PROJECT REPORT ON IRIS-FLOWER CLASSIFICATION (LOGISTIC REGRESSION)

AIML six-week course at NIELIT Ropar

Rakshita
rakshitap1122@gmail.com

Contents

Chapter 1:.....4
AI Ecosystem.....4
1.1. Introduction.....4
1.2. Benefits of Incorporating AI Ecosystems:.....5
1.3 TOOLS of AI:.....6
Chapter 2:.....7
Python Programming.....8
2.1 Introduction.....8
2.2 What is Python?.....8
2.3 Some Key Features of Python:.....8
2.4 Getting Started with Python:.....9
2.5 Python programming with Colab:.....9
Chapter 3:.....12
Numpy.....12
3.1 Introduction.....12
3.2 Benefits of Using NumPy:.....14
Chapter 4:.....15
EDA with Pandas.....15
4.1 Introduction.....15
4.2 Benefits of EDA with Pandas:.....16
Chapter 5:.....17
Machine Learning.....17
5.1 Introduction.....17
5.2 Common Algorithms of Machine Learning:.....17
5.3 Components of Machine Learning:.....18
5.5 Elements of a Typical Machine Learning Project:.....18
5.4 Applications of Machine Learning:.....19
.....19
5.6 Benefits of Machine Learning:.....19
Chapter 6.....21
IRIS-FLOWER CLASSIFICATION (LOGISTIC REGRESSION).....21

6.1 About Project 21
6.3 Pre-Processing Data 21
6.3.1 Import the packages 21
6.3.2 reading the CSV file 22
6.3.3 cleaning the data 22
6.3.4 Visualize the clean data – matplotlib,seaborn 22
6.3.5 Machine learning 23
6.4 Front End 24
6.4.1: Introduction to HTML/CSS/JAVASCRIPT 24
6.4.2 Web server Development using Flask 27
6.4 Project walkthrough 27
References: 28

List of Figures

Figure 1:Ai Ecosystem 4
Figure 2:Benefits of Ai 5
Figure 3:Tools of Ai 6
Figure 4:Features of python language 9
Figure 5:Python Programming 9
Figure 6:first page of colab 10
Figure 7:my first colab code 10
Figure 8:Colab 11
Figure 9:Multidimensional array 12
Figure 10:Indexing and Slicing 13
Figure 11:Benefits of numpy 14
Figure 12:Functions of EDA 15
Figure 13:EDA with pandas 16
Figure 14:Benefits of EDA with pandas 16
Figure 15:Types of machine learning 17
Figure 16:ML 19
Figure 17:Applications of ML 19
Figure 18:Benefits s of ML 20
Figure 19:colab code of ML 20
Figure 20:import the package 21
Figure 21:reading the CSV file 22
Figure 22:cleaning the data 22
Figure 23:visualizing the data using matplotlib 22

Figure 24:visualizing the data using seaborn 23
Figure 25:machine learning 23
Figure 26:tags 24
Figure 27:src attributes 24
Figure 28:href attributes 25
Figure 29:selectors 25
Figure 30:variables 26
Figure 31:my first glitch code 26
Figure 32:web server development using flask 27
Figure 33:FIRST STEPS 27
Figure 34:SECOND STEPS 27
Figure 35:THIRD STEPS 28

Chapter 1:

AI Ecosystem

1.1. Introduction

An AI ecosystem describes a network of various bodies, technologies, and investors that work in coordination for the purpose of developing, deploying, and using AI technologies. It involves a large number of components comprising:

1. Technology Providers: These are companies/ organizations developing AI algorithms, frameworks, and tools, making up such players as Google, OpenAI, Microsoft.

2. Data Providers: Those providing data for the training and improvement of AI models—for example, social media platforms and IoT devices.

3. Research Community: Academic institutions, research organizations, and individual researchers pushing the frontier of AI theory and applications.

4. Government and Regulatory Bodies: Agencies in charge of framing policies, regulations, and guidelines of ethical principles pertaining to the development and deployment of AI.

Figure 1:Ai Ecosystem

5. Enterprises and Businesses: Those organizations using AI in their processes for automation, customer service, decision-making, or any other purposes.

6. Startups and Innovators: Small companies and startups aimed at introducing innovations in AI applications and technologies.

7. Users and Consumers: People and businesses using AI-empowered products and services.

8. AI Ethics and Safety Advocates: Organizations and individuals who concentrate on ensuring that the development and use of AI is responsible and safe.

9. Investors and Funding: The investors, venture capitalists, angel investors are some of the funding agencies that support research and development in AI.

10. AI Talent: Professionals—AI engineers, data scientists, and ethicists—come under the umbrella of an AI ecosystem.

The many components that go into making this highly complex and dynamic ecosystem have much to do with the way in which we see artificial intelligence evolving and having an impact on various sectors like health, financial services, and transportation. Therefore, interactions and collaborations of the components within the ecosystem are critical to advancing AI capabilities, mitigating ethical considerations, and ensuring maximal societal benefits.

1.2. Benefits of Incorporating AI Ecosystems:

Value Addition to a Variety of Domains: AI ecosystems add value to a number of domains in several ways, including the following:

1. Efficiency and Productivity: AI technologies can automate repetitive tasks, optimize processes, and smoothen operations to make them efficient and productive. Put simply, this would enable any organization to shift more resources to strategic activities.

2. Better Decision Making: AI systems can quickly and correctly analyze large volumes of data to provide insights that will support better decision-making. This is particularly pertinent in industries such as finance, healthcare, and manufacturing.

3. Personalization and Customer Experience: AI makes it possible to have personalized recommendations, customer service chatbots, individually tailored marketing campaigns, etc., all of which help improve the overall customer experience.

Figure 2:Benefits of Ai

4. Reduction: AI saves operational costs through labor, energy consumption, resource utilization, etc by automating tasks and speeding up operations.

5.Innovation and New Opportunities: AI thus enables innovation in creating new products and services as well as business models. A startup and traditional players alike are able to explore new markets and opportunities through the help of AI.

6. Predictive Maintenance: AI bringing this function in industries like manufacturing and transport can predict breakdowns of equipment way ahead in time and hence schedule maintenance in advance to reduce downtime and costs in maintenance.

7. Advancements in Health: AI in healthcare is applied to diagnosis, formulation of personal treatment plans, drug discovery, and monitoring patients, helping in getting better outcomes at reduced healthcare expenses.

8. Better Security: AI systems can trace patterns suggestive of potential security threats or anomalies in real-time, improving cybersecurity measures toward the protection of sensitive data.

9. Environment: AI technologies can offer support toward environmental sustainability through energy use optimization, reduction of wastage, and even other initiatives such as smart grid management and precision agriculture.

10. Ethical Considerations: Responsible integration of AI in the ecosystem involves considerations of ethics for fore-fronted fairness, transparency, and accountability of decision-making processes.

