, till the setup installation is finished
 - c) Finally you are having an interface for Python programming

There is a variety of IDEs for python, Like Jupyter Notebook, Spyder, VS Code etc, and all are available at Anaconda (a free and open-source distribution of the Python and R programming languages for scientific computing (data science, machine learning applications, large-scale data processing, predictive analytics, etc.)). To start with Anaconda Just perform following steps.

- 2) Just browse <https://www.anaconda.com/> and perform following steps:
 - a) Click the Download button on the webpage of <https://www.anaconda.com/>
 - b) the distribution section <https://www.anaconda.com/distribution/> will open,
 - c) click the download option given on this page <https://www.anaconda.com/distribution/>
 - d) It is recommended to download 64bit version of Python 3.x
 - e) Now, just run the setup, and click next-next, till the installation is completed.
 - f) Finally, you will find Anaconda Navigators shortcut on your desktop, click on it and you can start working with any IDE be it Jupyter Notebook, Spyder, VS Code etc., even you can work with R-Programming.
- 3) Many a times the learners may not be equipped with the systems having latest hardware configurations, as desired for the installation of Python, or their might be compatibility issues with operating system or may be due to any reason you are not able to install and start your work with Python. Under such circumstances the solution is Google Colab Notebook (<https://colab.research.google.com/notebooks/welcome.ipynb>), use this and just login with your gmail account and start your work on Jupyter Notebook, as simple as that.



2.5 RUNNING PYTHON PROGRAMS

Just click file option and select new workbook, and new Jupyter notebook will open in Google Colab, now you may start your work.

To begin with just copy paste the code given below, in Google Colab :

(Note : It is a program in Python to understand how Python's Scikit-Learn library can be used to implement the KNN algorithm.)

