

A Mini Project on
CLASSIFYING IRIS FLOWERS: A MACHINE
LEARNING APPROACH BASED ON PETAL AND
SEPAL MEASUREMENTS

Submitted to

JAWAHARLAL NEHRU TECHNOLOGICAL UNIVERSITY, HYDERABAD

In Partial fulfilment of the requirement for the award of degree of

BACHELOR OF TECHNOLOGY
IN
COMPUTER SCIENCE AND ENGINEERING

By

N.SAI VIGYAN REDDY - 21RA1A05B0

N.NITHIN KUMAR - 21RA1A0573

S.RAVI TEJA - 21RA1A0597

Under the guidance of

Mr. Ritesh Kumar

Assistant Professor, CSE Dept.

Department of Computer Science and Engineering

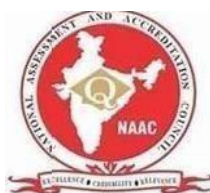


DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
KOMMURI PRATAP REDDY INSTITUTE OF TECHNOLOGY
(Affiliated to JNTUH, Ghanpur(V), Ghatkesar(M), Medchal(D)-500088)

2021 -2025

KOMMURI PRATAP REDDY INSTITUTE OF TECHNOLOGY

(Affiliated to JNTUH, Ghanpur(V), Ghatkesar(M), Medchal(D)-500088)



CERTIFICATE

This is to certify that the project work entitled “**CLASSIFYING IRIS FLOWERS: A MACHINE LEARNING APPROACH BASED ON PETAL AND SEPAL MEASUREMENTS**” is submitted by Mr. N. SAI VIGNYAN, Mr. N. NITHIN KUMAR, Mr. S. RAVI TEJA Bonafide students of **Kommuri Pratap Reddy Institute of Technology** in partial fulfilment of the requirement for the award of the degree of Bachelor of Technology in **Computer Science and Engineering** of the **Jawaharlal Nehru Technological University Hyderabad**, during the year

2024-25.

Internal Guide

Mr.Ritesh Kumar

HOD

Dr. S. Kavitha

Project Coordinator

Mr.M.Rakesh

External Examiner

DECLARATION

We hereby declare that this project work entitled “**CLASSIFYING IRIS FLOWERS: A MACHINE LEARNING APPROACH BASED ON PETAL AND SEPAL MEASUREMENTS**” in partial fulfillment of requirements for the award of degree of **Computer Science and Engineering** is a Bonafide work carried out by us during the academic year 2024- 25.

We further declare that this project is a result of our effort and has not been submitted for the award of any degree by us to any institute.

By

N. SAI VIGNYAN (21RA1A05B0)

N. NITHIN KUMAR (21RA1A0573)

S. RAVI TEJA (21RA1A0597)

ACKNOWLEDGEMENT

It gives us immense pleasure to acknowledge with gratitude, the help and support extended throughout the project report from the following:

We will be very much grateful to almighty our **Parents** who have made us capable of carrying out our job.

We express our profound gratitude to **Dr. RAVINDRA EKLARKER, Principal of Kommuri Pratap Reddy Institute of Technology**, who has provided necessary infrastructure and resources in completing our project report successfully.

We are grateful to **Dr. S . KAVITHA** who is our **Head of the Department, CSE** for her amiable ingenious and adept suggestions and pioneering guidance during the project report.

We express our gratitude and thanks to the **coordinator Mr.M.Rakesh** of our department for his contribution for making it success within the given time duration.

We express our deep sense of gratitude and thanks to **Internal Guide, Mr. Ritesh Kumar Assistant Professor** for his guidance during the project report.

We are also very thankful to our **Management, Staff Members** and all **Our Friends** for their valuable suggestions and timely guidance without which we would not have been completed it.

By

N. SAI VIGNYAN (21RA1A05B0)

N. NITHIN KUMAR (21RA1A0573)

S. RAVI TEJA (21RA1A0597)



KOMMURI PRATAP REDDY INSTITUTE OF TECHNOLOGY

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

Vision of the Institute

To emerge as a premier institute for high quality professional graduates who can contribute to economic and social developments of the Nation.

Mission of the Institute

Mission	Statement
IM₁	To have holistic approach in curriculum and pedagogy through industry interface to meet the needs of Global Competency.
IM₂	To develop students with knowledge, attitude, employability skills, entrepreneurship, research potential and professionally Ethical citizens.
IM₃	To contribute to advancement of Engineering & Technology that would help to satisfy the societal needs.
IM₄	To preserve, promote cultural heritage, humanistic values and Spiritual values thus helping in peace and harmony in the society.



KOMMURI PRATAP REDDY INSTITUTE OF TECHNOLOGY

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

Vision of the Department

To Provide Quality Education in Computer Science for the innovative professionals to work for the development of the nation.

Mission of the Department

Mission	Statement
DM₁	Laying the path for rich skills in Computer Science through the basic knowledge of mathematics and fundamentals of engineering
DM₂	Provide latest tools and technology to the students as a part of learning infrastructure
DM₃	Training the students towards employability and entrepreneurship to meet the societal needs.
DM₄	Grooming the students with professional and social ethics.



KOMMURI PRATAP REDDY INSTITUTE OF TECHNOLOGY

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

Program Educational Objectives (PEOs)

PEO's	Statement
PEO1	The graduates of Computer Science and Engineering will have successful career in technology.
PEO2	The graduates of the program will have solid technical and professional foundation to continue higher studies.
PEO3	The graduate of the program will have skills to develop products, offer services and innovation.
PEO4	The graduates of the program will have fundamental awareness of industry process, tools and technologies.

TABLE OF CONTENTS

1. Abstract	10
2. Chapter 1: Introduction	11
1.1 Overview	11
1.2 Problem Statement	12
1.3 Research Motivation	12
1.4 Existing Systems	13
1.5 Research Objective	14
1.6 Need	14
1.7 Applications	15
3. Chapter 2: Literature Survey.....	16
4. Chapter 3: Existing Methodology	19
3.1 KNN Algorithm	19
5. Chapter 4: Proposed System	23
4.1 Overview	23
4.2 Data Preprocessing	24
4.3 Splitting the Dataset	26
4.4 Proposed Algorithm	27
6. Chapter 5: UML Diagrams	29
- Class Diagram	30
- Sequence Diagram	31
- Activity Diagram	32
- Deployment Diagram	33
- Use Case Diagram	34
- Component Diagram	35

- Data Flow Diagram	36
7. Chapter 6: Software Environment	38
8. Chapter 7: System Requirements	42
- Software Requirements	42
- Hardware Requirements	43
9. Chapter 8: Functional Requirements	44
- Input Design	45
- Output Design	46
10. Chapter 9: Source Code	48
11. Chapter 10: Results and Discussion	54
10.1 Implementation Description	54
10.2 Dataset Description	56
12. Chapter 11: Conclusion and Future Scope	77
13. References	78

LIST OF FIGURES

1. Overview of KNN Algorithm Workflow	19
2. Proposed System Block Diagram	23
3. Feature Scaling Illustration	25
4. Dataset Splitting Diagram	26
5. UML Class Diagram	30
6. Sequence Diagram	31
7. Activity Diagram	32
8. Deployment Diagram	33
9. Use Case Diagram	34
10. Component Diagram	35

11. Data Flow Diagram	36
12. Confusion Matrix for KNN	50
13. Confusion Matrix for Logistic Regression	52

LIST OF SCREENSHOTS

1. Python Environment Setup	39
2. Sample Dataset Preview	40
3. Data Preprocessing Steps	41
4. KNN Algorithm Output	50
5. Logistic Regression Output	52
6. Model Performance Comparison Table	53
7. Final Predictions on Test Data	56

CLASSIFYING IRIS FLOWERS: A MACHINE LEARNING APPROACH BASED ON PETAL AND SEPAL MEASUREMENTS

ABSTRACT

Classifying iris flowers based on petal and sepal measurements is a fundamental problem in botanical research. In botany, this approach aids researchers in species identification and taxonomy, facilitating the study of plant biodiversity and evolution. Additionally, in horticulture and agriculture, accurate classification of iris species can inform breeding programs, helping to develop new cultivars with desirable traits. Moreover, in environmental science, understanding the distribution and abundance of different iris species contributes to ecosystem monitoring and conservation efforts. Furthermore, the machine learning techniques employed in this approach can be generalized to other classification tasks in fields such as healthcare, finance, and marketing. Traditional methods for classifying iris flowers often rely on manual measurements and expert knowledge, which can be time-consuming and subjective. These methods may also lack scalability and generalization capabilities, as they rely heavily on human expertise for feature selection and classification. Additionally, manual classification may lead to inconsistencies and errors, especially when dealing with large datasets or subtle differences between species. Furthermore, traditional approaches may struggle to handle high-dimensional feature spaces or complex relationships between features, limiting their effectiveness in accurately classifying iris flowers. In contrast to traditional methods, the proposed system employs a machine learning approach to classify iris flowers based on petal and sepal measurements. This work utilizes a supervised learning algorithm to automatically learn discriminative patterns from the input features. Through feature extraction and model training on a labeled dataset of iris flowers, our system learns to distinguish between different species based on their petal and sepal characteristics. Moreover, this work employs techniques such as cross-validation and hyperparameter tuning to optimize the model's performance and ensure robustness.

CHAPTER 1

INTRODUCTION

1.1 Overview

In the realm of botanical research, the classification of iris flowers based on their petal and sepal measurements stands as a foundational challenge. This task holds significance not only in the context of botany but also extends its implications into horticulture, agriculture, and environmental science. By utilizing machine learning techniques, this research endeavors to automate and enhance the process of iris flower classification. The proposed system harnesses the power of supervised learning algorithms to discern discriminative patterns from input features, paving the way for efficient species identification. Through meticulous feature extraction and model training on a labeled dataset, the system learns to differentiate between iris species based on their unique petal and sepal characteristics. Moreover, the incorporation of cross-validation and hyperparameter tuning techniques ensures the robustness and reliability of the classification model.

1.2 Problem Statement

Traditional methods for classifying iris flowers heavily rely on manual measurements and expert knowledge, resulting in time-consuming and subjective processes prone to inconsistencies and errors. Moreover, these methods often struggle to handle the complexities of high-dimensional feature spaces and subtle differences between species. The need for a more efficient and accurate classification approach is evident, especially considering the importance of iris species identification in various domains such as botany, horticulture, and environmental science. Thus, the primary problem addressed by this research is to develop a machine learning-based system capable of automating iris flower classification based on petal and sepal measurements while overcoming the limitations of traditional methods.

1.3 Research Motivation

The motivation behind this research stems from the multifaceted implications of iris flower classification. In botany, accurate species identification facilitates the study of plant biodiversity and evolution. In horticulture and agriculture, it aids in breeding programs aimed at developing new cultivars with desirable traits. Additionally, in environmental science, understanding the distribution and abundance of iris species contributes to ecosystem

monitoring and conservation efforts. By employing machine learning techniques, this research seeks to streamline the classification process, thereby advancing research in these fields and fostering interdisciplinary collaborations.

1.4 Existing Systems

Existing systems for classifying iris flowers predominantly rely on manual measurements and expert knowledge, leading to subjective and error-prone outcomes. These traditional approaches often lack scalability and generalization capabilities, limiting their effectiveness in handling large datasets or complex feature relationships. By contrast, the proposed machine learning approach automates the classification process, leveraging supervised learning algorithms to extract discriminative patterns from input features. This shift towards automation not only improves accuracy but also enhances scalability and generalization capabilities.

1.5 Research Objective

The primary objective of this research is to develop a machine learning-based system capable of accurately classifying iris flowers based on their petal and sepal measurements. By employing supervised learning algorithms and techniques such as feature extraction, model training, cross-validation, and hyperparameter tuning, the system aims to achieve robust and reliable classification performance. Additionally, the research seeks to compare the effectiveness of the proposed machine learning approach with traditional methods, highlighting its advantages in terms of accuracy, scalability, and generalization.

1.6 Need

The need for a machine learning-based approach to iris flower classification arises from the shortcomings of traditional methods, including their reliance on manual measurements and subjective expert knowledge. Furthermore, traditional approaches often struggle to handle the complexities of high-dimensional feature spaces and subtle differences between species, hindering their effectiveness in accurate classification. By automating the classification process and leveraging machine learning techniques, this research addresses these challenges, offering a more efficient and accurate solution for iris flower classification.

1.7 Application

The application of machine learning-based iris flower classification extends beyond the realm of botany, encompassing various domains such as horticulture, agriculture, and environmental

science. In botany, the automated classification of iris species aids researchers in studying plant biodiversity and evolution. In horticulture and agriculture, it informs breeding programs aimed at developing new cultivars with desirable traits. Additionally, in environmental science, understanding the distribution and abundance of iris species contributes to ecosystem monitoring and conservation efforts. Moreover, the techniques employed in this research can be generalized to other classification tasks in fields such as healthcare, finance, and marketing, showcasing the versatility and applicability of the proposed approach.

CHAPTER 2

LITERATURE SURVEY

The practice of categorizing distinct database objects into one or more groups or categories is known as data mining. The objective of the classification step is to assign each instance to the relevant target class. This section provides an overview of the most recent and practical classification methods that have been developed by researchers in the last two years across many ML domains.