1.3 TOOLS of AI:

AI tools are just about any software, framework, library, and other platforms that allow for artificial intelligence solution development, deployment, and usage. The following are some of the key categories with typical examples of AI tools:

1. Machine Learning Frameworks and Libraries:
   a) TensorFlow: An open-source machine learning framework developed by Google, greatly used for building and training neural networks.
   b) PyTorch: Yet another principal open-source machine learning library, primarily developed by the AI Research lab (FAIR) at Facebook. Known for its flexibility and user-friendliness.
   c) Scikit Learn: A simple, general-purpose library for data mining and data analysis built on NumPy, SciPy, and Matplotlib viewed as a tool for machine learning and statistical modeling.
2. Deep Learning Frameworks:
   a) Keras: A Python open-source neural network library which acts as a high-level API to build and train deep learning models.
   b) Caffe: A Berkeley AI Research-built framework for deep learning, especially in image classification and segmentation tasks.
   c) MXNet: A flexible, efficient deep learning framework that supports symbolic and imperative programming with support for multiple languages.

Figure 3:Tools of Ai

3. NLP Tools:
    a. NLTK (Natural Language Toolkit): A suite of libraries and programs for symbolic and statistical NLP for Python.
    b. SpaCy: An open-source library to perform advanced NLP in Python, known for its speed and ease of use.
    c. Transformers (Hugging Face): A library to provide state-of-the-art NLP based on the Transformer architecture, featuring pre-trained models for many different NLP tasks.
4. Computer Vision Tools:
    a. OpenCV (Open Source Computer Vision Library): This is a library of programming functions, mainly related to real-time computer vision. It provides tools for image and video analyses.
    b. YOLO (You Only Look Once): This is a real-time object detection system processing images for the detection and classification of objects.
5. Reinforcement Learning Frameworks:
    a. OpenAI Gym: A toolkit for developing and comparing reinforcement learning algorithms; it provides environments and tools to test and benchmark.
    b. RLlib (Reinforcement Learning Library): An open-source library for reinforcement learning, offering scalable algorithms and environments for research and applications.
6. Data processing and Visualization Tools:
    a. Pandas: A quick, powerful, flexible Python library for data analysis and manipulation.
    b. Matplotlib: Python's plotting package for making static, animated, and interactive visualizations.
    c. Tableau: Data visualization software used to build and share interactive dashboards.
7. AI Development Platforms and API:
    a. Google AI Platform: Cloud-based platform providing tools and APIs to build, train and deploy machine learning scoping models at scale.
    b. Amazon SageMaker: This is a completely managed service recourse, used by developers and data scientists alike, to build, train, and deploy machine learning models rapidly and easily at scale.
    c. Microsoft Azure AI: A set of end-to-end AI services and tools on Microsoft Azure—with cognitive services, machine learning, and AI infrastructure.

Such tools play a very important role in building up AI capabilities across organizations, hence serving developers, researchers, and many organizations with the effective utilization of AI technologies in solving complex problems or creating leading-edge solutions.

Chapter 2:

Python Programming

2.1 Introduction

Python is one of the versatile and very extended programming languages due to its simplicity, readability, and flexibility. It is highly preferred by beginners and professional developers alike because of its easiness to learn and has many potential applications—starting from web development to data analytical domains, artificial intelligence, and scientific computing—many more. Here is a brief introduction to Python programming:

2.2 What is Python?

Python is a high-level, interpreted programming language developed by Guido van Rossum. The first release was in 1991. It has an emphasis on readability and simplicity, thereby possessing an elegant syntax that enables developers to express concepts in fewer lines of code compared to other languages

.

2.3 Some Key Features of Python:

1. Easy to learn: The syntax of Python is transparent and very readable. It serves as an excellent platform for beginners without proportional loss in power and flexibility.

2. Versatility: A Python developer has the choice to work in multiple programming paradigms—procedural, object-oriented, and functional programming styles.

3. Interpreted and Interactive: Python is an interpreted language line by line; hence, the results of the executed code could be seen immediately. Moreover, it supports interactive mode, wherein one can experiment and execute code interactively.

4. Large Standard Library: Python has an extensive standard library that includes modules and packages to almost all problems one can think of, from working with files and networks through data structures to implementing web services.

5. Community and Ecosystem: Active community of developers contributing libraries and frames that extend this language, onto various functionality, from web development with Django and Flask, to scientific computations and machine learning.

Figure 4:Features of python language

2.4 Getting Started with Python:

To develop a program in, one typically needs to:

1. Installing Python: A user has to visit the official website of Python (python.org) to download the version compatible with either Windows, MacOS, or Linux.

2. IDE / Text Editor: Choose on IDE or text editor in which you will be writing and running your Python code. Famous choices include PyCharm, VS Code, and Sublime Text; however, Python can also be run from the command line and scripts directly.

3. Learn the Basics: Know basic Python syntax, data types: integers, floats, strings, lists, dictionaries); control structures: if statements, loops; functions; and modules.

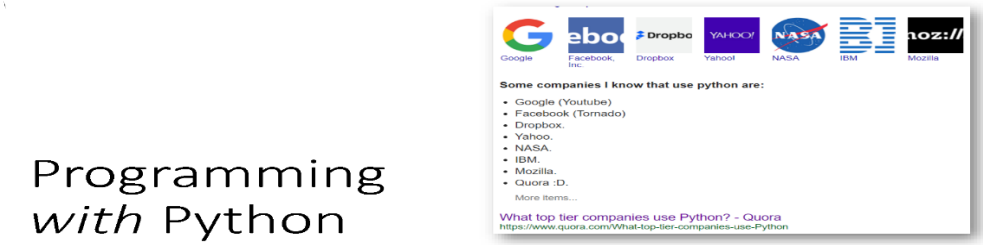

Figure 5:Python Programming

2.5 Python programming with Colab:

Google Colab for Colaboratory is a cloud-based free service from Google for the developing ML and AI. It basically provides an online environment for Jupyter notebooks to run and generate Python code from right there in the user's browser, with no setup in the local environment.

To start using Google colab:

1. Open Google Colab: Visit https://colab.research.google.com/and sign in with your Google account.

2. Create or Upload a Notebook: You can create a new notebook or, alternatively, upload files from your machine or Google Drive.

3. Write and Run Code: Python code can be written in code cells; these are executable interactively. Results are displayed directly in the notebook.

4. Save and Share: Save all your work to Google Drive or GitHub and share the notebook with your team or make it public to share with the world.

