INTERNSHIP PROJECT DOCUMENT I

ON

Iris Flower Classification using Machine Learning model – Model Training and Accuracy Evaluation

Submitted by

Swarupa Balaji

(Artificial Intelligence Intern)

Submitted to:

Founder- Kanduri Abhinay

CTO - Saniya Begam

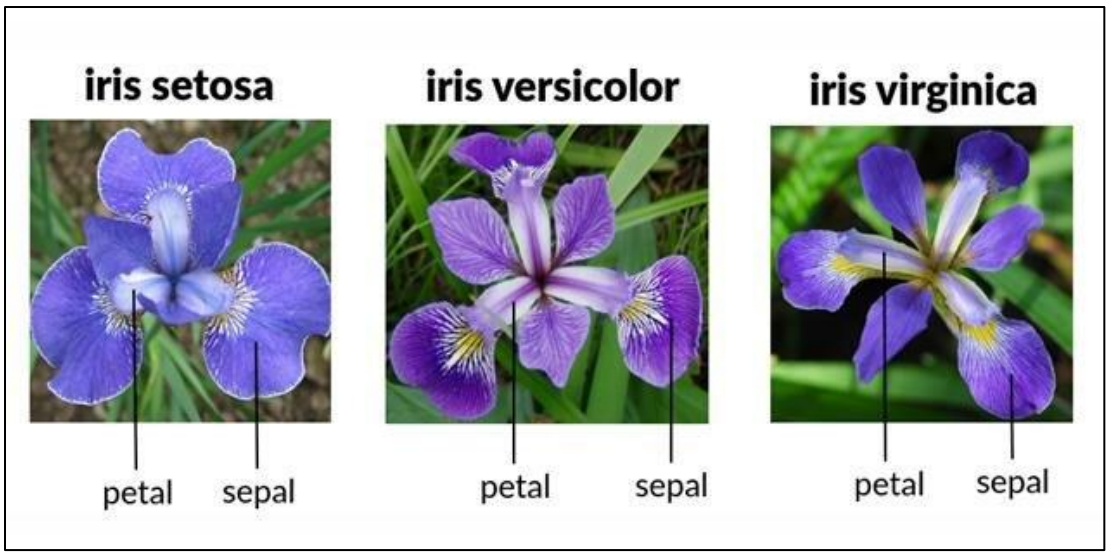
Team Lead – Surya Poojavi

PROJECT I

Title: Iris Flower Classification

Project Description:

Iris flower classification is a classic problem in machine learning, aimed at identifying species of iris flowers based on their features. The dataset includes measurements of sepal length, sepal width, petal length, and petal width for three iris species: Setosa, Versicolor, and Virginica. Using a logistic regression model, the classification involves training the algorithm to differentiate between these species based on the input features. The model outputs probabilities that help predict the class label, enabling accurate species identification. This application not only showcases basic machine learning principles but also the importance of feature selection in classification tasks.



Software and Tools Used in Project:

Google Colab: An online Jupyter notebook environment that provides free access to computing resources, including GPUs.

Python: Widely used for machine learning projects due to its simplicity and extensive libraries can be installed very easily.

Scikit-Learn: Scikit-learn is a powerful Python library for machine learning that provides simple and efficient tools for data mining, data analysis, and model training, featuring numerous algorithms and utilities for preprocessing and evaluation.

Packages and Libraries Installed:

Data Loading and Preprocessing:

NumPy: A library for numerical operations, essential for handling arrays and performing mathematical functions efficiently.

Pandas: For data manipulation and analysis, allowing easy handling of datasets in tabular form. Some techniques include: renaming columns, handling missing data, converting data formats and dtypes, loading data, grouping categories etc.

Data Visualization:

Matplotlib: Matplotlib is a Python library that allows users to create visualizations such as histograms, bar charts, scatter plots, and pie charts. It's a popular tool for data visualization, and is often used to create static, interactive, and animated visualizations.

Seaborn: It helps users create statistical graphics and visualizations for data analysis and interpretation. It's built on top of the Matplotlib library and integrates with Pandas data structures. Built-in themes and color palette allow users to develop attractive visuals.