[1] David W. Corne and Ziauddin Ursani proposed in their paper an evolutionary algorithm for nonlinear discriminant classifier, in which they mentioned that it was not appropriate for learning tasks with any individual single value. Hence they tested this method on two data sets, Iris Flower and Balance Scale, where decisions of class membership can only be affected collectively by individual lineaments of flower. [2] Detlef Nauck and Rudolf Kruse have proposed a new approach in which they classify the data on the basis of fuzzy Neural Networks. They used backpropagation algorithm to define other class of fuzzy perceptron. They concluded that on increasing the number on hidden layer, increase the need of more training cycles and raises incorrect results. Hence the better result can be evaluated using 3 hidden layers also. [3] To overcome the problem of data depth, long parameters, long training time and slow convergence of Neural Networks, two other algorithms Transfer Learning and Adam Deep Learning optimization algorithms were considered for flower recognition by Jing FENG, Zhiven WANG, Min ZHA and Xiliang CAO. Where, Transfer Learning was based on features in isomorphic spaces. They concluded in their paper that if the pictures of flowers placed into model training in the form of batches, then it will meliorate the speed of updating the value of parameters and provide the best optimal result of parameter values. [4] Rong-Guo Huang, Sang-Hyeon Jin, Jung-Hyun Kim, Kwang-Seog Hong focus on recognition of flower using Difference Image Entropy (DIE), which is based on feature extraction. According to their research, the experimental results give 95% of recognition rate as an average. The DIE based approach takes original image of flower as an input, and applies pre-processing and DIE computation to produce recognition result. The Gaussian Naive Bayes technique is used by Zainab Iqbal [5] to categorize the species of the iris flower. We analyze the iris dataset using a scatter matrix and a scatter plot that is constructed. The algorithm and Python are both utilized in the paper to categorize the many species of iris flowers. We can see that this technique is effective for supervised learning classification because it achieves a 95% accuracy rate. A C4.5

decision tree was suggested by Mijwil and Abttan [6] as a way to lessen the impacts of overfitting. IRIS, Car Assessment, Bottle, and WINE were the datasets utilized; both of these may be found in the UCI ML library. The issue with this classifier is that it overfits because of its large number of nodes and divisions. It is possible that this overfitting will undermine the classification system. The experimental results demonstrated that, with an accuracy of roughly 92%, the genetic algorithm was effective in reducing the effects of overfitting on the four datasets and increasing the Confidence Factor (CF) of the C4.5 decision tree. Rong-Guo Huang [7] focuses on flower detection using Difference Image Entropy (DIE), a feature extraction-based method. Their analysis of the experimental findings shows that the average recognition rate was 95%. The DIE-based approach utilizes pre-processing and DIE computing to provide a recognition result from an original image of the flower.

Patrick [8] concentrated on the dataset's statistical analysis using the iris flower example. They are examining two alternative approaches in his study. To identify the various classification patterns, the dataset is plotted. Then, using a java program they developed, they may retrieve statistical data. In her research, Poojitha [9] employed neural networks to examine data sets on iris flowers. A branch of computer science called machine learning. We have already loaded the iris dataset and have divided it into three groups. They divided the dataset into groups using the k-means technique. Large-scale information aggregation is the main use of a neural network. Additionally, it is employed in the mining of data, quantization of vectors, work approximation, division of images, and highlight extraction. Without any oversight, the findings are divided into three distinct iris species. Lakhdoura and Elayachi [10] used WEKA 3.9 to do a test comparing the performance of two classifier methods: J48 (c4.5) and RF on the IRIS features. As a result, the University of California, Irvine's ML library provides access to the IRIS plant dataset, one of the most popular datasets for classification problems (UCI). Zebari, D. A et.al [11] The researchers also contrasted the outcomes of both classifiers on numerous efficacy assessment metrics. According to the results, the J48 classifier performs better than the Random Forest (RF) classifier for predicting IRIS variety using a range of measures, including classification precision, mean absolute error, and construction time. The accuracy of the J48 classifier is 95.83%, while that of the Random Forest is 95.55%.

] Abdulqadir, H. R et.al [12] Numerous research has been done using different methods to identify the species of the iris flower. Every study employs a different method. The issue is the categorization and identification of iris flower species based on their characteristics. Ibrahim, D. A. et.al [13] With the use of this classification and pattern, future predictions for unknown

data can be made with greater accuracy. The dataset for iris flowers is placed into the machine learning prototype for the iris flower species approach.

CHAPTER 3

EXISTING METHODOLOGY

3.1 KNN ALGORITHM

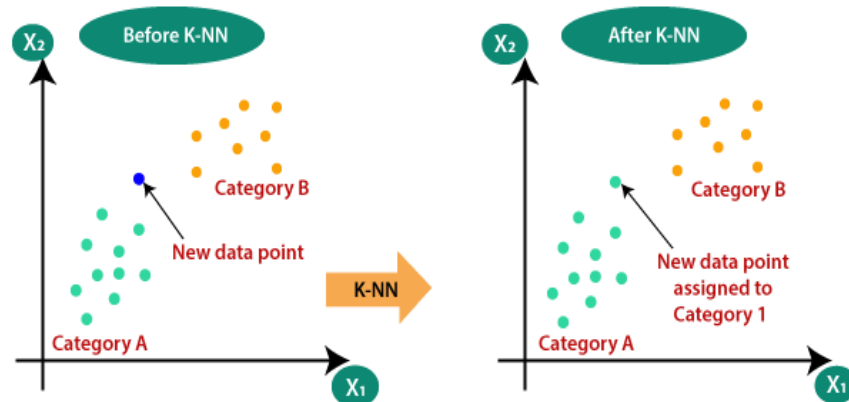
K-Nearest Neighbor (KNN) Algorithm for Machine Learning

- K-Nearest Neighbour is one of the simplest Machine Learning algorithms based on Supervised Learning technique.
- K-NN algorithm assumes the similarity between the new case/data and available cases and put the new case into the category that is most similar to the available categories.
- K-NN algorithm stores all the available data and classifies a new data point based on the similarity. This means when new data appears then it can be easily classified into a well suite category by using K- NN algorithm.
- K-NN algorithm can be used for Regression as well as for Classification but mostly it is used for the Classification problems.
- K-NN is a non-parametric algorithm, which means it does not make any assumption on underlying data.
- It is also called a lazy learner algorithm because it does not learn from the training set immediately instead it stores the dataset and at the time of classification, it performs an action on the dataset.
- KNN algorithm at the training phase just stores the dataset and when it gets new data, then it classifies that data into a category that is much similar to the new data.
- **Example:** Suppose, we have an image of a creature that looks similar to cat and dog, but we want to know either it is a cat or dog. So for this identification, we can use the KNN algorithm, as it works on a similarity measure. Our KNN model will find the similar features of the new data set to the cats and dogs images and based on the most similar features it will put it in either cat or dog category.



Why do we need a K-NN Algorithm?

Suppose there are two categories, i.e., Category A and Category B, and we have a new data point x_1 , so this data point will lie in which of these categories. To solve this type of problem, we need a K-NN algorithm. With the help of K-NN, we can easily identify the category or class of a particular dataset. Consider the below diagram:



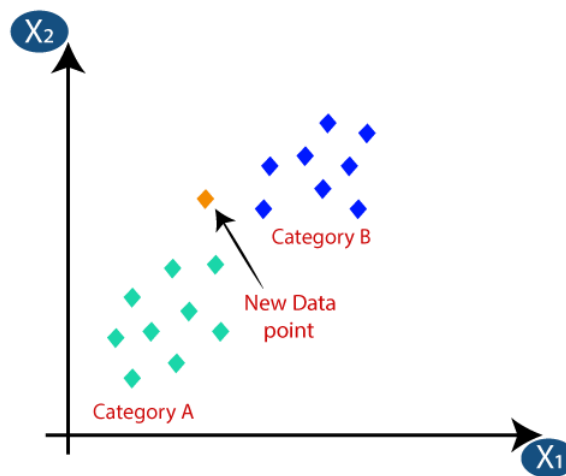
How does K-NN work?

The K-NN working can be explained on the basis of the below algorithm:

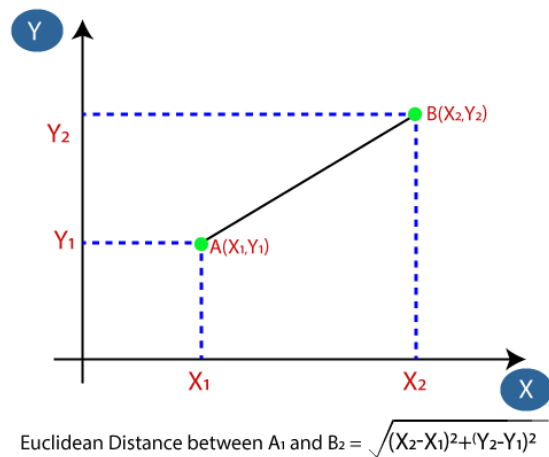
- **Step-1:** Select the number K of the neighbours
- **Step-2:** Calculate the Euclidean distance of K number of neighbours
- **Step-3:** Take the K nearest neighbours as per the calculated Euclidean distance.

- **Step-4:** Among these k neighbours, count the number of the data points in each category.
- **Step-5:** Assign the new data points to that category for which the number of the neighbour is maximum.
- **Step-6:** Our model is ready.

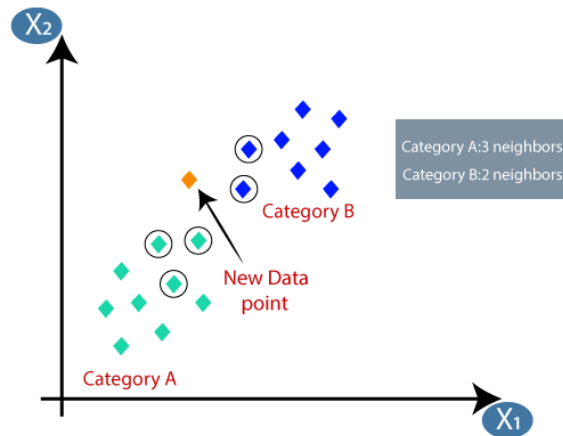
Suppose we have a new data point and we need to put it in the required category. Consider the below image:



- Firstly, we will choose the number of neighbours, so we will choose the $k=5$.
- Next, we will calculate the Euclidean distance between the data points. The Euclidean distance is the distance between two points, which we have already studied in geometry. It can be calculated as:



- By calculating the Euclidean distance, we got the nearest neighbours, as three nearest neighbours in category A and two nearest neighbours in category B. Consider the below image:



- As we can see the 3 nearest neighbours are from category A, hence this new data point must belong to category A.

How to select the value of K in the K-NN Algorithm?

- There is no particular way to determine the best value for "K", so we need to try some values to find the best out of them. The most preferred value for K is 5.
- A very low value for K such as K=1 or K=2, can be noisy and lead to the effects of outliers in the model.
- Large values for K are good, but it may find some difficulties.

Advantages of KNN Algorithm:

- It is simple to implement.
- It is robust to the noisy training data
- It can be more effective if the training data is large.

Disadvantages of KNN Algorithm:

- Always needs to determine the value of K which may be complex some time.
- The computation cost is high because of calculating the distance between the data points for all the training samples.

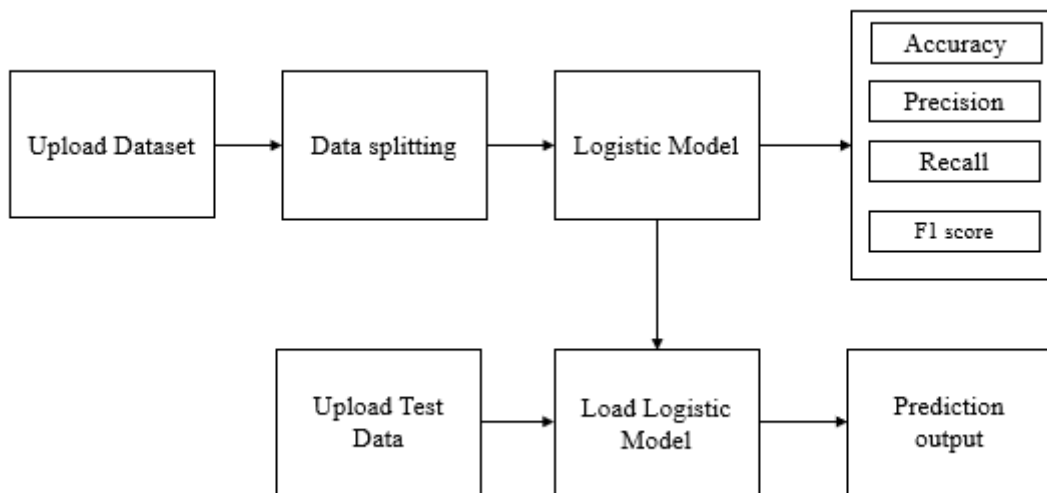
CHAPTER 4

PROPOSED SYSTEM

4.1 Overview

Step 1: Iris dataset

The first step in this research procedure involves obtaining the Iris dataset. This dataset is a classic in the field of machine learning and consists of measurements of various iris flowers. These measurements typically include attributes such as sepal length, sepal width, petal length, and petal width, along with the corresponding species of iris. The dataset serves as the foundation for training and evaluating machine learning models to classify iris flowers based on their characteristics.



4.1 Block Diagram.

Step 2: Dataset preprocessing

Once the dataset is acquired, it undergoes preprocessing to ensure its suitability for machine learning algorithms. This involves handling any missing or null values in the data, which can adversely affect the performance of the models. Additionally, label encoding may be applied to convert categorical labels, such as the species of iris, into numerical values that can be processed by machine learning algorithms.

Step 3: Existing KNN

The K-Nearest Neighbors (KNN) algorithm is a simple yet effective method for classification. In this step, the existing KNN algorithm is applied to the preprocessed and feature-selected dataset to classify iris flowers based on their petal and sepal measurements. KNN works by assigning a class label to a data point based on the majority class among its K nearest neighbors in the feature space.

Step 4: Proposed Logistics

Logistic Regression is a commonly used linear classification algorithm that models the probability of a binary outcome. In this step, the proposed Logistic Regression model is trained on the preprocessed and feature-selected dataset to classify iris flowers. Logistic Regression estimates the probability that a given input belongs to a particular class using a logistic function, making it suitable for multi-class classification tasks such as iris flower classification.

Step 5: Performance comparison

Once both the existing KNN and proposed Logistic Regression models are trained, their performance is evaluated and compared. Performance metrics such as accuracy, precision, recall, and F1-score are typically calculated to assess the effectiveness of each model in correctly classifying iris flowers. This step helps identify the strengths and weaknesses of each approach and determine which model performs better for the task at hand.

Step 6: Prediction of output from test data with Logistics regression trained model

Finally, the trained Logistic Regression model is used to predict the class labels of iris flowers in unseen test data. This step involves feeding the test data into the trained model and obtaining predictions for the corresponding class labels. The accuracy of the model's predictions on the test data provides insight into its generalization ability and real-world performance.

This research procedure outlines a systematic approach to classifying iris flowers using machine learning techniques, from data acquisition and preprocessing to model training, evaluation, and prediction. By following these steps, researchers can develop accurate and reliable classification models for various applications in botany, horticulture, agriculture, environmental science, and beyond.

4.2 Data Preprocessing

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. A real-world data generally contains noises, missing values, and maybe in an unusable format which cannot be directly used for machine learning models. Data pre-processing is required tasks for cleaning the data and making it suitable for a machine learning model which also increases the accuracy and efficiency of a machine learning model.