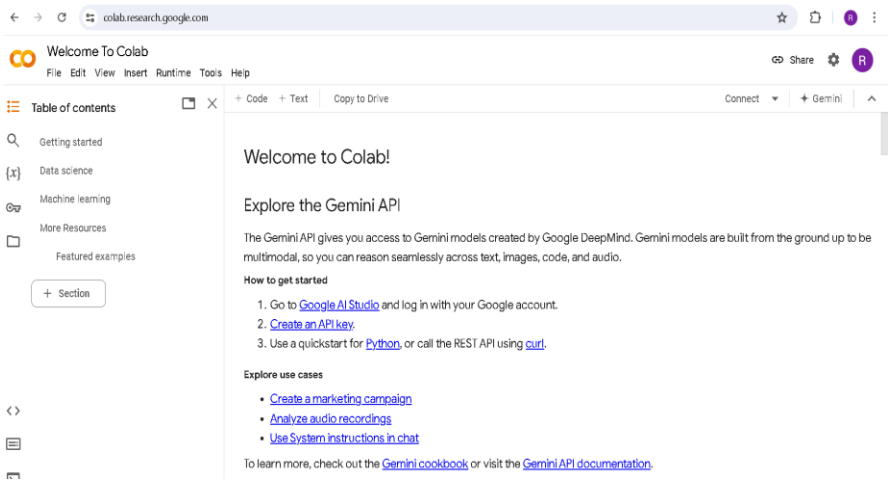

Figure 6:first page of colab

Figure 7:my first colab code

https://colab.research.google.com/drive/1VEZ6foWPbwnUNYLiYDc-VQH-u9q9_KgH?usp=sharing

Following are the main features of Google Colab:

1. Free Access: Google Colab is absolutely offered at no cost. It, therefore, provides support for both CPU and GPU resources which is very useful in the training of machine learning models, going ahead to provide TPUs—Tensor Processing Units.

2. Jupyter Notebook Integration: Colab supports Jupyter notebooks. Users can, therefore, write and execute Python code in individual cells, making it very easy to iterate and experiment among any code.

3. Cloud-Based Execution: Colab runs all computations on Google's cloud servers, so users can utilize as much power from Google's computing and storage resources as needed, while needing only an average, mid-end local hardware.

4. Pre-installed Libraries: Colab comes with most of the standard Python ML/data analysis task libraries pre-installed, which include TensorFlow, PyTorch, Keras, Pandas, NumPy, among many others. Quite helpful while making code for these topics.

5. Collaboration and Sharing: A user can share their notebooks with others using Colab, much like sharing Google Docs. This collaboration option exists for the smooth sharing of findings in research or project work.

6.Google Drive Integration: The level of integration with Google Drive in Colab has been quite remarkable. Users are able to store, share, and open Colab notebooks directly in their Drive account.

7. Interactive Visualizations: Through Matplotlib, Seaborn, and Plotly, the user can produce and explore an unlimited number of data visualizations while still inside their notebooks.

8. Markdown Support: Yes, cells in Colab Notebooks support Markdown, through which the users can include formatted text, images, links, and even LaTeX equations to document their code and analysis.

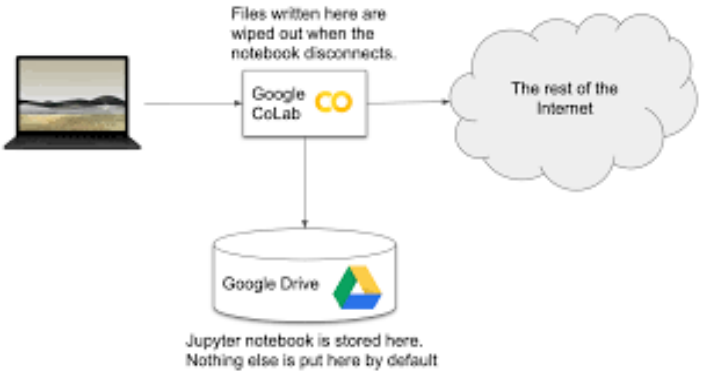

Figure 8:Colab

Over the years, Google Colab has become majorly popular among researchers, data scientists, and machine learning practitioners due to the fact that it is free and easy to use while providing them with very strong computational resources, among other services being offered under the Google infrastructure. A great way to prototype ML models, work on your research project, and cooperate with others, its need for intensive setup and large hardware resources is basically removed as well.

Chapter 3:

Numpy

3.1 Introduction

NumPy, or Numerical Python, is one of the core packets for numerical computing within Python. It provides support for large, multidimensional arrays and matrices, plus a large collection of high-level mathematical functions to operate on these arrays. Thus, NumPy is an intrinsic library in Python for data manipulation and scientific computing. It forms the root of many other libraries and tools in the ecosystem.

Some of the Core Features of NumPy:

1.Multidimensional arrays: With NumPy, hosting homogeneous data elements provides an efficient way of storing and handling large datasets. Later on, arrays can range from being unidimensional, two-dimensional (matrices), or multidimensional.

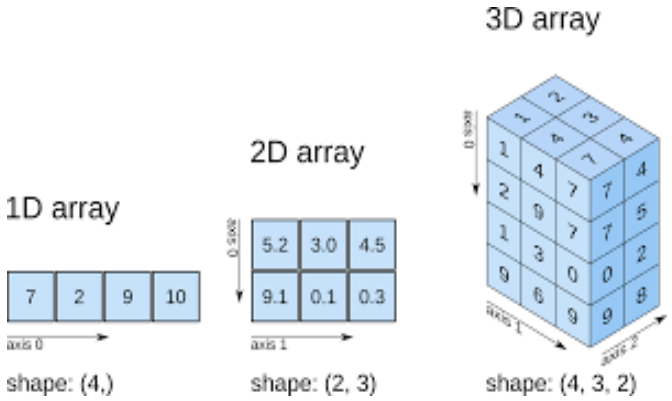

Figure 9:Multidimensional array

2. Speedy operations: Equipped with vectorized functions and operations, NumPy allows fast element-wise operations on arrays, which are implemented in compiled C code, making them very fast unlike their pure Python loop equivalent.

3. Broadcasting: NumPy allows the formation of arrays with different shapes to be combined—added, multiplied, etc.—and operated on as an integral whole in an efficient manner. These broadcasting rules allow arithmetic operations to be performed on arrays of different sizes and shapes.

4. Other Library Integrations: NumPy works well with other scientific libraries of Python: SciPy—Scientific Python; Pandas, a library for data analysis; Matplotlib, a plotting library; and scikit-learn, a machine learning library.

5. Mathematical Functions: NumPy provides an extensive collection of inherent mathematical functions that work element-wise on arrays. These functions range from basic arithmetic and statistical operations to linear algebra and Fourier transform operations;

6. Indexing and Slicing: A superior indexing method is supported by NumPy arrays to effectively extract, change, and manipulate subsets of data. This needs to be integrated with boolean indexing, fancy indexing, and slicing.