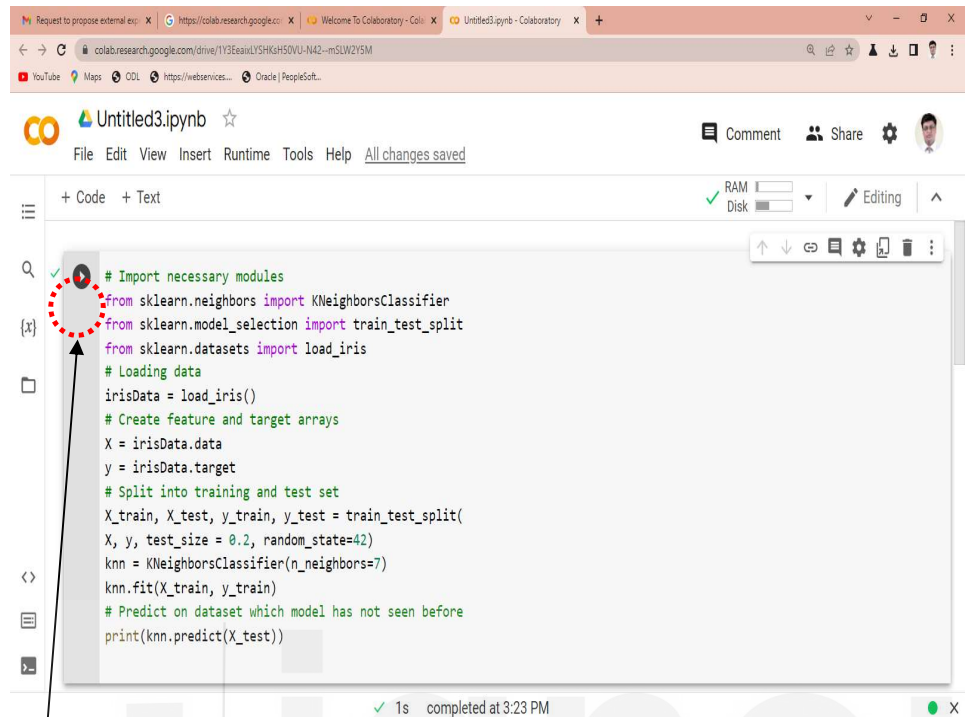
```
# Import necessary modules
from sklearn.neighbors import KNeighborsClassifier
from sklearn.model_selection import train_test_split
from sklearn.datasets import load_iris

# Loading data
irisData = load_iris()

# Create feature and target arrays
X = irisData.data
y = irisData.target

# Split into training and test set
X_train, X_test, y_train, y_test = train_test_split(
    X, y, test_size = 0.2, random_state=42)
knn = KNeighborsClassifier(n_neighbors=7)
knn.fit(X_train, y_train)

# Predict on dataset which model has not seen before
print(knn.predict(X_test))
```



```
# Import necessary modules
from sklearn.neighbors import KNeighborsClassifier
from sklearn.model_selection import train_test_split
from sklearn.datasets import load_iris

# Loading data
irisData = load_iris()

# Create feature and target arrays
X = irisData.data
y = irisData.target

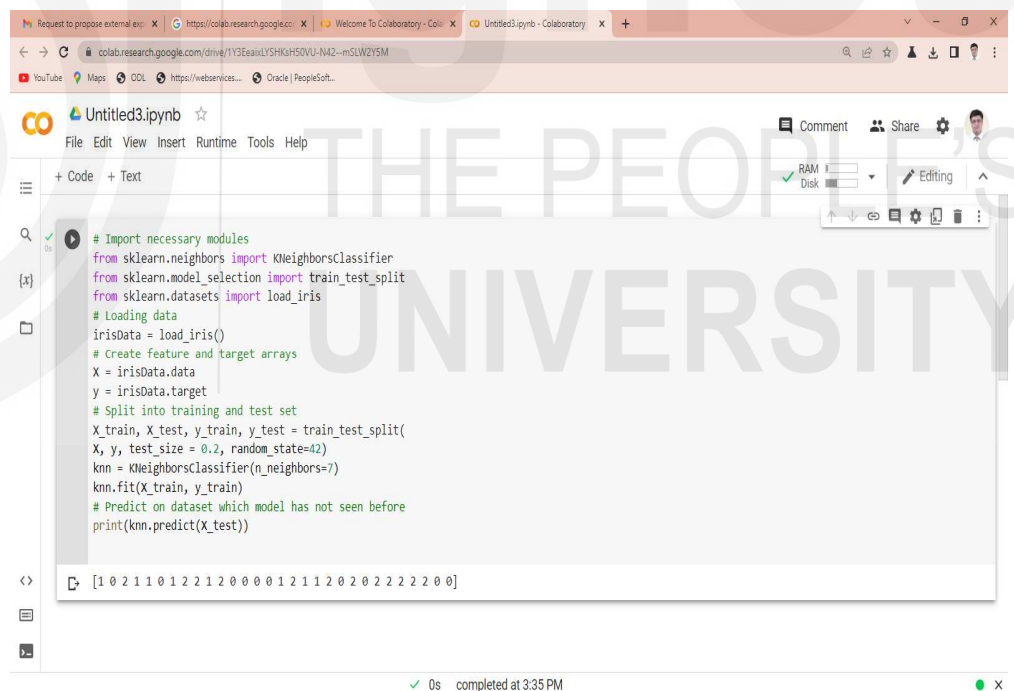
# Split into training and test set
X_train, X_test, y_train, y_test = train_test_split(
    X, y, test_size = 0.2, random_state=42)

knn = KNeighborsClassifier(n_neighbors=7)
knn.fit(X_train, y_train)

# Predict on dataset which model has not seen before
print(knn.predict(X_test))
```

1s completed at 3:23 PM

Execute it by simply pressing the play button, you may see that the output comes just beneath, as shown below:



```
# Import necessary modules
from sklearn.neighbors import KNeighborsClassifier
from sklearn.model_selection import train_test_split
from sklearn.datasets import load_iris

# Loading data
irisData = load_iris()

# Create feature and target arrays
X = irisData.data
y = irisData.target

# Split into training and test set
X_train, X_test, y_train, y_test = train_test_split(
    X, y, test_size = 0.2, random_state=42)

knn = KNeighborsClassifier(n_neighbors=7)
knn.fit(X_train, y_train)

# Predict on dataset which model has not seen before
print(knn.predict(X_test))
```