- Getting the dataset
- Importing libraries
- Importing datasets
- Finding Missing Data
- Encoding Categorical Data
- Splitting dataset into training and test set

Importing Libraries: 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:

```
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:

```
import matplotlib.pyplot as mpt
```

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

Pandas: The last library is the Pandas library, which 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. Here, we have used pd as a short name for this library. Consider the below image:

```
1 # importing libraries
2 import numpy as nm
3 import matplotlib.pyplot as mtp
4 import pandas as pd
5
```

Handling Missing data: The next step of data preprocessing is to handle missing data in the datasets. If our dataset contains some missing data, then it may create a huge problem for our machine learning model. Hence it is necessary to handle missing values present in the dataset. There are mainly two ways to handle missing data, which are:

- By deleting the particular row: The first way is used to commonly deal with null values. In this way, we just delete the specific row or column which consists of null values. But this way is not so efficient and removing data may lead to loss of information which will not give the accurate output.
- By calculating the mean: In this way, we will calculate the mean of that column or row which contains any missing value and will put it on the place of missing value. This strategy is useful for the features which have numeric data such as age, salary, year, etc.

Encoding Categorical data: Categorical data is data which has some categories such as, in our dataset; there are two categorical variables, 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.

Feature Scaling: Feature scaling is the final step of data preprocessing in machine learning. It is a technique to standardize the independent variables of the dataset in a specific range. In feature scaling, we put our variables in the same range and in the same scale so that no variable dominates the other variable. A machine learning model is based on Euclidean distance, and if we do not scale the variable, then it will cause some issue in our machine learning model. Euclidean distance is given as:

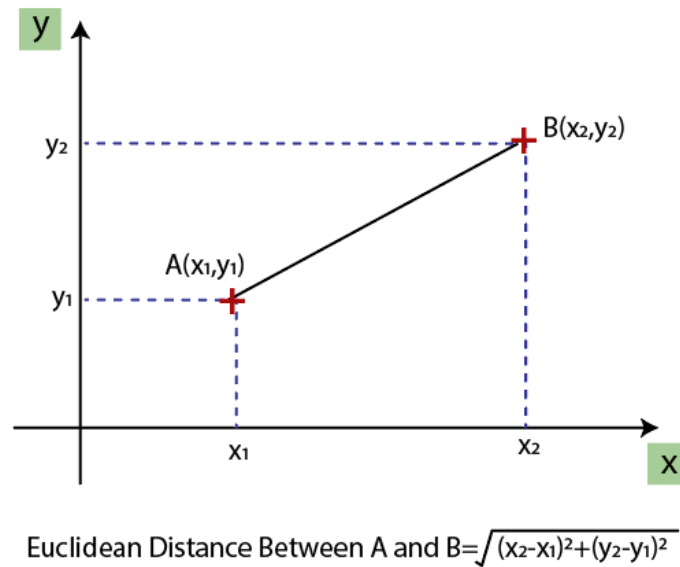


Figure 4.2: Feature scaling

If we compute any two values from age and salary, then salary values will dominate the age values, and it will produce an incorrect result. So, to remove this issue, we need to perform feature scaling for machine learning.

4.3 Splitting the Dataset

In machine learning data preprocessing, we divide our dataset into a training set and test set. This is one of the crucial steps of data preprocessing 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:

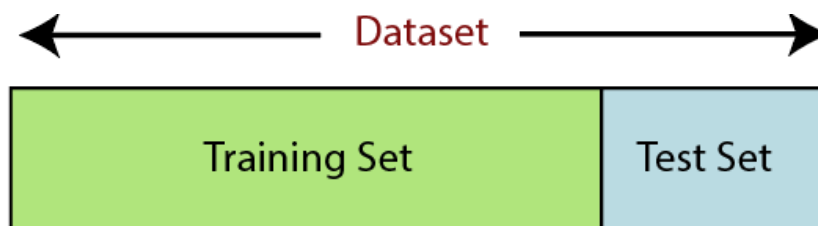


Figure 4.2: Splitting the dataset.

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.

For splitting the dataset, we will use the below lines of code:

```
from sklearn.model_selection import train_test_split

x_train, x_test, y_train, y_test= train_test_split(x, y, test_size= 0.2, random_state=0)
```

Explanation: 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. The last parameter random_state is used to set a seed for a random generator so that you always get the same result, and the most used value for this is 42.

4.4 Proposed Algorithm

Logistic regression is used for binary classification where we use sigmoid function, that takes input as independent variables and produces a probability value between 0 and 1.

For example, we have two classes Class 0 and Class 1 if the value of the logistic function for an input is greater than 0.5 (threshold value) then it belongs to Class 1 otherwise it belongs to Class 0. It's referred to as regression because it is the extension of linear regression but is mainly used for classification problems.

Key Points:

Logistic regression predicts the output of a categorical dependent variable. Therefore, the outcome must be a categorical or discrete value.

It can be either Yes or No, 0 or 1, true or False, etc. but instead of giving the exact value as 0 and 1, it gives the probabilistic values which lie between 0 and 1.

In Logistic regression, instead of fitting a regression line, we fit an “S” shaped logistic function, which predicts two maximum values (0 or 1).

Logistic Function – Sigmoid Function

The sigmoid function is a mathematical function used to map the predicted values to probabilities.

It maps any real value into another value within a range of 0 and 1. The value of the logistic regression must be between 0 and 1, which cannot go beyond this limit, so it forms a curve like the “S” form.

The S-form curve is called the Sigmoid function or the logistic function.

In logistic regression, we use the concept of the threshold value, which defines the probability of either 0 or 1. Such as values above the threshold value tends to 1, and a value below the threshold values tends to 0.

Types of Logistic Regression

On the basis of the categories, Logistic Regression can be classified into three types:

Binomial: In binomial Logistic regression, there can be only two possible types of the dependent variables, such as 0 or 1, Pass or Fail, etc.

Multinomial: In multinomial Logistic regression, there can be 3 or more possible unordered types of the dependent variable, such as “cat”, “dogs”, or “sheep”

Ordinal: In ordinal Logistic regression, there can be 3 or more possible ordered types of dependent variables, such as “low”, “Medium”, or “High”.

Assumptions of Logistic Regression

We will explore the assumptions of logistic regression as understanding these assumptions is important to ensure that we are using appropriate application of the model. The assumption include:

Independent observations: Each observation is independent of the other. meaning there is no correlation between any input variables.

Binary dependent variables: It takes the assumption that the dependent variable must be binary or dichotomous, meaning it can take only two values. For more than two categories SoftMax functions are used.

Linearity relationship between independent variables and log odds: The relationship between the independent variables and the log odds of the dependent variable should be linear.

No outliers: There should be no outliers in the dataset.

Large sample size: The sample size is sufficiently large

Terminologies involved in Logistic Regression

Here are some common terms involved in logistic regression:

Independent variables: The input characteristics or predictor factors applied to the dependent variable's predictions.

Dependent variable: The target variable in a logistic regression model, which we are trying to predict.

Logistic function: The formula used to represent how the independent and dependent variables relate to one another. The logistic function transforms the input variables into a probability value between 0 and 1, which represents the likelihood of the dependent variable being 1 or 0.

Odds: It is the ratio of something occurring to something not occurring. it is different from probability as the probability is the ratio of something occurring to everything that could possibly occur.

Log-odds: The log-odds, also known as the logit function, is the natural logarithm of the odds. In logistic regression, the log odds of the dependent variable are modeled as a linear combination of the independent variables and the intercept.

Coefficient: The logistic regression model's estimated parameters, show how the independent and dependent variables relate to one another.

Intercept: A constant term in the logistic regression model, which represents the log odds when all independent variables are equal to zero.

Maximum likelihood estimation: The method used to estimate the coefficients of the logistic regression model, which maximizes the likelihood of observing the data given the model.

Advantage of Logistics regression

Logistic regression offers numerous advantages, making it a widely used and versatile statistical method in various fields such as healthcare, finance, marketing, and more. Firstly, its interpretability is a key asset. The coefficients in logistic regression models directly indicate the strength and direction of the relationship between independent variables and the probability of a binary outcome. This transparency facilitates understanding and trust in the model's predictions, enabling stakeholders to make informed decisions based on the factors driving the outcomes. logistic regression provides a probabilistic interpretation of outcomes. Instead of simply classifying instances into discrete categories, it estimates the probability that a given input belongs to a particular class. This probability estimation allows decision-makers to assess risk levels and uncertainty associated with different scenarios, aiding in risk management and strategic planning.

Additionally, logistic regression is computationally efficient, particularly with large datasets, making it suitable for real-time or high-throughput applications. Its efficiency stems from its simplicity and linear nature, allowing for rapid training and inference compared to more complex algorithms.

Another advantage is its robust performance with small datasets. Logistic regression can yield reliable results even when data is limited, making it applicable in scenarios where data collection is constrained by factors such as cost, time, or availability.

Furthermore, logistic regression is less susceptible to multicollinearity compared to other models. Even if independent variables are correlated, logistic regression can still provide accurate estimates of the relationship between each predictor and the outcome, ensuring robustness in the presence of correlated features.

Lastly, logistic regression's ease of implementation and interpretation is noteworthy. Its straightforward mathematical formulation and intuitive graphical representation make it accessible to a wide range of users, including those without advanced statistical expertise, facilitating its adoption and use across various domains. Overall, these advantages position logistic regression as a valuable tool for predictive modeling and decision support in diverse applications.

CHAPTER 5

UML DIAGRAMS

UML stands for Unified Modeling Language. UML is a standardized general-purpose modeling language in the field of object-oriented software engineering. The standard is managed, and was created by, the Object Management Group. The goal is for UML to become a common language for creating models of object-oriented computer software. In its current form UML is comprised of two major components: a Meta-model and a notation. In the future, some form of method or process may also be added to; or associated with, UML.

The Unified Modeling Language is a standard language for specifying, Visualization, Constructing and documenting the artifacts of software system, as well as for business modeling and other non-software systems. The UML represents a collection of best engineering practices that have proven successful in the modeling of large and complex systems. The UML is a very important part of developing objects-oriented software and the software development process. The UML uses mostly graphical notations to express the design of software projects.

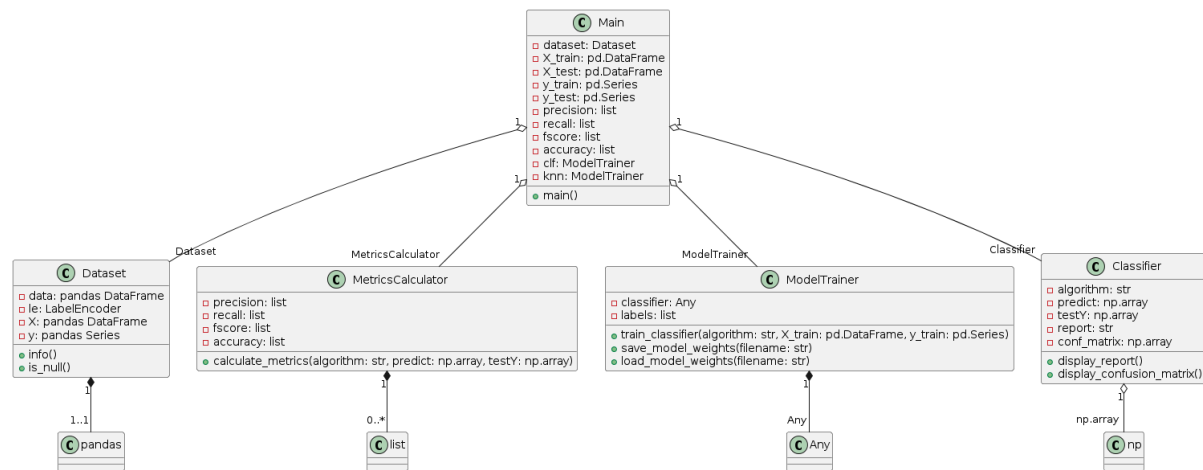
GOALS: The Primary goals in the design of the UML are as follows:

- Provide users a ready-to-use, expressive visual modeling Language so that they can develop and exchange meaningful models.
- Provide extendibility and specialization mechanisms to extend the core concepts.
- Be independent of particular programming languages and development process.
- Provide a formal basis for understanding the modeling language.
- Encourage the growth of OO tools market.
- Support higher level development concepts such as collaborations, frameworks, patterns and components.
- Integrate best practices.

Class Diagram

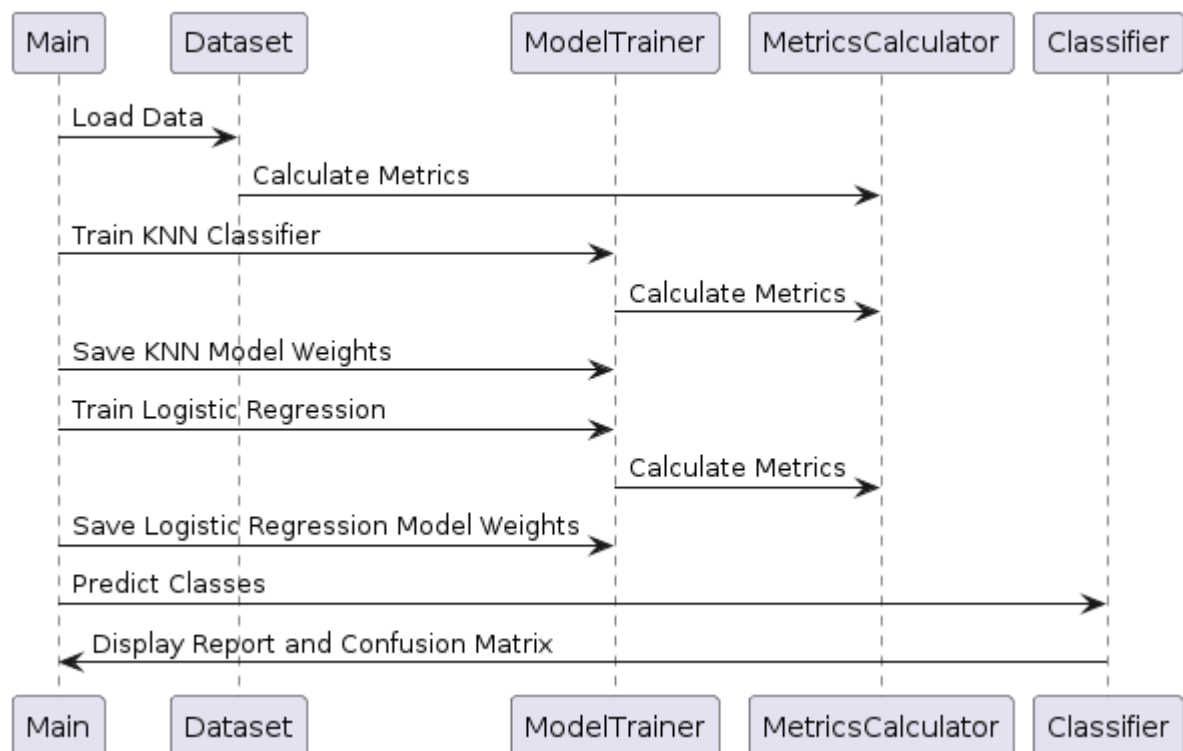
The class diagram is used to refine the use case diagram and define a detailed design of the system. The class diagram classifies the actors defined in the use case diagram into a set of interrelated classes. The relationship or association between the classes can be either an “is-a”

or “has-a” relationship. Each class in the class diagram may be capable of providing certain functionalities. These functionalities provided by the class are termed “methods” of the class. Apart from this, each class may have certain “attributes” that uniquely identify the class

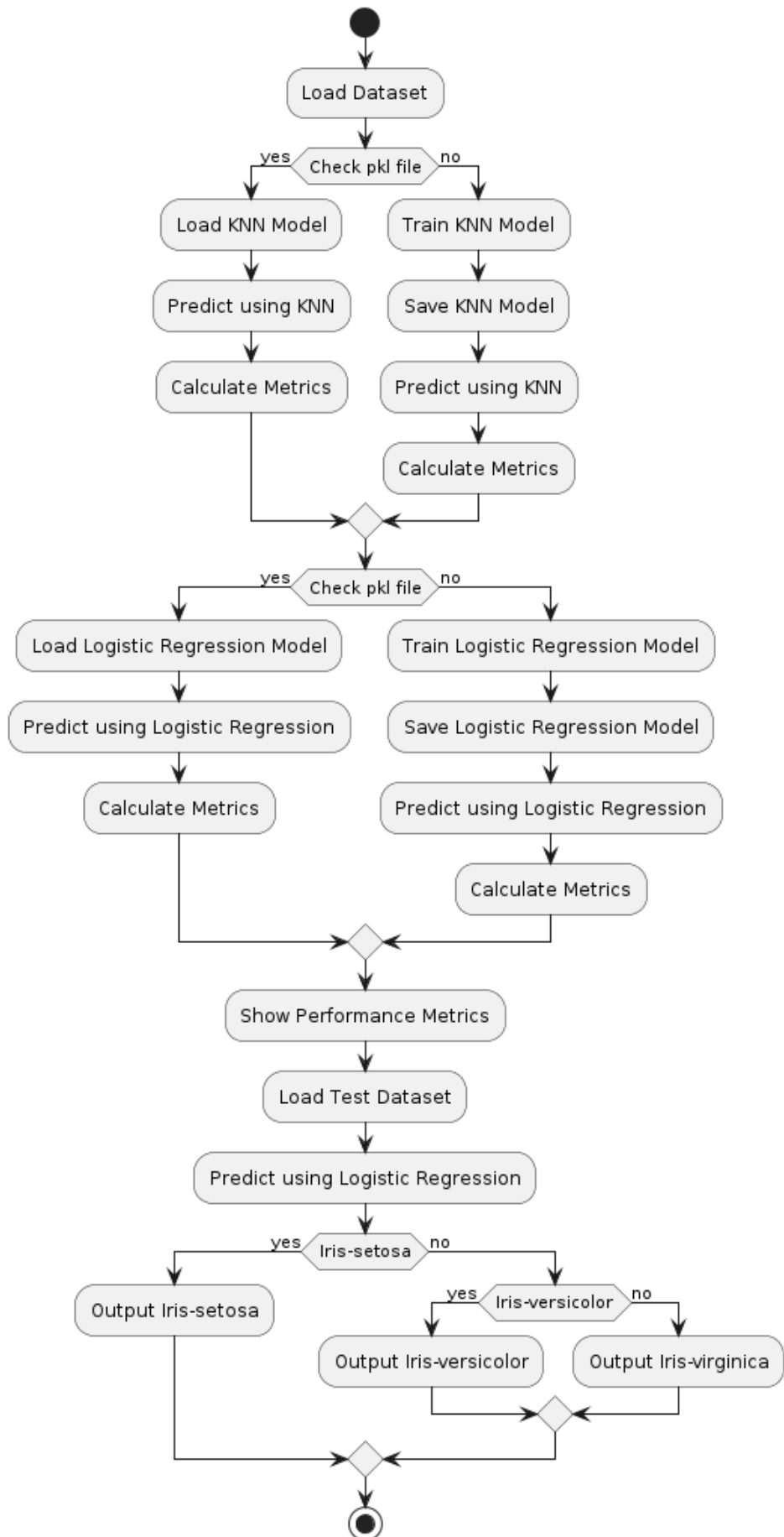


Sequence Diagram

A sequence diagram in Unified Modeling Language (UML) is a kind of interaction diagram that shows how processes operate with one another and in what order. It is a construct of a Message Sequence Chart. A sequence diagram shows, as parallel vertical lines (“lifelines”), different processes or objects that live simultaneously, and as horizontal arrows, the messages exchanged between them, in the order in which they occur. This allows the specification of simple runtime scenarios in a graphical manner.

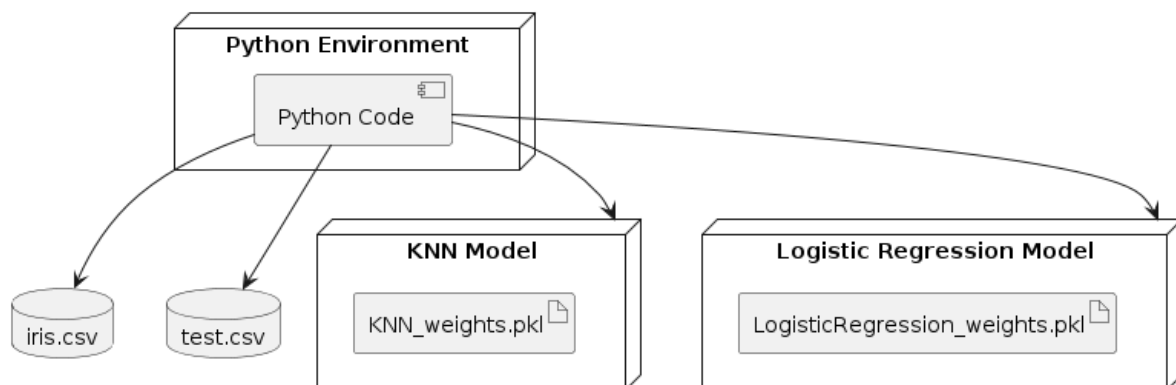


Activity diagram: Activity diagram is another important diagram in UML to describe the dynamic aspects of the system.



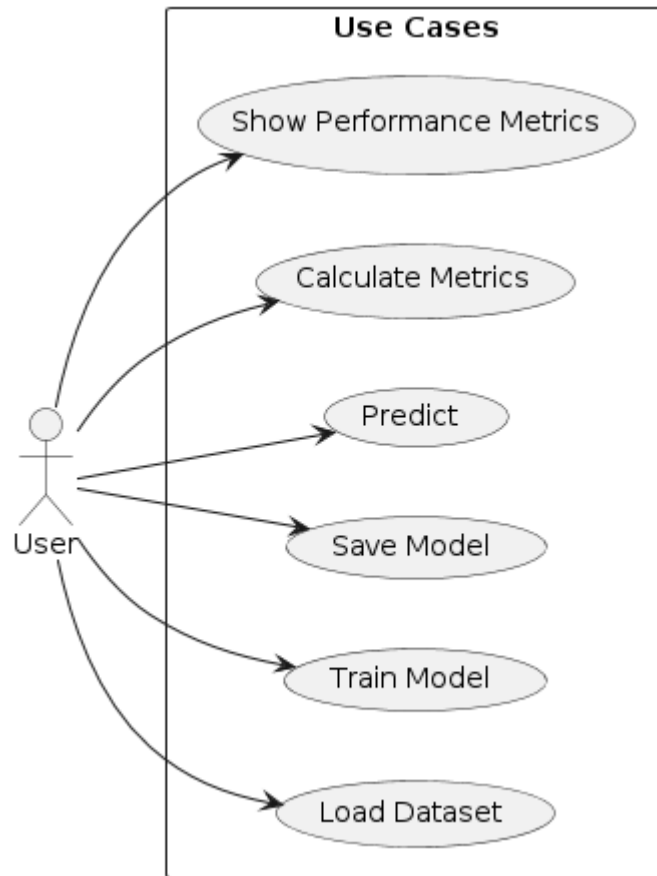
Deployment diagram

A deployment diagram in the Unified Modeling Language models the physical deployment of artifacts on nodes. To describe a web site, for example, a deployment diagram would show what hardware components (“nodes”) exist (e.g., a web server, an application server, and a database server), what software components (“artifacts”) run on each node (e.g., web application, database), and how the different pieces are connected (e.g., JDBC, REST, RMI). The nodes appear as boxes, and the artifacts allocated to each node appear as rectangles within the boxes. Nodes may have sub nodes, which appear as nested boxes. A single node in a deployment diagram may conceptually represent multiple physical nodes, such as a cluster of database servers.

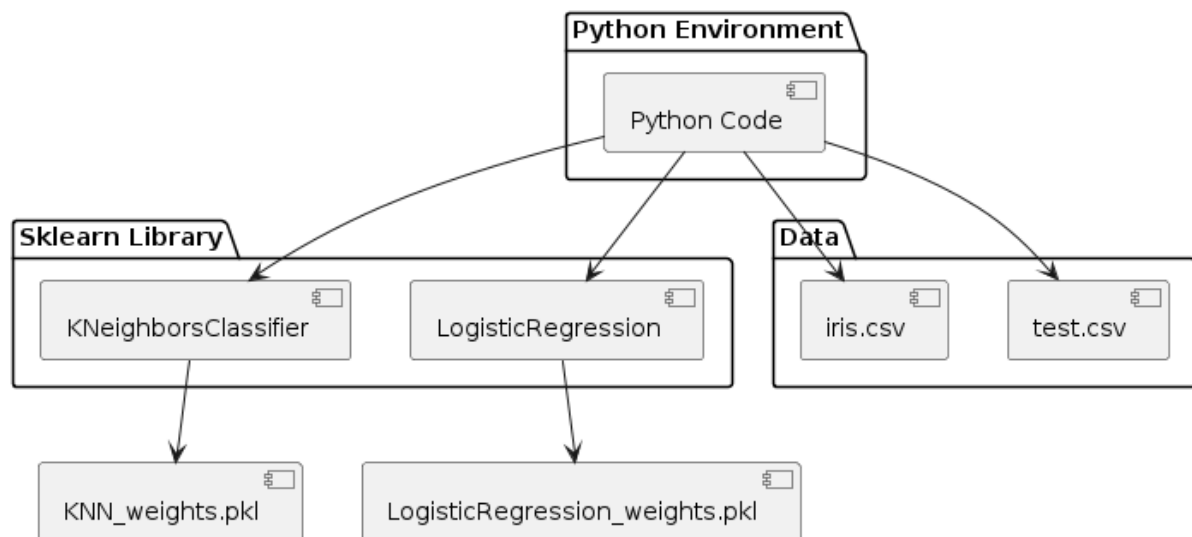


Use case diagram

A use case diagram in the Unified Modeling Language (UML) is a type of behavioral diagram defined by and created from a Use-case analysis. Its purpose is to present a graphical overview of the functionality provided by a system in terms of actors, their goals (represented as use cases), and any dependencies between those use cases. The main purpose of a use case diagram is to show what system functions are performed for which actor. Roles of the actors in the system can be depicted.



Component diagram: Component diagram describes the organization and wiring of the physical components in a system.



Data Flow Diagram (DFD):

In UML (Unified Modeling Language) is a graphical representation of the flow of data through a system. It's particularly useful for understanding the data inputs, outputs, processes, and storage within a system or software application. Here's

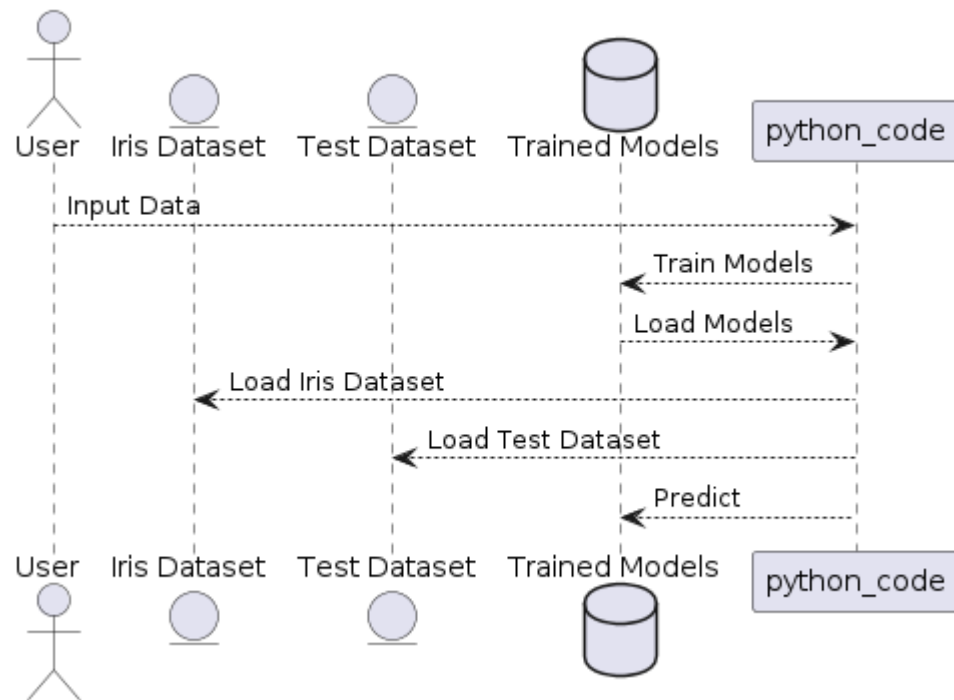
External Entities: These are entities outside the system that interact with it, such as users, other systems, or databases. They are represented as rectangles on the edges of the diagram.

Processes: Processes represent the actions or transformations that occur within the system. They are depicted as circles or ovals and typically have labels describing the action they perform on the data.

Data Stores: Data stores represent where data is stored within the system. They can be databases, files, or any other storage medium. Data stores are typically represented as rectangles.

Data Flows: Data flows represent the movement of data between external entities, processes, and data stores. They are depicted as arrows and indicate the direction of data flow.

Data Transformations: These represent the conversion or manipulation of data within a process. They can be depicted using labels on the data flow arrows or as separate process symbols.



CHAPTER 6

SOFTWARE ENVIRONMENT

What is Python?

Below are some facts about Python.

- Python is currently the most widely used multi-purpose, high-level programming language.
- Python allows programming in Object-Oriented and Procedural paradigms. Python programs generally are smaller than other programming languages like Java.
- Programmers have to type relatively less and indentation requirement of the language, makes them readable all the time.
- Python language is being used by almost all tech-giant companies like – Google, Amazon, Facebook, Instagram, Dropbox, Uber... etc.

The biggest strength of Python is huge collection of standard library which can be used for the following –

- Machine Learning
- GUI Applications (like Kivy, Tkinter, PyQt etc.)
- Web frameworks like Django (used by YouTube, Instagram, Dropbox)
- Image processing (like Opencv, Pillow)
- Web scraping (like Scrapy, BeautifulSoup, Selenium)
- Test frameworks
- Multimedia

Advantages of Python

Let's see how Python dominates over other languages.

1. Extensive Libraries

Python downloads with an extensive library and it contain code for various purposes like regular expressions, documentation-generation, unit-testing, web browsers, threading, databases, CGI, email, image manipulation, and more. So, we don't have to write the complete code for that manually.

2. Extensible

As we have seen earlier, Python can be extended to other languages. You can write some of your code in languages like C++ or C. This comes in handy, especially in projects.

3. Embeddable

Complimentary to extensibility, Python is embeddable as well. You can put your Python code in your source code of a different language, like C++. This lets us add scripting capabilities to our code in the other language.

4. Improved Productivity

The language's simplicity and extensive libraries render programmers more productive than languages like Java and C++ do. Also, the fact that you need to write less and get more things done.

5. IOT Opportunities

Since Python forms the basis of new platforms like Raspberry Pi, it finds the future bright for the Internet Of Things. This is a way to connect the language with the real world.

6. Simple and Easy

When working with Java, you may have to create a class to print 'Hello World'. But in Python, just a print statement will do. It is also quite easy to learn, understand, and code. This is why when people pick up Python, they have a hard time adjusting to other more verbose languages like Java.

7. Readable

Because it is not such a verbose language, reading Python is much like reading English. This is the reason why it is so easy to learn, understand, and code. It also does not need curly braces to define blocks, and indentation is mandatory. This further aids the readability of the code.

8. Object-Oriented

This language supports both the procedural and object-oriented programming paradigms. While functions help us with code reusability, classes and objects let us model the real world. A class allows the encapsulation of data and functions into one.

9. Free and Open-Source

Like we said earlier, Python is freely available. But not only can you download Python for free, but you can also download its source code, make changes to it, and even distribute it. It downloads with an extensive collection of libraries to help you with your tasks.

10. Portable

When you code your project in a language like C++, you may need to make some changes to it if you want to run it on another platform. But it isn't the same with Python. Here, you need to code only once, and you can run it anywhere. This is called Write Once Run Anywhere (WORA). However, you need to be careful enough not to include any system-dependent features.

11. Interpreted

Lastly, we will say that it is an interpreted language. Since statements are executed one by one, debugging is easier than in compiled languages.

Any doubts till now in the advantages of Python? Mention in the comment section.

Advantages of Python Over Other Languages

1. Less Coding

Almost all of the tasks done in Python requires less coding when the same task is done in other languages. Python also has an awesome standard library support, so you don't have to search for any third-party libraries to get your job done. This is the reason that many people suggest learning Python to beginners.