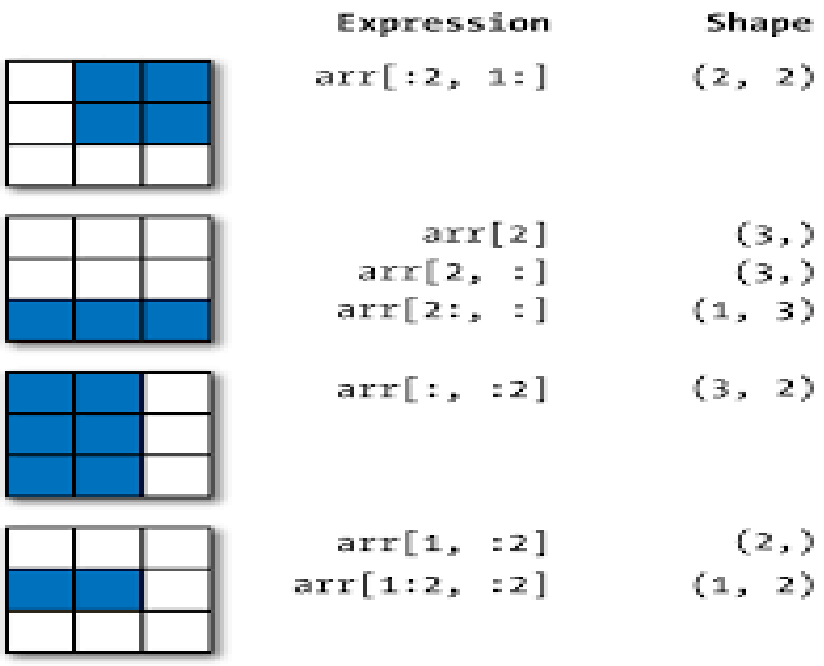

Figure 10:Indexing and Slicing

Do the following to get started with NumPy:

1. Installation: The installation of NumPy is done via the Python package manager, pip. Run this in your terminal or command prompt: **pip install numpy**.

2. Import statement: Use **import numpy as np** to import NumPy to any of your Python scripts or Jupyter notebooks.

3. Array Creation: Use **`np.array()`** to create NumPy arrays from Python lists or tuples, or any of the other functions NumPy provides for creating arrays.

4. Operations and Functions: Use NumPy's built-in functions and methods for many useful mathematical and statistical operations to be performed on the arrays efficiently.

3.2 Benefits of Using NumPy:

1. Efficiency: All manipulations of NumPy arrays are executed in compiled C code; therefore, they are much faster than equivalent operations implemented in Python with loops.
2. Easy to use: NumPy provides a clean, Pythonian syntax for array manipulation that enormously simplifies the code required to implement complex mathematical operations and data manipulations.
3. Interoperability: NumPy arrays can often be passed directly to other libraries and tools in the Python ecosystem without performance loss.
4. Wide Adoption: NumPy provides the foundation for most scientific computing applications and data analysis workflows across academia, research, and industry at large.

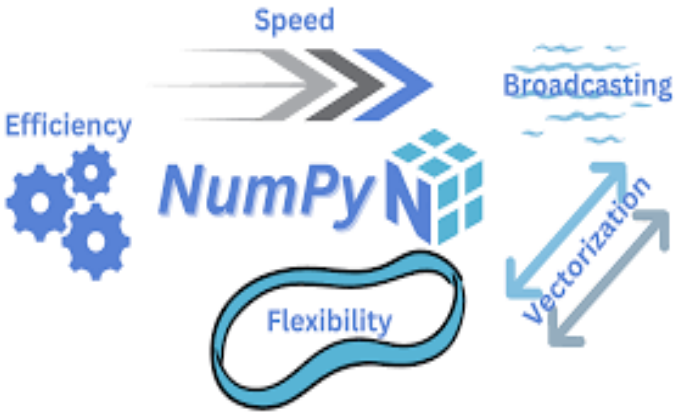

Figure 11:Benefits of numpy

The power of Python mostly lies in the domain of numerical data and scientific computing, so NumPy offers versatility, performance, and very broad functionality, making it a basic tool for anyone dealing with such data. It provides the base for a large number of sophisticated applications in data analysis and machine learning by virtue of its efficiency and simplicity of use.

Chapter 4:

EDA with Pandas

4.1 Introduction

EDA(Exploratory Data Analysis)is a step of paramount importance in data analysis. This phase consists of a review and visualization of data to understand what structure, pattern, and relationship it contains. Pandas is one of the main libraries in Python for doing data manipulation and analysis and is thus bound to play a very key role in EDA since it deals efficiently with structured data. Here is a brief overview of how to perform EDA with Pandas:

Key Steps in EDA with Pandas:

1. Loading Data: Reading with Pandas: It will read data from a variety of sources: independent CSV files, Excel files, SQL databases, and even web APIs. Pandas has functions such as read_csv(), read_excel(), and read_sql() for this.

2. Understanding the Data:Dimension and structure of data with methods like `.shape` returning (# of rows, # of columns) and `.head()` or `.tail()` methods to view first or last few rows of dataset.

3. Data Exploration:Summarize numerical columns with techniques like `.describe()` and `.info()` for an overview on column dtypes and missing values, among other things.

4. Missing Values imputation:Missing values will be spotted and treated with the .isnull() method which enables to detect missing values and, therefore, fill them or drop rows columns with this kind of information.

5. Data Visualization:Follow Pandas's integration with Matplotlib or Seaborn for data visualization. Plot relationships and distributions using `.plot()`, `.hist()`, `.scatter()`, etc.

6. Iterative Exploration and Insights:Iteratively explore the dataset to create visualizations of different views that derive insights into understanding patterns, outliers, trends, and distributions within dataset