[1 0 2 1 1 0 1 2 2 1 2 0 0 0 1 2 1 1 2 0 2 0 2 2 2 2 0 0]

0s completed at 3:35 PM

Note :

- You may refer to the python codes given in Unit no. 16 of MCS-224
- You may refer to various links given at section 2.7

2.6 SESSION WISE-LIST OF LAB ASSIGNMENTS

Now, let us try solving a different kind of a AI and Machine Learning problems using Python.

LAB SESSIONS ON AI

***Note :** In each of the problems given in the respective sessions, you are required to write the algorithm for the given problem, and map the steps of your algorithm with your python code, by using suitable comments.*

Session-1

- a) **Write a Python Program to Solve N-Queen Problem without using Recursion.**
- b) Write a Python Program to implement the Backtracking approach to solve N Queen's problem

Session - 2

- a) Write a Python Program to implement Min-Max Algorithm

Session – 3

- a) Write a Python Program to implement Alpha-Beta Pruning Algorithm

Session - 4

- a) Write a Python Program to implement Breadth First Search

Session – 5

- a) Write a Python Program to implement Depth First Search

Session – 6

- a) Write a Python Program to implement Iterative Deepening Depth First search (IDDFS)

Session - 7

- a) Write a Python Program to implement Best First Search

Session - 8

- a) Write a Python Program to implement A* Algorithm

Session - 9

- a) Write a Python Program to implement AO* Algorithm

Session -10

- a) Write a Python Program to implement for IDA* (Iterative Deepening A*) algorithm

LAB SESSIONS ON MACHINE LEARNING

Note : *In each of the problems given in the respective sessions, you are required to write the algorithm for the given problem, and map the steps of your algorithm with your python code, by using suitable comments.*

Also you are required to analyze the results obtained after the implementation of your code on respective dataset, and submit your analysis as a summary to each problem given in respective sessions.

Session-11

- a) Write a Python Program to implement K-Nearest Neighbor Algorithm for data classification , choose dataset of your own choice.
- b) Write a Python Program to implement Naïve Bayes Algorithm for data classification , choose dataset of your own choice

Session – 12

- a) Write a Python Program to implement Decision Trees for data classification , choose data set of your own choice
- b) Write a Python Program to implement Logistic Regression for data classification , choose dataset of your own choice

Session – 13

- a) Write a Python Program to implement Support Vector Machines for data classification , choose dataset of your own choice

Session – 14

- a) Write a Python Program to implement Linear regression on a dataset of your own choice
- b) Write a Python Program to implement Polynomial Regression on a dataset of your own choice

Session – 15

- a) Write a Python Program to implement Support Vector Regression on a dataset of your own choice

Session – 16

- a) Write a Python Program to implement Artificial Neural Network for data classification , choose dataset of your own choice
- b) Write a Python Program to implement Feed Forward Neural Network on a given dataset for data classification , choose dataset of your own choice

Session – 17

- a)** Write a Python Program to implement Principal Component Analysis on a dataset of your own choice
- b)** Write a Python Program to implement Linear Discriminant Analysis on a dataset of your own choice

Session – 18

- a)** Write a Python Program to implement Apriori Algorithm on a dataset of your own choice
- b)** Write a Python Program to implement FP tree growth Algorithm on a dataset of your own choice

Session – 19

- a)** Write a Python Program to implement K-Means Algorithm on a dataset of your own choice

Session -20

- a)** Write a Python Program to implement DBSCAN Algorithm on a dataset of your own choice

2.7 REFERENCES

- <https://www.analyticsvidhya.com>
- <https://www.geeksforgeeks.org>
- <https://www.datacamp.com>
- <https://towardsdatascience.com>
- <https://medium.com>
- <https://machinelearningmastery.com>