2. Affordable

Python is free therefore individuals, small companies or big organizations can leverage the free available resources to build applications. Python is popular and widely used so it gives you better community support.

The 2019 Github annual survey showed us that Python has overtaken Java in the most popular programming language category.

3. Python is for Everyone

Python code can run on any machine whether it is Linux, Mac or Windows. Programmers need to learn different languages for different jobs but with Python, you can professionally build

web apps, perform data analysis and machine learning, automate things, do web scraping and also build games and powerful visualizations. It is an all-rounder programming language.

Disadvantages of Python

So far, we've seen why Python is a great choice for your project. But if you choose it, you should be aware of its consequences as well. Let's now see the downsides of choosing Python over another language.

1. Speed Limitations

We have seen that Python code is executed line by line. But since Python is interpreted, it often results in slow execution. This, however, isn't a problem unless speed is a focal point for the project. In other words, unless high speed is a requirement, the benefits offered by Python are enough to distract us from its speed limitations.

2. Weak in Mobile Computing and Browsers

While it serves as an excellent server-side language, Python is much rarely seen on the client-side. Besides that, it is rarely ever used to implement smartphone-based applications. One such application is called Carbonnelle.

The reason it is not so famous despite the existence of Brython is that it isn't that secure.

3. Design Restrictions

As you know, Python is dynamically-typed. This means that you don't need to declare the type of variable while writing the code. It uses duck-typing. But wait, what's that? Well, it just means that if it looks like a duck, it must be a duck. While this is easy on the programmers during coding, it can raise run-time errors.

4. Underdeveloped Database Access Layers

Compared to more widely used technologies like JDBC (Java DataBase Connectivity) and ODBC (Open DataBase Connectivity), Python's database access layers are a bit underdeveloped. Consequently, it is less often applied in huge enterprises.

5. Simple

No, we're not kidding. Python's simplicity can indeed be a problem. Take my example. I don't do Java, I'm more of a Python person. To me, its syntax is so simple that the verbosity of Java code seems unnecessary.

This was all about the Advantages and Disadvantages of Python Programming Language.

History of Python

What do the alphabet and the programming language Python have in common? Right, both start with ABC. If we are talking about ABC in the Python context, it's clear that the programming language ABC is meant. ABC is a general-purpose programming language and programming environment, which had been developed in the Netherlands, Amsterdam, at the CWI (Centrum Wiskunde & Informatica). The greatest achievement of ABC was to influence the design of Python. Python was conceptualized in the late 1980s. Guido van Rossum worked that time in a project at the CWI, called Amoeba, a distributed operating system. In an interview with Bill Venners¹, Guido van Rossum said: "In the early 1980s, I worked as an implementer on a team building a language called ABC at Centrum voor Wiskunde en Informatica (CWI). I don't know how well people know ABC's influence on Python. I try to mention ABC's influence because I'm indebted to everything I learned during that project and to the people who worked on it. "Later on in the same Interview, Guido van Rossum continued: "I remembered all my experience and some of my frustration with ABC. I decided to try to design a simple scripting language that possessed some of ABC's better properties, but without its problems. So I started typing. I created a simple virtual machine, a simple parser, and a simple runtime. I made my own version of the various ABC parts that I liked. I created a basic syntax, used indentation for statement grouping instead of curly braces or begin-end blocks, and developed a small number of powerful data types: a hash table (or dictionary, as we call it), a list, strings, and numbers."

Python Development Steps

Guido Van Rossum published the first version of Python code (version 0.9.0) at alt.sources in February 1991. This release included already exception handling, functions, and the core data types of list, dict, str and others. It was also object oriented and had a module system.

Python version 1.0 was released in January 1994. The major new features included in this release were the functional programming tools lambda, map, filter and reduce, which Guido

Van Rossum never liked. Six and a half years later in October 2000, Python 2.0 was introduced. This release included list comprehensions, a full garbage collector and it was supporting unicode. Python flourished for another 8 years in the versions 2.x before the next major release as Python 3.0 (also known as "Python 3000" and "Py3K") was released. Python 3 is not backwards compatible with Python 2.x. The emphasis in Python 3 had been on the removal of duplicate programming constructs and modules, thus fulfilling or coming close to fulfilling the 13th law of the Zen of Python: "There should be one -- and preferably only one -- obvious way to do it." Some changes in Python 7.3:

Print is now a function.

- Views and iterators instead of lists
- The rules for ordering comparisons have been simplified. E.g., a heterogeneous list cannot be sorted, because all the elements of a list must be comparable to each other.
- There is only one integer type left, i.e., int. long is int as well.
- The division of two integers returns a float instead of an integer. "/" can be used to have the "old" behaviour.
- Text Vs. Data Instead of Unicode Vs. 8-bit

Purpose

We demonstrated that our approach enables successful segmentation of intra-retinal layers—even with low-quality images containing speckle noise, low contrast, and different intensity ranges throughout—with the assistance of the ANIS feature.

Python

Python is an interpreted high-level programming language for general-purpose programming. Created by Guido van Rossum and first released in 1991, Python has a design philosophy that emphasizes code readability, notably using significant whitespace.

Python features a dynamic type system and automatic memory management. It supports multiple programming paradigms, including object-oriented, imperative, functional and procedural, and has a large and comprehensive standard library.

- Python is Interpreted – Python is processed at runtime by the interpreter. You do not need to compile your program before executing it. This is similar to PERL and PHP.

- Python is Interactive – you can actually sit at a Python prompt and interact with the interpreter directly to write your programs.

Python also acknowledges that speed of development is important. Readable and terse code is part of this, and so is access to powerful constructs that avoid tedious repetition of code. Maintainability also ties into this may be an all but useless metric, but it does say something about how much code you have to scan, read and/or understand to troubleshoot problems or tweak behaviors. This speed of development, the ease with which a programmer of other languages can pick up basic Python skills and the huge standard library is key to another area where Python excels. All its tools have been quick to implement, saved a lot of time, and several of them have later been patched and updated by people with no Python background - without breaking.

Modules Used in Project

NumPy

NumPy is a general-purpose array-processing package. It provides a high-performance multidimensional array object, and tools for working with these arrays.

It is the fundamental package for scientific computing with Python. It contains various features including these important ones:

- A powerful N-dimensional array object
- Sophisticated (broadcasting) functions
- Tools for integrating C/C++ and Fortran code
- Useful linear algebra, Fourier transform, and random number capabilities

Besides its obvious scientific uses, NumPy can also be used as an efficient multi-dimensional container of generic data. Arbitrary datatypes can be defined using NumPy which allows NumPy to seamlessly and speedily integrate with a wide variety of databases.

Pandas

Pandas is an open-source Python Library providing high-performance data manipulation and analysis tool using its powerful data structures. Python was majorly used for data munging and preparation. It had very little contribution towards data analysis. Pandas solved this problem. Using Pandas, we can accomplish five typical steps in the processing and analysis of data, regardless of the origin of data load, prepare, manipulate, model, and analyze. Python with

Pandas is used in a wide range of fields including academic and commercial domains including finance, economics, Statistics, analytics, etc.

Matplotlib

Matplotlib is a Python 2D plotting library which produces publication quality figures in a variety of hardcopy formats and interactive environments across platforms. Matplotlib can be used in Python scripts, the Python and IPython shells, the Jupyter Notebook, web application servers, and four graphical user interface toolkits. Matplotlib tries to make easy things easy and hard things possible. You can generate plots, histograms, power spectra, bar charts, error charts, scatter plots, etc., with just a few lines of code. For examples, see the sample plots and thumbnail gallery.

For simple plotting the pyplot module provides a MATLAB-like interface, particularly when combined with IPython. For the power user, you have full control of line styles, font properties, axes properties, etc, via an object oriented interface or via a set of functions familiar to MATLAB users.

Scikit – learn

Scikit-learn provides a range of supervised and unsupervised learning algorithms via a consistent interface in Python. It is licensed under a permissive simplified BSD license and is distributed under many Linux distributions, encouraging academic and commercial use. Python

Python is an interpreted high-level programming language for general-purpose programming. Created by Guido van Rossum and first released in 1991, Python has a design philosophy that emphasizes code readability, notably using significant whitespace.

Python features a dynamic type system and automatic memory management. It supports multiple programming paradigms, including object-oriented, imperative, functional and procedural, and has a large and comprehensive standard library.

- Python is Interpreted – Python is processed at runtime by the interpreter. You do not need to compile your program before executing it. This is similar to PERL and PHP.
- Python is Interactive – you can actually sit at a Python prompt and interact with the interpreter directly to write your programs.

Python also acknowledges that speed of development is important. Readable and terse code is part of this, and so is access to powerful constructs that avoid tedious repetition of code.

Maintainability also ties into this may be an all but useless metric, but it does say something about how much code you have to scan, read and/or understand to troubleshoot problems or tweak behaviors. This speed of development, the ease with which a programmer of other languages can pick up basic Python skills and the huge standard library is key to another area where Python excels. All its tools have been quick to implement, saved a lot of time, and several of them have later been patched and updated by people with no Python background - without breaking.

Install Python Step-by-Step in Windows and Mac

Python a versatile programming language doesn't come pre-installed on your computer devices. Python was first released in the year 1991 and until today it is a very popular high-level programming language. Its style philosophy emphasizes code readability with its notable use of great whitespace.

The object-oriented approach and language construct provided by Python enables programmers to write both clear and logical code for projects. This software does not come pre-packaged with Windows.

How to Install Python on Windows and Mac

There have been several updates in the Python version over the years. The question is how to install Python? It might be confusing for the beginner who is willing to start learning Python but this tutorial will solve your query. The latest or the newest version of Python is version 3.7.4 or in other words, it is Python 3.

Note: The python version 3.7.4 cannot be used on Windows XP or earlier devices.

Before you start with the installation process of Python. First, you need to know about your System Requirements. Based on your system type i.e. operating system and based processor, you must download the python version. My system type is a Windows 64-bit operating system. So the steps below are to install python version 3.7.4 on Windows 7 device or to install Python 3. Download the Python Cheatsheet [here](#). The steps on how to install Python on Windows 10, 8 and 7 are divided into 4 parts to help understand better.

Download the Correct version into the system

Step 1: Go to the official site to download and install python using Google Chrome or any other web browser. OR Click on the following link: <https://www.python.org>



Now, check for the latest and the correct version for your operating system.

Step 2: Click on the Download Tab.



Step 3: You can either select the Download Python for windows 3.7.4 button in Yellow Color or you can scroll further down and click on download with respective to their version. Here, we are downloading the most recent python version for windows 3.7.4

Looking for a specific release?

Python releases by version number:

Release version	Release date		Click for more
Python 3.7.4	July 8, 2019	Download	Release Notes
Python 3.6.9	July 2, 2019	Download	Release Notes
Python 3.7.3	March 25, 2019	Download	Release Notes
Python 3.4.10	March 18, 2019	Download	Release Notes
Python 3.5.7	March 18, 2019	Download	Release Notes
Python 2.7.16	March 4, 2019	Download	Release Notes
Python 3.7.2	Dec. 24, 2018	Download	Release Notes

Step 4: Scroll down the page until you find the Files option.

Step 5: Here you see a different version of python along with the operating system.

Files

Version	Operating System	Description	MD5 Sum	File Size	GPG
Clipped source tarball	Source release		6811671e5b2db4ae7b9ab010f09be	23017663	SIG
XZ compressed source tarball	Source release		d33e4aan6097051c2eca45ee3604803	17131432	SIG
macOS 64-bit/32-bit installer	Mac OS X	for Mac OS X 10.6 and later	6428b4fa7583da91a442cbaace08ef	34898416	SIG
macOS 64-bit installer	Mac OS X	for OS X 10.9 and later	5dd605c38217a45773bf5e4a936d41f	28082845	SIG
Windows help file	Windows		d63999573a2c56b2ac56cade6b47cd2	8131761	SIG
Windows x86-64 embeddable zip file	Windows	for AMD64/EM64/x64	9800c3cfd3ec0b9abe83184a40729a2	7504291	SIG
Windows x86-64 executable installer	Windows	for AMD64/EM64/x64	a702b4b0ad76de9db3543a383e563400	2688368	SIG
Windows x86-64 web-based installer	Windows	for AMD64/EM64/x64	28c81c608be073ae8e53a3bd351b4bd2	1362904	SIG
Windows x86 embeddable zip file	Windows		9fab3b818841879fda94113574139d8	6741626	SIG
Windows x86 executable installer	Windows		33cc802942a54446a3d6451476394789	25663848	SIG
Windows x86 web-based installer	Windows		1b670cfa5d317df82c30983ea371d87c	1324608	SIG

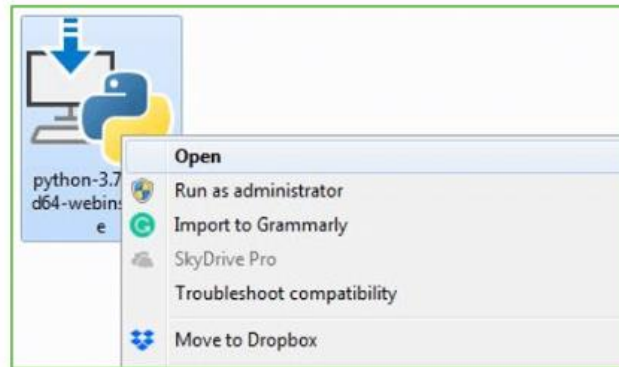
- To download Windows 32-bit python, you can select any one from the three options: Windows x86 embeddable zip file, Windows x86 executable installer or Windows x86 web-based installer.
- To download Windows 64-bit python, you can select any one from the three options: Windows x86-64 embeddable zip file, Windows x86-64 executable installer or Windows x86-64 web-based installer.

Here we will install Windows x86-64 web-based installer. Here your first part regarding which version of python is to be downloaded is completed. Now we move ahead with the second part in installing python i.e. Installation

Note: To know the changes or updates that are made in the version you can click on the Release Note Option.

Installation of Python

Step 1: Go to Download and Open the downloaded python version to carry out the installation process.



Step 2: Before you click on Install Now, Make sure to put a tick on Add Python 3.7 to PATH.



Step 3: Click on Install NOW After the installation is successful. Click on Close.



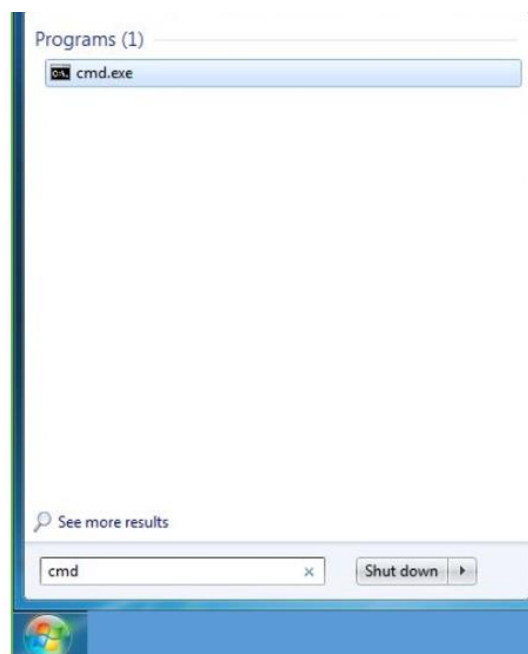
With these above three steps on python installation, you have successfully and correctly installed Python. Now is the time to verify the installation.

Note: The installation process might take a couple of minutes.

Verify the Python Installation

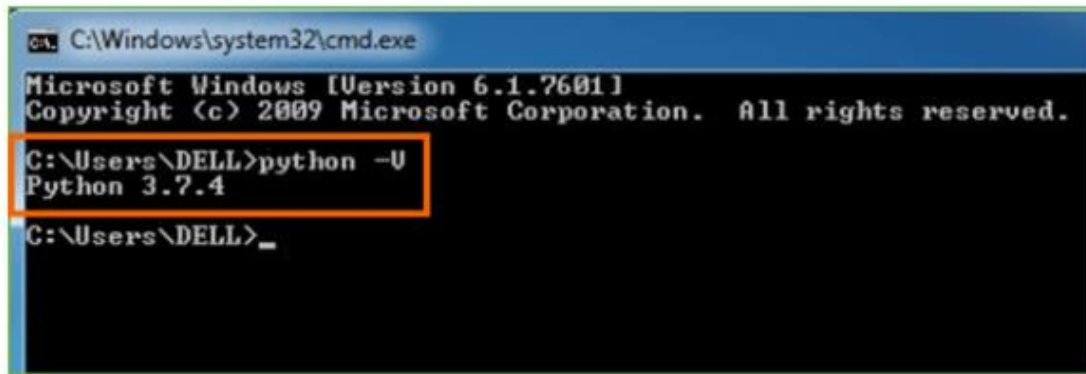
Step 1: Click on Start

Step 2: In the Windows Run Command, type “cmd”.



Step 3: Open the Command prompt option.

Step 4: Let us test whether the python is correctly installed. Type python -V and press Enter.