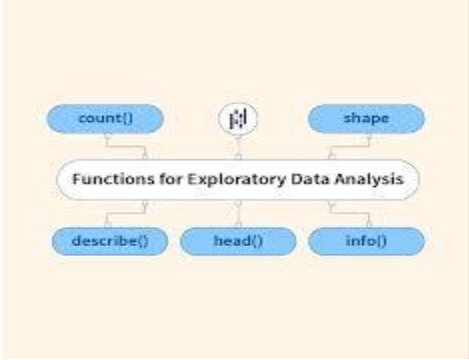

Figure 12:Functions of EDA

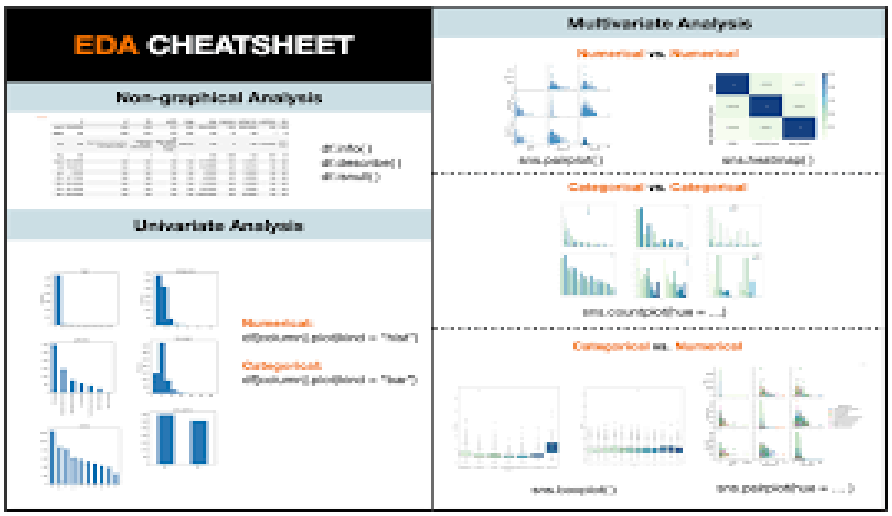

Figure 13:EDA with pandas

4.2 Benefits of EDA with Pandas:

1. Data Understanding: Deep understanding of data, structure, and characteristics before proceedings to some more complex analyses or modeling.
2. Identifying the Patterns: It aids in finding out any patterns, trends, and relationships within the data that can guide the subsequent steps of data preprocessing or hypothesis generation.
3. Data Quality Checks: EDA will determine if there are any quality issues with the data in terms of missing values, outliers, or inconsistencies—thereby helping in data cleaning and preprocessing.
4. Communication: It results in a wide array of visualization and summaries that will help communicate insights effectively to stakeholders, thereby assisting in the correct decision-making process.

Figure 14:Benefits of EDA with pandas

Pandas puts at one's disposal a powerful set of tools and functions that extend the EDA process in Python, making it accessible and efficient for the exploration and analysis of datasets. It is one of the cardinal tools in the toolkit of any data scientist, analyst, or researcher dealing with structured data.

Chapter 5:

Machine Learning

5.1 Introduction

Machine Learning is a sub-division of artificial intelligence that has to do with developing algorithms and statistical models which provides the ability for a computer to learn from data to make decisions or predictions without being explicitly programmed to execute a particular task. This implies allowing computers to recognize patterns in data and learn from examples so as to improve their performance over some time.

Type of Machine Learning:

a) Supervised Learning: The algorithm learns from labeled data, with the values of the output variable known, to predict outcomes for new data.

b) Unsupervised Learning: The algorithm learns the structure of the data from unlabeled data in order to extract hidden relations out of the data.

c) Reinforcement Learning: Agents learn optimal actions by interacting with some environment and receiving a scalar evaluation of their performance in the form of a reward or punishment.

5.2 Common Algorithms of Machine Learning:

a) Supervised: Linear Regression, Decision Trees, Random Forests, Support Vector Machines, Neural Networks.

b) Unsupervised: K-Means Clustering, Principal Component Analysis, Association Rule Learning.

c) Reinforcement Learning: Q-Learning, Deep Q-Networks, Policy Gradient Methods.

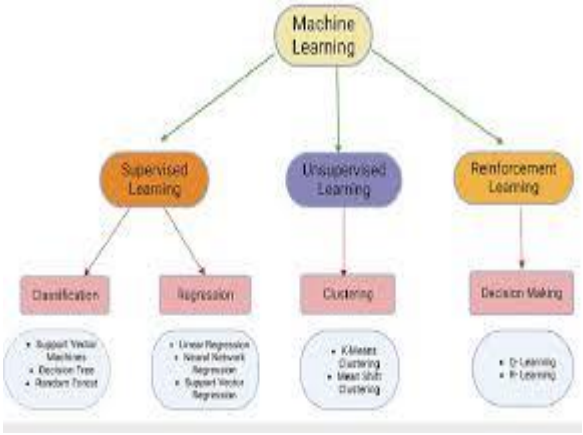

Figure 15:Types of machine learning

5.3 Components of Machine Learning:

a) Data: A dataset contains features, input variables, and labels related to the desired outputs in connection with supervised learning.

b) Model: A mathematical representation or algorithm that learns patterns from data to make predictions or decisions.

c) Training: Running data in a machine learning algorithm to optimize its parameters for better performance.

d) Evaluation: The testing of a model's performance on unseen data for the purpose of estimating model accuracy and the ability of the model to generalize.

e) Deployment: The deployment of trained models into a production environment for generating predictions on new data.

5.5 Elements of a Typical Machine Learning Project:

1. Problem Definition: Exactly define the problem sought to be solved or in general how it all fits into the objectives of the machine learning project.

2. Dataset Collection: Gather relevant datasets containing features and labels in the case of supervised learning.

3. Data Preprocessing: Clean, transform, and preprocess data in readiness for training.

4. Feature Engineering: Features informative for training a model are to be selected, extracted or created.

5. Model Selection: On the type and characteristics of the problem, an appropriate machine learning algorithm is chosen.

6. Train: Train a model on training data using different optimization techniques that minimize losses or maximize specific metrics of performance.

7. Evaluation: A model shall be tested using validation or test data to know the accuracy on real data and get an estimate of general performance.

8. Tuning: The tuning of the model parameters and hyper-parameters to provide optimum performance.

9. Deployment: The deployment of the trained model to production for the prediction on new unseen data.

10. Monitoring and Maintenance: Follow up of the model performance over time and update or re-train the model when needed.

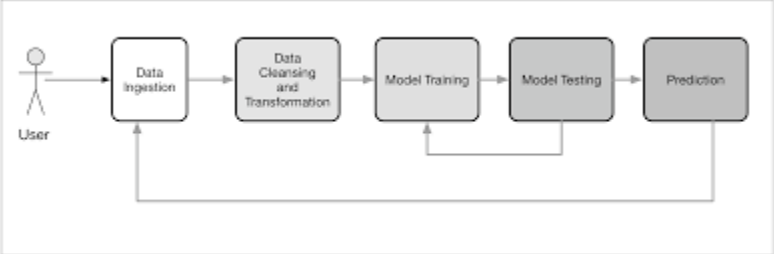

Figure 16:ML

5.4 Applications of Machine Learning:

1. Natural Language Processing: Text classification, Sentiment Analysis, machine translation.
2. Computer Vision: Object Detection, Image Classification, Facial Recognition.
3. Healthcare: Diagnosis of Diseases, Health Diagnosis, Personalized Treatment Recommendation.
4. Finance: Credit scoring, Fraud Detection, Stock Price Prediction.
5. Recommendation Systems: Product Recommendations, Content Personalization.
6. Autonomous Vehicles: Path planning, detects and avoids objects.

Figure 17:Applications of ML

5.6 Benefits of Machine Learning:

1. Automation: ML manages the automation of complex tasks and processes which decreases manual efforts and human errors.

2. Scalability: ML models have the power to process very large data loads and can perform tasks at scale.

3. Personalization: ML provides the facility of tailored recommendations and experiences according to individual preference.

4. Decision Support: Based on data analysis insights and its predictive nature, ML supports making decisions.

5. Innovation: Innovation in different domains is increased by artificial intelligence, which results in the creation of new products, services, and solution works.