Model Development:

Scikit-Learn: Scikit-learn is an open-source library for machine learning, providing efficient tools for classification, regression, clustering, and preprocessing, along with utilities for model evaluation and selection.

In this project, Scikit-learn has been used for:

- Model Selection
- Splitting training data and testing data
- Analysis of accuracy
- Metrics like precision and recall
- Evaluation with confusion matrix

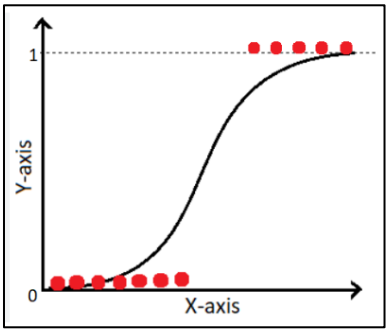
Algorithm used in Model:

The model used, Logistic Regression models the relationship between a dependent binary variable and one or more independent variables using the logistic function. The output is a probability value between 0 and 1, which can be mapped to two classes. It's widely used in fields like medicine, finance, and social sciences for tasks such as disease diagnosis, credit scoring, and marketing analysis.

Mathematical formula:

Logistic regression is based on the sigmoid function that is represented as:

$$G(x) = \sum_{i=1}^{N} \frac{1}{1+e^{-z}i}$$



Where,

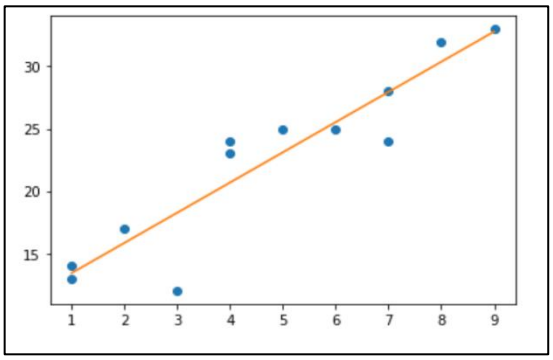
G(x) = Logistic Regression function

z = dot product of weights and input + bias

Summation of I = 1 to N given to denote number of dot products

The dot product between weights and input x with the addition of bias is a separate concept of linear regression which produces continuous outputs, unlike logistic regression. Formula is as follows:

$$z = \sum_{i=1}^{N} \omega_i x_i + b$$



Where,
F(x) = denotes linear regression
w and b = denote the weights and bias
Summation of I = 1 to N given to denote number of weights and input x.

Project Execution:

- Data is prepared by handling missing values and data transformation technique.
- Before applying the model, the data can be visualized using scatter plot to observe the datapoint placement of different types of iris flower.
- Data is split into training data that is used to train the model and test data for model evaluation.
- Logistic Regression model is trained with the training data and predictions read.
- Using the testing data, the model is tested to see its accuracy. A confusion matrix helps in finding times the model did correct or incorrect predictions with respect to the true output and predicted output.
- Accuracy score, precision and recall used to evaluate the model.

Scikit-Learn libraries used:

Train_test_split: used to split the data into training and test data with emphasis to proportion of division using train size or test size.

Linear_model – Logistic Regression: model is imported with least number of code, where model.fit is used to fit the training data and model.predict(test_input) can be used to predict the output.

Metrics: accuracy score and classification report that displays precision (ratio of true positive predictions to the total number of positive predictions made by the model.) and recall (ratio of true positive predictions to the total number of actual positive instances in the dataset.)

Confusion matrix: Evaluation function that show ratio of true positive, true negative, false positive and false negatives.

My observations from this project:

In the iris flower classification project using logistic regression, I observed that the model effectively predicted species based on features like sepal and petal dimensions. Logistic regression's interpretability helped in understanding feature importance. I learnt how important precision and recall are for model evaluation. Confusion matrix helped me compare the difference in the actual output and predicted outputs. Other classification methods, such as Support Vector Machines (SVM) and Decision Trees, offer distinct advantages. SVM excels in high-dimensional spaces and provides robust margin separation, while Decision Trees offer easy interpretability and flexibility in capturing non-linear relationships.

Sources:

- Kaggle
- Chatgpt
- Youtube
- Medium
- Blackbox.ai