```
C:\Windows\system32\cmd.exe
Microsoft Windows [Version 6.1.7601]
Copyright (c) 2009 Microsoft Corporation. All rights reserved.

C:\Users\DELL>python -U
Python 3.7.4

C:\Users\DELL>_
```

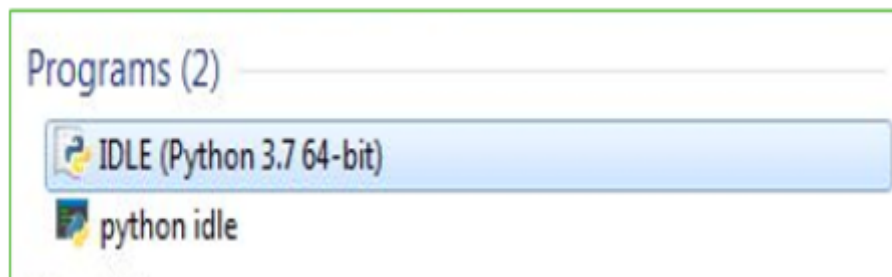
Step 5: You will get the answer as 3.7.4

Note: If you have any of the earlier versions of Python already installed. You must first uninstall the earlier version and then install the new one.

Check how the Python IDLE works

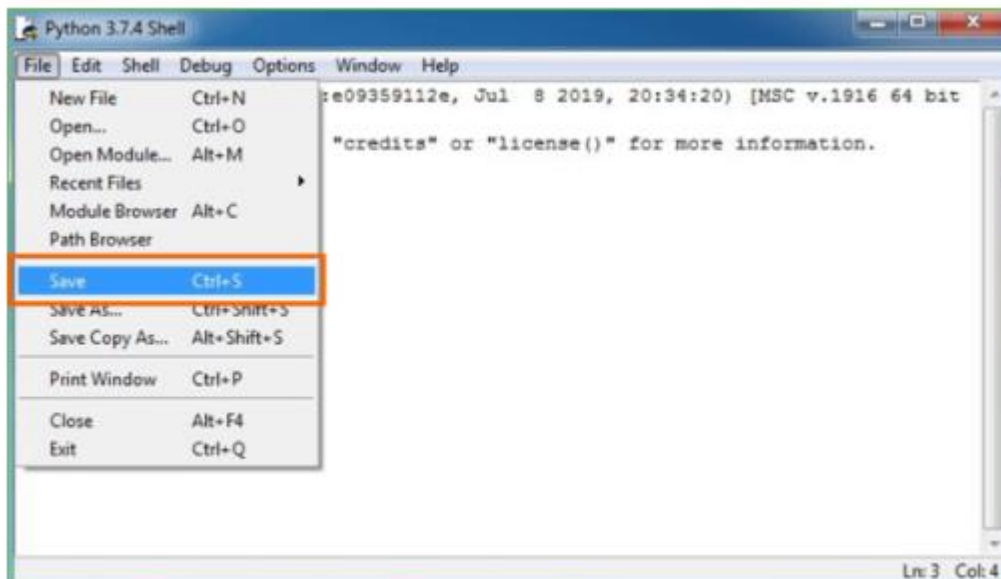
Step 1: Click on Start

Step 2: In the Windows Run command, type “python idle”.



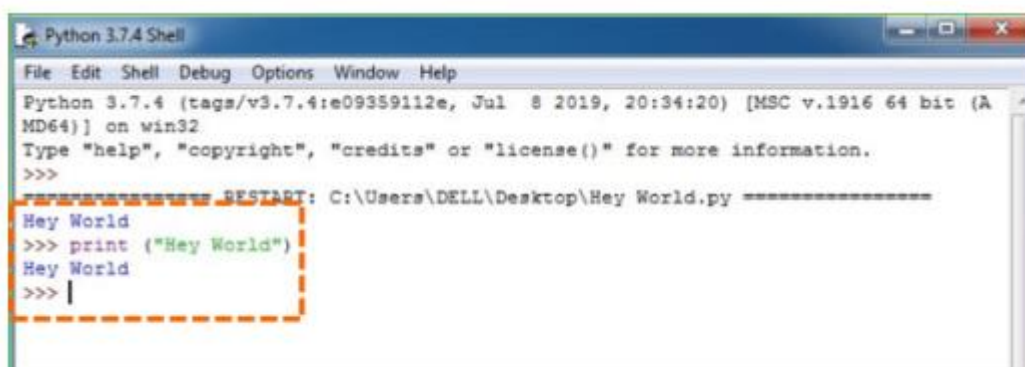
Step 3: Click on IDLE (Python 3.7 64-bit) and launch the program

Step 4: To go ahead with working in IDLE you must first save the file. Click on File > Click on Save



Step 5: Name the file and save as type should be Python files. Click on SAVE. Here I have named the files as Hey World.

Step 6: Now for e.g. enter print ("Hey World") and Press Enter.



You will see that the command given is launched. With this, we end our tutorial on how to install Python. You have learned how to download python for windows into your respective operating system.

Note: Unlike Java, Python does not need semicolons at the end of the statements otherwise it won't work.

CHAPTER 7

SYSTEM REQUIREMENTS

SOFTWARE REQUIREMENTS

The functional requirements or the overall description documents include the product perspective and features, operating system and operating environment, graphics requirements, design constraints and user documentation.

The appropriation of requirements and implementation constraints gives the general overview of the project in regard to what the areas of strength and deficit are and how to tackle them.

- Python IDLE 3.7 version (or)
- Anaconda 3.7 (or)
- Jupiter (or)
- Google colab

HARDWARE REQUIREMENTS

Minimum hardware requirements are very dependent on the particular software being developed by a given Enthought Python / Canopy / VS Code user. Applications that need to store large arrays/objects in memory will require more RAM, whereas applications that need to perform numerous calculations or tasks more quickly will require a faster processor.

- Operating system : Windows, Linux
- Processor : minimum intel i3
- Ram : minimum 4 GB
- Hard disk : minimum 250GB

CHAPTER 8

FUNCTIONAL REQUIREMENTS

OUTPUT DESIGN

Outputs from computer systems are required primarily to communicate the results of processing to users. They are also used to provide a permanent copy of the results for later consultation. The various types of outputs in general are:

- External Outputs, whose destination is outside the organization
- Internal Outputs whose destination is within organization and they are the
- User's main interface with the computer.
- Operational outputs whose use is purely within the computer department.
- Interface outputs, which involve the user in communicating directly.

OUTPUT DEFINITION

The outputs should be defined in terms of the following points:

- Type of the output
- Content of the output
- Format of the output
- Location of the output
- Frequency of the output
- Volume of the output
- Sequence of the output

It is not always desirable to print or display data as it is held on a computer. It should be decided as which form of the output is the most suitable.

INPUT DESIGN

Input design is a part of overall system design. The main objective during the input design is as given below:

- To produce a cost-effective method of input.
- To achieve the highest possible level of accuracy.
- To ensure that the input is acceptable and understood by the user.

INPUT STAGES

The main input stages can be listed as below:

- Data recording
- Data transcription
- Data conversion
- Data verification
- Data control
- Data transmission
- Data validation
- Data correction

INPUT TYPES

It is necessary to determine the various types of inputs. Inputs can be categorized as follows:

- External inputs, which are prime inputs for the system.
- Internal inputs, which are user communications with the system.
- Operational, which are computer department's communications to the system?
- Interactive, which are inputs entered during a dialogue.

INPUT MEDIA

At this stage choice has to be made about the input media. To conclude about the input media consideration has to be given to;

- Type of input
- Flexibility of format
- Speed
- Accuracy
- Verification methods
- Rejection rates
- Ease of correction
- Storage and handling requirements
- Security
- Easy to use
- Portability

Keeping in view the above description of the input types and input media, it can be said that most of the inputs are of the form of internal and interactive. As

Input data is to be directly keyed in by the user, the keyboard can be considered to be the most suitable input device.

ERROR AVOIDANCE

At this stage care is to be taken to ensure that input data remains accurate from the stage at which it is recorded up to the stage in which the data is accepted by the system. This can be achieved only by means of careful control each time the data is handled.

ERROR DETECTION

Even though every effort is made to avoid the occurrence of errors, still a small proportion of errors is always likely to occur, these types of errors can be discovered by using validations to check the input data.

DATA VALIDATION

Procedures are designed to detect errors in data at a lower level of detail. Data validations have been included in the system in almost every area where there is a possibility for the user to commit errors. The system will not accept invalid data. Whenever an invalid data is keyed in, the system immediately prompts the user and the user has to again key in the data and the system will accept the data only if the data is correct. Validations have been included where necessary.

The system is designed to be a user friendly one. In other words the system has been designed to communicate effectively with the user. The system has been designed with popup menus.

USER INTERFACE DESIGN

It is essential to consult the system users and discuss their needs while designing the user interface:

USER INTERFACE SYSTEMS CAN BE BROADLY CLASSIFIED AS:

- User initiated interface the user is in charge, controlling the progress of the user/computer dialogue. In the computer-initiated interface, the computer selects the next stage in the interaction.
- Computer initiated interfaces

In the computer-initiated interfaces the computer guides the progress of the user/computer dialogue. Information is displayed and the user response of the computer takes action or displays further information.

USER INITIATED INTERFACES

User initiated interfaces fall into two approximate classes:

- Command driven interfaces: In this type of interface the user inputs commands or queries which are interpreted by the computer.
- Forms oriented interface: The user calls up an image of the form to his/her screen and fills in the form. The forms-oriented interface is chosen because it is the best choice.

COMPUTER-INITIATED INTERFACES

The following computer – initiated interfaces were used:

- The menu system for the user is presented with a list of alternatives and the user chooses one; of alternatives.
- Questions – answer type dialog system where the computer asks question and takes action based on the basis of the users reply.

Right from the start the system is going to be menu driven, the opening menu displays the available options. Choosing one option gives another popup menu with more options. In this way every option leads the users to data entry form where the user can key in the data.

ERROR MESSAGE DESIGN

The design of error messages is an important part of the user interface design. As user is bound to commit some errors or other while designing a system the system should be designed to be helpful by providing the user with information regarding the error he/she has committed.

This application must be able to produce output at different modules for different inputs.

PERFORMANCE REQUIREMENTS

Performance is measured in terms of the output provided by the application. Requirement specification plays an important part in the analysis of a system. Only when the requirement specifications are properly given, it is possible to design a system, which will fit into required environment. It rests largely in the part of the users of the existing system to give the requirement specifications because they are the people who finally use the system. This is because the requirements have to be known during the initial stages so that the system can be

designed according to those requirements. It is very difficult to change the system once it has been designed and on the other hand designing a system, which does not cater to the requirements of the user, is of no use.

The requirement specification for any system can be broadly stated as given below:

- The system should be able to interface with the existing system
- The system should be accurate
- The system should be better than the existing system
- The existing system is completely dependent on the user to perform all the duties.

CHAPTER 9

SOURCE CODE