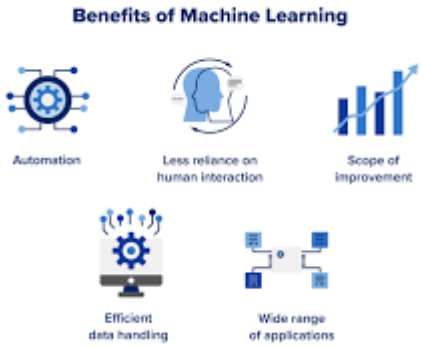

Figure 18:Benefits s of ML

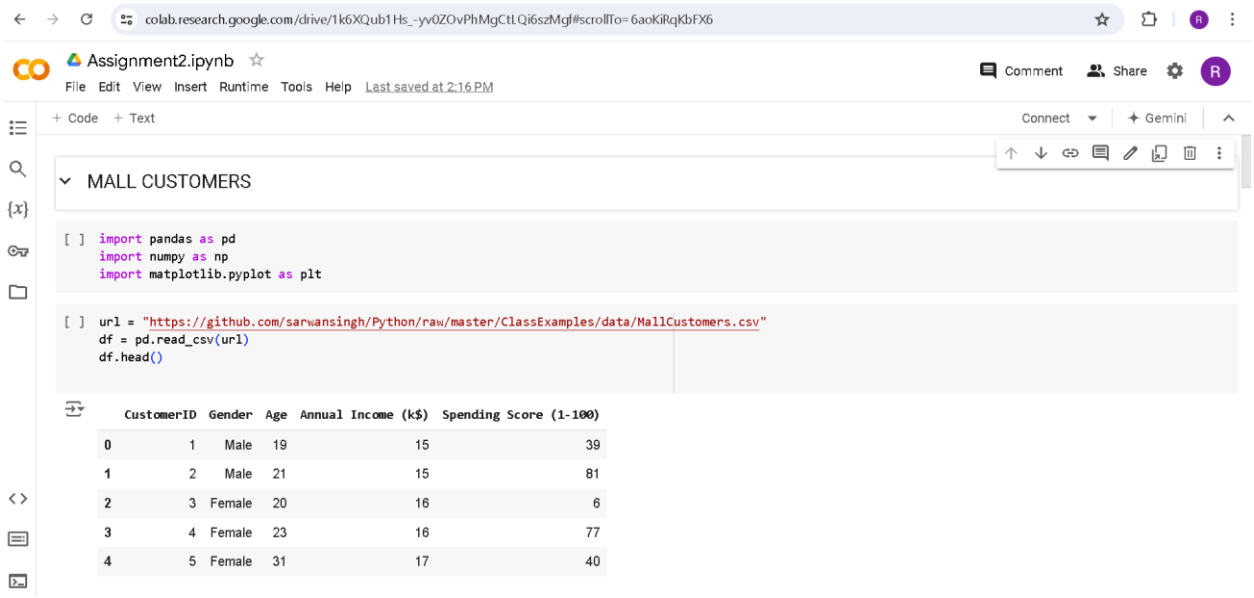

Figure 19:colab code of ML

https://colab.research.google.com/drive/1k6XQub1Hs_-yv0ZOvPhMgCtLQi6szMgf?usp=sharing

The advancement in the evolution of machine learning has continued with improvements in algorithms, computing power, and data availability, thereby making this unsung hero a very potent way for resolving various real-world problems and furthering technological progress across different industries.

Chapter 6

IRIS-FLOWER CLASSIFICATION (LOGISTIC REGRESSION)

6.1 About Project

The Iris flower classification project using logistic regression is a classic machine learning example that demonstrates the application of supervised learning algorithms to predict the species of Iris flowers based on their characteristics.

6.2 Dataset:

The Iris dataset is a well-known dataset in the machine learning community. It consists of 150 samples of Iris flowers, where each sample includes the following four features:

- Sepal length
- Sepal width
- Petal length
- Petal width

Each sample is labelled with one of three classes (species):