```
import numpy as np

import pandas as pd

import matplotlib.pyplot as plt

import warnings

warnings.filterwarnings('ignore')

import joblib

from sklearn.model_selection import train_test_split

from sklearn.metrics import accuracy_score

from sklearn.metrics import classification_report

from sklearn.naive_bayes import GaussianNB

from sklearn.metrics import confusion_matrix

import seaborn as sns

import os

from sklearn.linear_model import LogisticRegression

from sklearn.metrics import classification_report, confusion_matrix

import joblib

import os

from sklearn.preprocessing import LabelEncoder

from sklearn.metrics import precision_score

from sklearn.metrics import recall_score

from sklearn.metrics import f1_score

dataset = pd.read_csv("iris.csv")
```

```

dataset

dataset.info()

dataset.isnull().sum()


le=LabelEncoder()

dataset['Species']=le.fit_transform(dataset['Species'])

X= dataset.iloc[:,1:5]

X

y= dataset.iloc[:, -1]

y

# Split the dataset into training and testing sets

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)


labels = ["Iris-setosa",
          "Iris-versicolor",
          "Iris-virginica"]


#defining global variables to store accuracy and other metrics

precision = []

recall = []

fscore = []

accuracy = []


#function to calculate various metrics such as accuracy, precision etc

```

```

def calculateMetrics(algorithm, predict, testY):

    testY = testY.astype('int')

    predict = predict.astype('int')

    p = precision_score(testY, predict,average='macro') 100

    r = recall_score(testY, predict,average='macro') 100

    f = f1_score(testY, predict,average='macro') 100

    a = accuracy_score(testY,predict) 100

    accuracy.append(a)

    precision.append(p)

    recall.append(r)

    fscore.append(f)

    print(algorithm+' Accuracy  : '+str(a))

    print(algorithm+' Precision  : '+str(p))

    print(algorithm+' Recall    : '+str(r))

    print(algorithm+' FSCORE    : '+str(f))

    report=classification_report(predict, testY,target_names=labels)

    print("\n",algorithm+" classification report\n",report)

    conf_matrix = confusion_matrix(testY, predict)

    plt.figure(figsize =(5, 5))

    ax = sns.heatmap(conf_matrix, xticklabels = labels, yticklabels = labels, annot = True,
cmap="Blues" ,fmt ="g");

    ax.set_ylim([0,len(labels)])

    plt.title(algorithm+" Confusion matrix")

    plt.ylabel('True class')

    plt.xlabel('Predicted class')

```

```

plt.show()

# Check if the pkl file exists

if os.path.exists('KNN_weights.pkl'):

    # Load the model from the pkl file

    classifier= joblib.load('KNN_weights.pkl')

    predict = classifier.predict(X_test)

    calculateMetrics("KNN Classifier", predict, y_test)

else:

    classifier =
KNeighborsClassifier(weights='distance',algorithm='ball_tree',leaf_size=3,p=1,metric='mink
owski',)

    # Train the classifier on the training data

    classifier.fit(X_train, y_train)

    # Make predictions on the test data

    predict=classifier.predict(X_test)

    # Save the model weights to a pkl file

    joblib.dump(classifier, 'KNN_weights.pkl')

    print("KNN classifier_model trained and model weights saved.")

    calculateMetrics("KNeighborsClassifier", predict, y_test)

# Check if the pkl file exists

if os.path.exists('LogisticRegression_weights.pkl'):

    # Load the model from the pkl file

```



```

rf_classifier= joblib.load('LogisticRegression_weights.pkl')

predict = rf_classifier.predict(X_test)

calculateMetrics("LogisticRegression", predict, y_test)

else:

    clf = LogisticRegression()

    # Train the classifier on the training data

    clf.fit(X_train, y_train)

    # Make predictions on the test data

    predict=clf.predict(X_test)

    joblib.dump(clf, 'LogisticRegression_weights.pkl')

    print("LogisticRegression model trained and model weights saved.")

    calculateMetrics("LogisticRegression", predict, y_test)


#showing all algorithms performance values

columns = ["Algorithm Name","Precison","Recall","FScore","Accuracy"]

values = []

algorithm_names = ["KNeighborsClassifier", "LogisticRegression"]

for i in range(len(algorithm_names)):

    values.append([algorithm_names[i],precision[i],recall[i],fscore[i],accuracy[i]])

temp = pd.DataFrame(values,columns=columns)

temp

A="Iris-setosa"

```

```
B="Iris-versicolor"
```

```
C="Iris-virginica"
```

```
dataset = pd.read_csv(r"test.csv")
```

```
predict = clf.predict(dataset)
```

```
for i in range(len(predict)):
```

```
    if predict[i] == 0:
```

```
        print("{} :{} ".format(dataset.iloc[i,:],A))
```

```
    elif predict[i]== 1:
```

```
        print("{} :{} ".format(dataset.iloc[i, :],B))
```

```
    elif predict[i]== 2:
```

```
        print("{} :{} ".format(dataset.iloc[i, :],C))
```

CHAPTER 10

RESULTS AND DISCUSSION

10.1 Implementation Description

Implementing a machine learning approach to classify iris flowers based on petal and sepal measurements is essential for various applications in botany, horticulture, agriculture, and environmental science. This method involves utilizing a dataset containing measurements of iris flowers' petal length, petal width, sepal length, and sepal width, along with their corresponding species labels. The dataset is typically divided into training and testing sets to train and evaluate machine learning models.

Initially, the dataset is imported using libraries like NumPy and Pandas. Necessary preprocessing steps, such as handling missing values and encoding categorical variables like species labels, are performed. The dataset is then split into training and testing sets using the `train_test_split` function from scikit-learn. Two machine learning algorithms, namely K-Nearest Neighbors (KNN) and Logistic Regression, are commonly employed for classification tasks. These algorithms are trained on the training data and evaluated on the testing data to assess their performance metrics such as accuracy, precision, recall, and F1-score. For KNN classification, the `KNeighborsClassifier` from scikit-learn is utilized, with parameters such as `weights`, `algorithm`, and `leaf_size` specified based on experimentation or domain knowledge. Similarly, Logistic Regression, implemented through the `LogisticRegression` class, is trained on the training data.

After training the models, predictions are made on the testing data, and performance metrics are calculated using functions like `accuracy_score`, `precision_score`, `recall_score`, and `f1_score`. Additionally, a confusion matrix and classification report are generated to visualize the model's performance and provide detailed insights into its predictive capabilities.

Finally, the trained models can be saved using `joblib` for future use. Moreover, the models can be applied to classify iris flowers in new datasets or real-world scenarios. For instance, given a new dataset of iris flower measurements, the trained models can predict the species of each flower, facilitating species identification, biodiversity studies, breeding programs, and ecosystem monitoring.

Implementing a machine learning approach to classify iris flowers based on petal and sepal measurements involves data preprocessing, model training, evaluation, and application, providing valuable insights for various fields beyond botany.

10.2 Dataset Description

The dataset provided contains measurements of sepal length, sepal width, petal length, and petal width for various iris flowers, along with their corresponding species labels. This dataset is commonly used in machine learning and statistical analysis to develop models for classifying iris flowers based on their morphological characteristics.

The dataset comprises 150 instances, each representing an individual iris flower. Each instance contains five attributes: an ID number, sepal length in centimeters, sepal width in centimeters, petal length in centimeters, petal width in centimeters, and the species of the iris flower.

The sepal and petal measurements serve as the features used to characterize each iris flower, while the species label indicates the species to which the flower belongs. The species labels include three categories: *Iris-setosa*, *Iris-versicolor*, and *Iris-virginica*, representing different species within the iris genus.

Sepal length and width refer to the dimensions of the outermost part of the iris flower, known as the sepal. These measurements provide information about the size and shape of the sepal, which can vary between different species of iris.

Similarly, petal length and width describe the dimensions of the inner floral structure, known as the petal. These measurements capture characteristics such as petal size, shape, and color, which are important factors in distinguishing between iris species. This dataset offers a comprehensive overview of the morphological characteristics of iris flowers, facilitating the development and evaluation of classification models. Researchers and practitioners can use this dataset to explore patterns and relationships within the data, develop predictive models for species classification, and assess the performance of machine learning algorithms in accurately identifying iris species based on their physical attributes. This dataset serves as a valuable resource for studying the biodiversity of iris flowers and understanding the variations that exist within different species. By analyzing the distribution of sepal and petal measurements across species, researchers can gain insights into the evolutionary relationships between iris species and their ecological adaptations.

The dataset provides a rich source of information for studying iris flowers' morphology and species classification, making it a valuable asset for both scientific research and practical applications in fields such as botany, ecology, and machine learning.

10.3 Results and discussion

```

KNN Classifier Accuracy      : 100.0
KNN Classifier Precision    : 100.0
KNN Classifier Recall       : 100.0
KNN Classifier FSCORE       : 100.0

KNN Classifier classification report
      precision    recall  f1-score   support

 Iris-setosa      1.00      1.00      1.00        10
 Iris-versicolor  1.00      1.00      1.00         9
 Iris-virginica   1.00      1.00      1.00        11

 accuracy      1.00      1.00      1.00        30
 macro avg     1.00      1.00      1.00        30
 weighted avg  1.00      1.00      1.00        30

```

Figure 1: Classification report of KNN

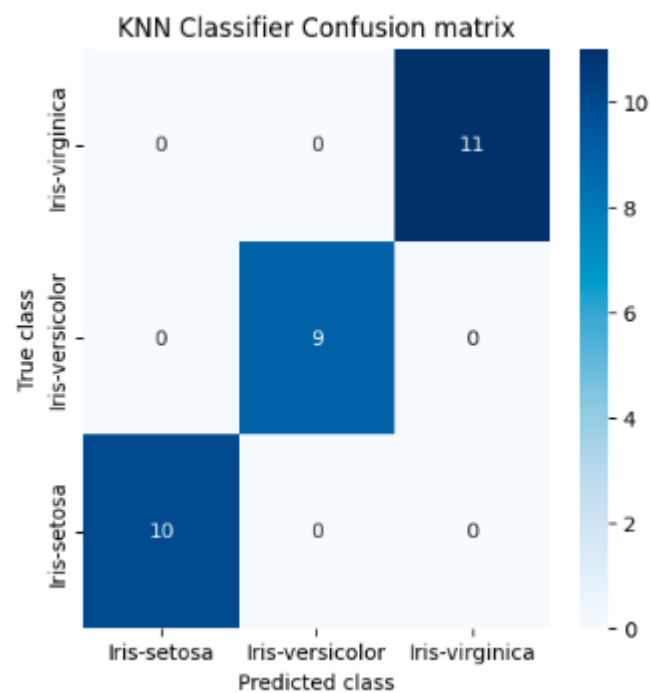


Figure 2: Confusion matrix of KNN

```

LogisticRegression Accuracy      : 100.0
LogisticRegression Precision     : 100.0
LogisticRegression Recall        : 100.0
LogisticRegression FSCORE       : 100.0

LogisticRegression classification report
              precision    recall  f1-score   support