- Setosa
- Versicolor
- Virginica

6.3 Pre-Processing Data

6.3.1 Import the packages

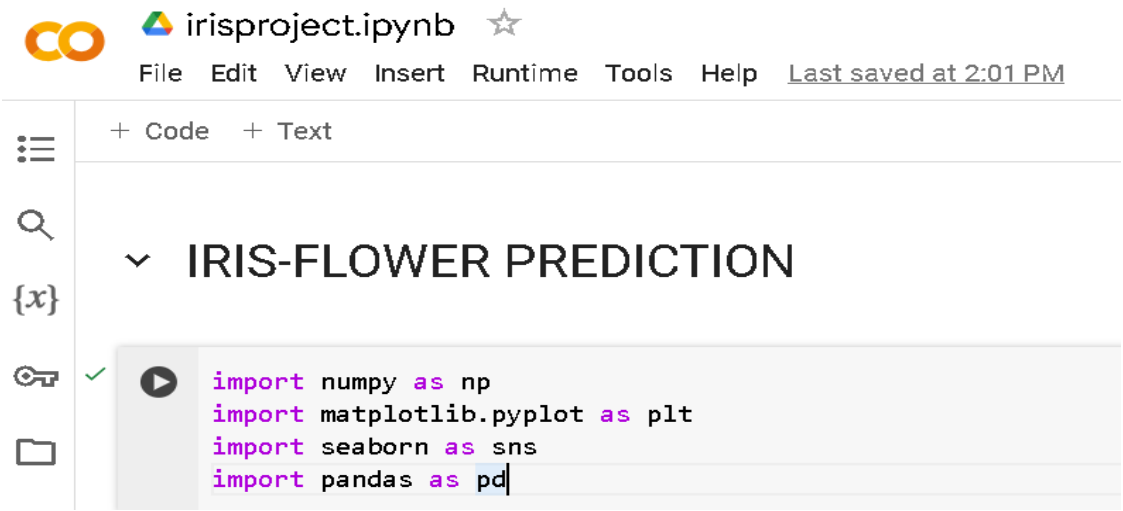

Figure 20:import the package

6.3.2 reading the CSV file

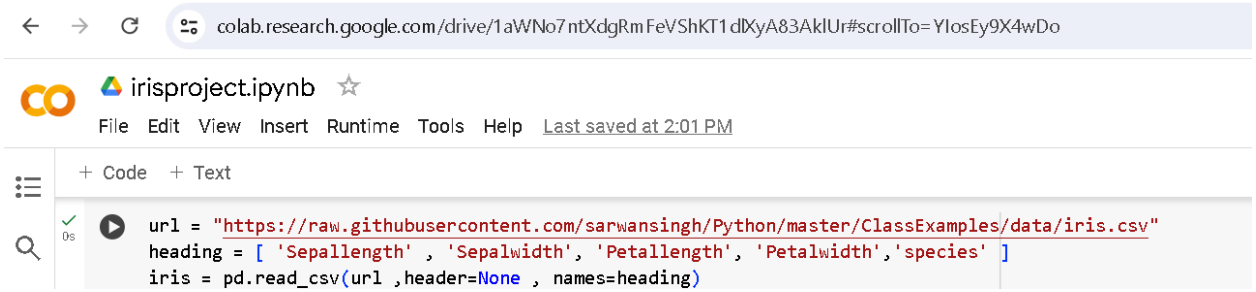

Figure 21:reading the CSV file

6.3.3 cleaning the data

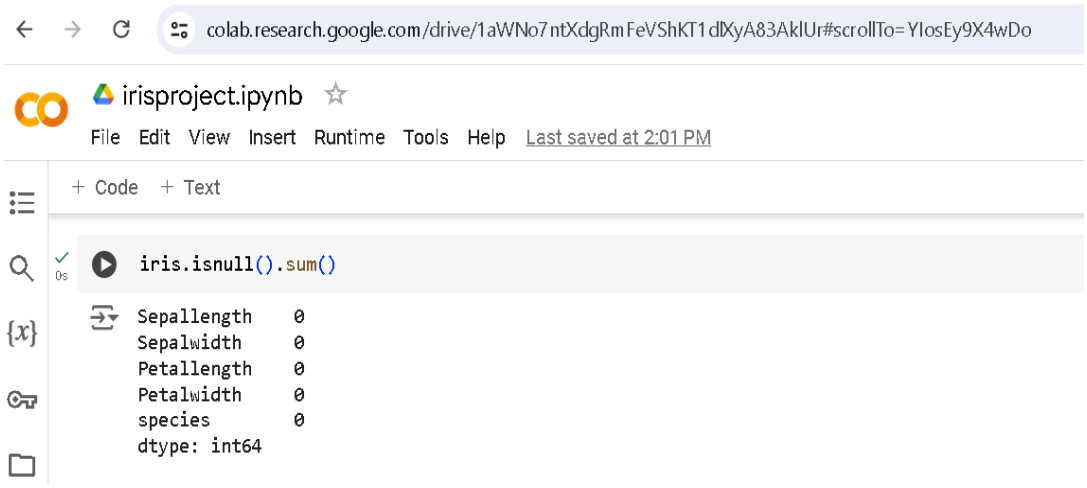

Figure 22:cleaning the data

6.3.4 Visualize the clean data – matplotlib,seaborn

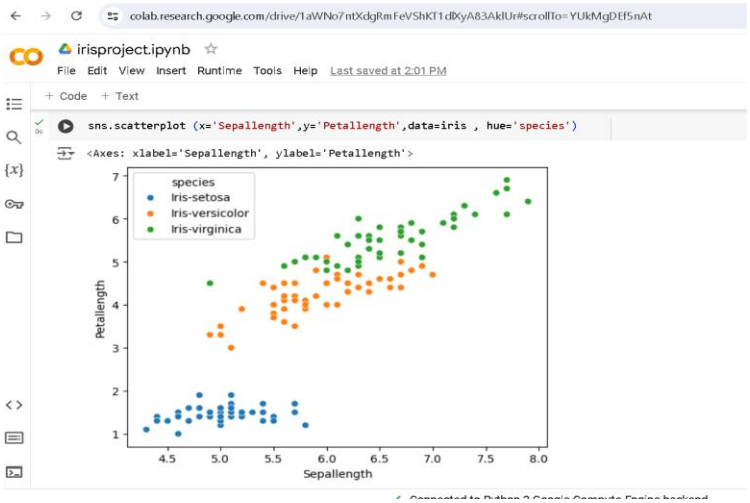

Figure 23:visualizing the data using matplotlib

```
sns.pairplot(iris, hue='species')
```

```
<seaborn.axisgrid.PairGrid at 0x7937aeed6f50>
```

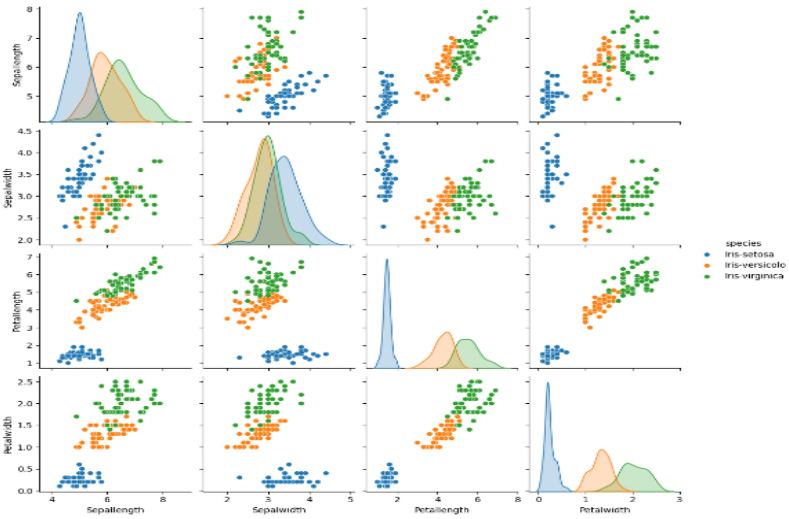

Figure 24:visualizing the data using seaborn

6.3.5 Machine learning

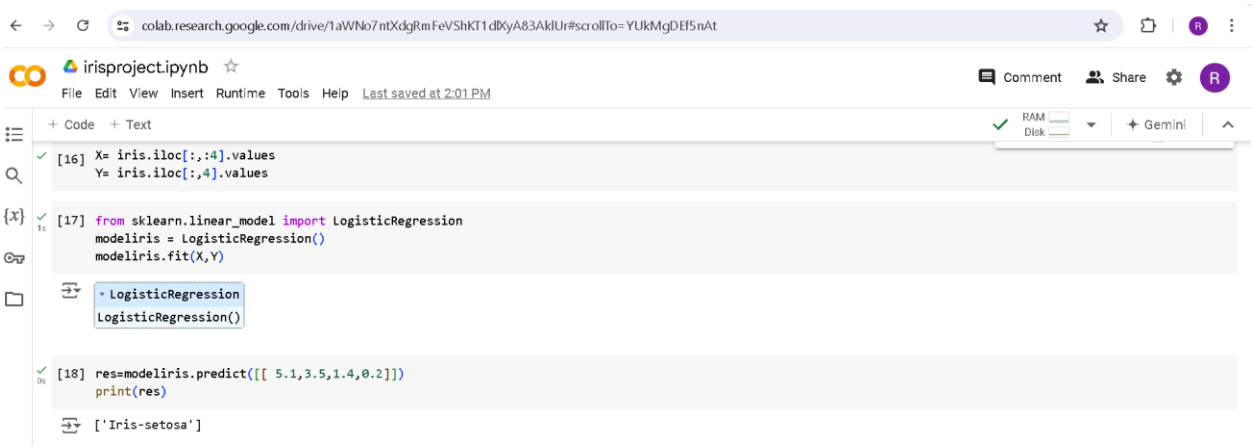

Figure 25:machine learning

Colab file link for project:

https://colab.research.google.com/drive/1aWNo7ntXdgRmFeVShKT1dlXyA83AklUr?usp=sharing

6.4 Front End

6.4.1: Introduction to HTML/CSS/JAVASCRIPT

HTML(HyperText Markup Language):

HTML is the standard markup language used in developing and formatting web pages. Language Conventions: This language forms both the structure and content of a webpage. Elements through which structures can be defined include headings, paragraphs, images, links, forms, etc.