   Iris-setosa              1.00      1.00      1.00        10
  Iris-versicolor           1.00      1.00      1.00         9
   Iris-virginica           1.00      1.00      1.00        11

   accuracy                   1.00      1.00      1.00        30
  macro avg                   1.00      1.00      1.00        30
 weighted avg                   1.00      1.00      1.00        30

```

Figure 3: Classification report of logistics Regression

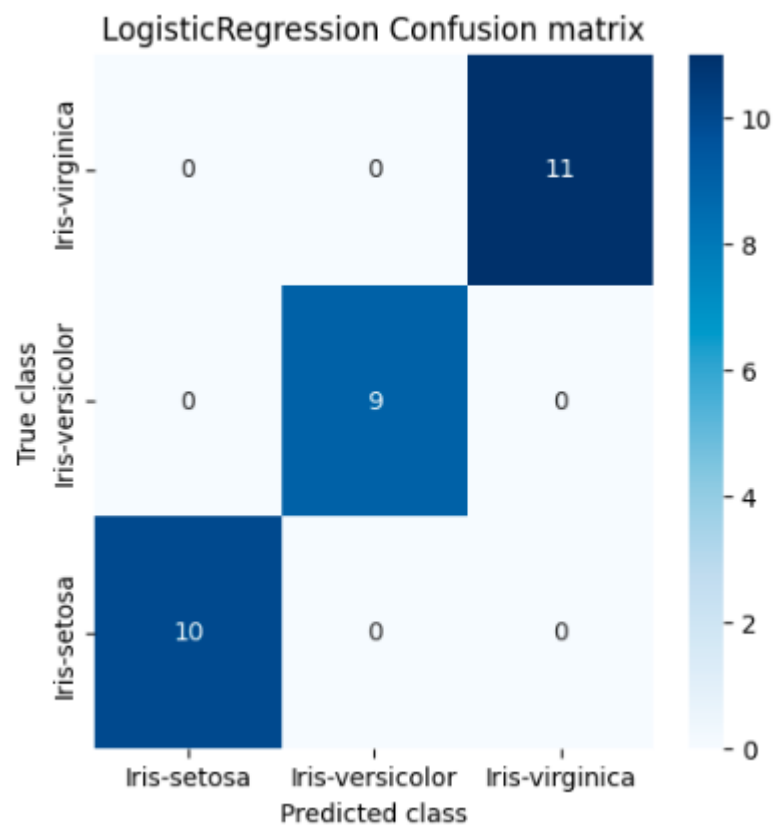


Figure 4: Confusion matrix of LR

	Algorithm Name	Precison	Recall	FScore	Accuracy
0	KNeighborsClassifier	100.0	100.0	100.0	100.0
1	LogisticRegression	100.0	100.0	100.0	100.0

Figure 5: Comparison

Figure 1 shows is a classification report for a K-nearest neighbors (KNN) classifier applied to the Iris flower dataset. The Iris dataset is a small dataset that is commonly used in machine learning tutorials. It has 150 data points, each of which has four features: sepal length, sepal width, petal length, and petal width. The data points are also labeled with one of three classes: Iris-setosa, Iris-versicolor, and Iris-virginica.

The KNN classifier is a simple algorithm that makes predictions based on the labels of the data points that are closest to the new data point. In the case of the Iris flower dataset, the KNN classifier would likely classify a new data point as Iris-setosa if it was closest to other data points that were labeled as Iris-setosa.

The classification report shows that the KNN classifier achieved 100% accuracy on the Iris flower dataset. This means that the classifier correctly classified all 150 data points. The report also shows that the classifier achieved 100% precision and 100% recall for all three classes. Precision is the proportion of positive predictions that were actually correct. Recall is the proportion of actual positive cases that were identified correct. It is important to note that the Iris flower dataset is a very small dataset. It is possible that the KNN classifier has simply overfit the data. Overfitting is a problem that occurs when a machine learning model memorizes the training data too well and does not generalize well to new data. In other words, the KNN classifier may have performed well on the Iris flower dataset, but it may not perform well on other datasets.

Figure 2 shows confusion matrix for a K-nearest neighbors (KNN) classifier applied to the Iris flower dataset. The Iris dataset is a classic dataset used in machine learning that contains 150 samples from three species of iris flowers: Iris-setosa, Iris-versicolor and Iris-virginica. Each flower is described by four features: sepal length, sepal width, petal length, and petal width.

The confusion matrix shows the number of correctly and incorrectly classified flower samples by the KNN model. Here's a breakdown of the information in the confusion matrix:

Rows represent the actual iris species (True Class).

Columns represent the iris species predicted by the KNN model (Predicted Class).

Figure 3 shows the results of a logistic regression model applied to the Iris flower dataset, a classic dataset used in machine learning. It appears the model achieved 100% accuracy on this dataset. Logistic Regression Accuracy: 100.0 - This indicates the model correctly classified all 30 data points in the Iris flower dataset.

Logistic Regression Precision: 100.0 - Precision refers to the ratio of true positives to the total number of positive predictions. A value of 1 here means the model identified only Iris flowers (positive cases) and none of the other plants (negative cases) as Iris flowers.

Logistic Regression Recall: 100.0 - Recall refers to the ratio of true positives to the total number of actual positive cases. A value of 1 here means the model identified all the Iris flowers (positive cases) in the dataset.

Logistic Regression F1-Score: 100.0 - The F1 score is a measure of a test's accuracy on a binary classification task. It considers both precision and recall. A value of 1 here means the model performed perfectly.

Figure 4 shows a confusion matrix is a table that is used to evaluate the performance of a classification model. It shows the number of correct and incorrect predictions made by the model. The confusion matrix in the image is a 3x3 matrix, because the logistic regression model is trying to classify the Iris flowers into three classes. The rows of the matrix represent the actual classes of the Iris flowers, and the columns of the matrix represent the classes that the model predicted.

The diagonal elements of the confusion matrix show the number of correct predictions. For example, the top-left element of the matrix shows that the model correctly predicted 10 Iris-setosa flowers as Iris-setosa.

The off-diagonal elements of the confusion matrix show the number of incorrect predictions. For example, the bottom-right element of the matrix shows that the model predicted 4 Iris-virginica flowers as Iris-versicolor.

Here is a more detailed explanation of the confusion matrix:

- **Iris-setosa:** The model correctly classified 10 Iris-setosa flowers and misclassified 0 flowers.
- **Iris-versicolor:** The model correctly classified 9 Iris-versicolor flowers and misclassified 1 Iris-versicolor flower as Iris-virginica.
- **Iris-virginica:** The model correctly classified 6 Iris-virginica flowers and misclassified 4 Iris-virginica flowers as Iris-versicolor.

The confusion matrix shows that the logistic regression model is performing well on the Iris flower dataset. The model is able to correctly classify the majority of flowers in all three classes.

Figure 5 shows Both Logistic Regression and KNeighbors Classifier algorithms achieved 100% accuracy on all metrics (precision, recall, F1-score, and accuracy). This suggests that both algorithms performed equally well on this dataset. But we choose logistics Regression as proposed algorithm because KNN have some limitations in this dataset i.e.

Data size: KNN can be computationally expensive for large datasets, as it needs to compare the new data point to all existing data points in order to make a prediction. Logistic regression is less computationally expensive.

Features: Logistic regression assumes a linear relationship between the features and the target variable. KNN can handle non-linear relationships as well.

CHAPTER 11

CONCLUSION AND FUTURE SCOPE

Conclusion:

In conclusion, the application of machine learning techniques for classifying iris flowers based on petal and sepal measurements offers a promising alternative to traditional methods. This approach not only streamlines the process of species identification in botany but also extends its utility across various domains such as horticulture, agriculture, and environmental science.

By leveraging supervised learning algorithms, our system demonstrates the capacity to autonomously discern patterns from input features, thereby reducing the reliance on manual measurements and expert knowledge. This automation not only accelerates the classification process but also enhances scalability and generalization capabilities, crucial for handling large datasets and subtle species differences.

The optimized model achieved through techniques like cross-validation and hyperparameter tuning ensures robust performance, enhancing the reliability of species classification. This reliability is particularly significant in fields like environmental science, where accurate identification of iris species contributes to ecosystem monitoring and conservation efforts.

Furthermore, the transferability of machine learning techniques demonstrated in this study highlights their potential for broader applications beyond botany. Industries ranging from healthcare to finance and marketing stand to benefit from the adaptable nature of these classification methods, paving the way for innovative solutions across diverse domains. In essence, the adoption of machine learning approaches for iris flower classification marks a significant advancement in botanical research and beyond. By combining automation with optimization, this methodology not only enhances efficiency and accuracy but also opens doors to interdisciplinary collaborations and novel applications.

Future Scope:

Future Scope

The machine learning approach to classifying iris flowers based on petal and sepal measurements offers several avenues for future research and development:

- 1. Enhanced Model Performance:**

- Future work could explore the integration of advanced machine learning techniques, such as deep learning and ensemble methods, to further improve the accuracy and robustness of iris species classification. Incorporating additional features or leveraging more complex models could lead to better generalization across different datasets.

2. Expanding to Other Plant Species:

- The methodology applied to iris flowers could be extended to classify other plant species with similar morphological traits. This expansion could contribute to a broader understanding of plant taxonomy and biodiversity, particularly in regions with rich flora.

3. Automated Plant Identification Systems:

- The development of portable or mobile-based plant identification systems using machine learning models could revolutionize botanical research, horticulture, and environmental monitoring. These systems could provide real-time classification and data collection in the field, aiding researchers and practitioners in their work.

4. Integration with Genomic Data:

- Future studies could integrate genomic or molecular data with morphological features like petal and sepal measurements. Combining these data types could enhance species identification accuracy and provide deeper insights into evolutionary relationships and genetic diversity.

5. Applications Beyond Botany:

- The techniques and models developed in this study could be adapted for use in other domains, such as healthcare, where similar classification problems arise. For instance, the principles applied here could be used for medical diagnostics, where classifying diseases based on various patient measurements is crucial.

6. Real-Time Monitoring and Conservation:

- Machine learning models can be employed in environmental monitoring programs to track the distribution and health of iris species in various

ecosystems. This real-time data could inform conservation efforts and help manage threatened species more effectively.

7. User-Friendly Tools for Non-Experts:

- Developing user-friendly software tools or applications that utilize machine learning for iris classification could empower non-experts, such as hobbyists or educators, to engage with botanical research and species identification.

8. Collaborative Research Platforms:

- Creating collaborative platforms where researchers can share data, models, and insights related to iris classification could accelerate progress in the field. These platforms could foster innovation and cross-disciplinary research efforts, leading to new discoveries and applications.

REFERENCES

- [1]. Ziauddin Ursani and David W. Corne , “A Novel Nonlinear Discriminant Classifier Trained by an Evolutionary Algorithm” DOI: 10.1145/3195106.3195132
- [2]. Detlef Nauck and Rudolf Kruse, “NEFCLASS-A Neuro-Fuzzzy approach for the classification of data” DOI: 10.1145/315891.316068
- [3] Jing FENG, Zhiwen WANG, Min ZHA and Xinliang CAO, “Flower Recognition Based on Transfer Learning and Adam Deep Learning Optimization Algorithm”. DOI: 10.1145/3366194.3366301
- [4] Rounq– Guo Huang, Sang-Hyeon Jin, Jung –Hyun Kim and KwangSeck Hong, “Flower Image Recognition Using Difference Image Entropy”. DOI: 10.1145/1821748.1821868
- [5] Shilpi Jain, V Poojitha, “By Using Neural Network Clustering tool in MATLAB Collecting the IRIS Flower”, Proc. IEEE , vol. 109, 2020.
- [6] M. M. Mijwil and R. A. Abttan, “Utilizing the Genetic Algorithm to Pruning the C4. 5 Decision Tree Algorithm,” Asian J. Appl. Sci. ISSN 2321– 0893, vol. 9, no. 1, 2021.
- [7] Rounq– Guo Huang, Sang-Hyeon Jin, Jung –Hyun Kim and Kwang- Seck Hong, “Flower Image Recognition Using Difference Image Entropy”. DOI: 10.1145/1821748.1821868
Academic Journal of Nawroz University (AJNU), Vol.11, No.4, 2022
 475
- [8] K R Rathy, Arya Vaishali, “Classification of Dataset using Efficient Neural Fuzzy Approach”, vol. 099, August 2019.
- [9] D. Decoste, E. Mjolsness. 2001. “State of the art and future prospects by using Machine Learning”, vol. 320, 2013.
- [10] Y. Lakhdoura and R. Elayachi, “Comparative Analysis of Random Forest and J48 Classifiers for ‘IRIS’ Variety Prediction,” Glob. J. Comput. Sci. Technol., 2020
- [11] Zebari, D. A., Abraham, A. R., Ibrahim, D. A., Othman, G. M., & Ahmed, F. Y. (2021). Analysis of Dense Descriptors in 3D Face Recognition. In *2021 IEEE 11th International Conference on System Engineering and Technology (ICSET)* (pp. 171-176). IEEE.
- [12] Abdulqadir, H. R., Abdulazeez, A. M., & Zebari, D. A. (2021). Data mining classification techniques for diabetes prediction. *Qubahan Academic Journal*, 1(2), 125-133.
- [13] Ibrahim, D. A., Zebari, D. A., Ahmed, F. Y., & Zeebaree, D. Q. (2021, November). Facial Expression Recognition Using Aggregated Handcrafted Descriptors based Appearance

Method. In 2021 IEEE 11th International Conference on System Engineering and Technology (ICSET) (pp. 177-182). IEEE.

Project Details				
Academic Year		2024-2025		
Title of the Project		CLASSIFYING IRIS FLOWERS: A MACHINE LEARNING APPROACH BASED ON PETAL AND SEPAL MEASUREMENTS		
Name of the Students and Hall Ticket No.		N.SAI VIGNYAN(21RA1A05B0) N.NITHIN KUMAR (21RA1A0573) S. RAVI TEJA (21RA1A0597)		
Name of the Guide		Mr.Ritesh kumar		
Project PO Mapping				
Name of Course From which Principles are applied in This Project	Related Course Outcomes Number	Description of the application	Page Number	Attained
Python Programming Software Engineering (C313)	C313.1	Students described the basis for their problem statement.	12	PO2
Machine Learning, Python Programming, Data Mining (C413, C411)	C322.2, C411.2	Students explained about Iris flower classification using machine learning approaches	13-14	PO1
Software Engineering, Python Programming (C313)	C313.3	Students identified the existing system and its Drawbacks and proposed a Solution to it.	17-21	P02, P03
Software Engineering (C313)	C313.1	Students identified the Hardware and Software required for the project.	55-56	PO5
Design Patterns, Software Engineering, DBMS (C313, C322)	C313.2, C222.3	Students explain the flow of the project using UML diagrams designed in STAR UML, ER diagram.	32-39	P03, P05, PO9, PSO3

Python Programming	C413.2, C411.2	Students explained about python programming language and developed code for the problem Statement.	61-66	PO3, PO4, PO5
Data Mining (C411)	C411.2	Students designed the modules for the solution of the problem.	67	PO2, PO3, PO4
Future Scope		Students explained about how they would like to further their project and develop it as their future scope.	74-76	PO12, PS02
Bibliography		Listed the references from which the literature was collected.	77	PO8, PO12
ENG		Prepared the thesis and intermediate progress reports and explained to the review panel. Also, continuously interact with guide and explain the progress.		PO9, PO10

SIGNATURE OF STUDENTS

SIGNATURE OF INTERNAL GUIDE