Key Concepts:

Tags: HTML comes with tags defining the elements in angle brackets (< >). Most of these tags are used in pairs: usually opening (<tag>) and closing (</tag>), with content between them.

Figure 26:tags

Attributes: Tags can have attributes that provide additional information about an element, such as class, src, href, etc.

Figure 27:src attributes

```
<!DOCTYPE html>
<html>
<body>

<h2>The href Attribute</h2>

<p>HTML links are defined with the a tag. The link address is specified in
the href attribute:</p>

<a href="https://www.w3schools.com">link</a>

</body>
</html>
```

The href Attribute

HTML links are defined with the a tag. The link address is specified in the href attribute:

link

Figure 28:href attributes

CSS (Cascading Style Sheets):

CSS is the trait that describes the styling of elements of HTML, defining their appearance and presentation on the webpage, to enhance the way things look on the page, by defining layout, colors, fonts, and more.

Selectors: CSS selectors target HTML elements using their tags, class, ID, or other attributes on which the applied styles are to be used.

```
<!DOCTYPE html>
<html>
<head>
<style>
p {
  color: yellow;
  text-align: center;
}
</style>
</head>
<body>

<p>Hello World!</p>

</body>
</html>
```

Hello World!

Figure 29:selectors

JavaScript:

JavaScript is a scripting language; it allows web pages to become interactive and alive. It allows you to manipulate the HTML and CSS, control events, create animations, validate forms, and much more.

Key Concepts:

Variables: Variables are containers for storing data in a JavaScript program. It is good to know that declarations of variable are done using the keywords `var`, `let`, or `const`.

```
<!DOCTYPE html>
<html>
<body>

<h2>JavaScript let</h2>

<p>Redeclaring a JavaScript variable with <b>var</b> is allowed anywhere in
a program:</p>

<p id="demo"></p>

<script>
var x = 2;
// Now x is 2

var x = 3;
// Now x is 3

document.getElementById("demo").innerHTML = x;
</script>

</body>
</html>
```

JavaScript let

Redeclaring a JavaScript variable with **var** is allowed anywhere in a program:

3

Figure 30:variables

My glitch:

WELCOME TO ALML PAGE

EMAIL:	
PASSWORD:	
GENDER:	○ MALE ○ FEMALE
SUBJECTS:	☐ MATHS ☐ PHYSICS

SUBMIT NOW

ABOUT ME

CHAMPS OF NIELIT ROPAR

Figure 31:my first glitch code

https://rakshitafirstproject.glitch.me/

6.4.2 Web server Development using Flask

Figure 32:web server development using flask

Link:

https:iris-flowerprojectbyrakshita.glitch.me/

6.4 Project walkthrough

Figure 33:FIRST STEPS

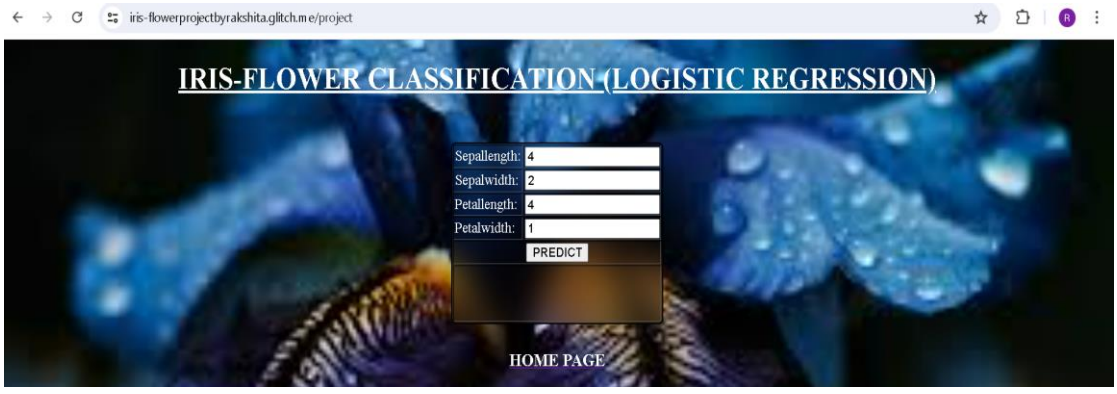

Figure 34:SECOND STEPS

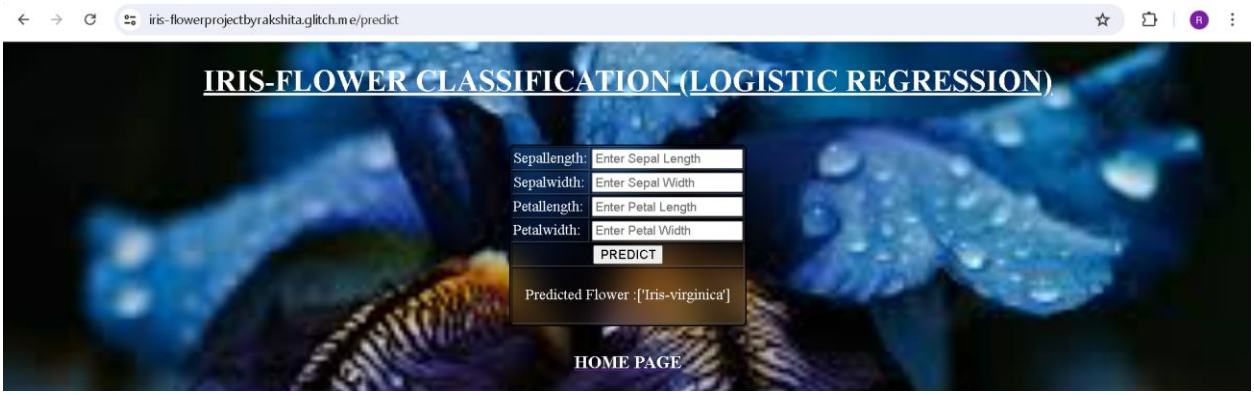

Figure 35:THIRD STEPS

References:

- W3Schools Online Web Tutorials
- Glitch: The friendly community where everyone builds the web