CODE ALPHA PROJECT

# CREATED BY: ARUNA.S

# STUDENT ID: CA/S1/2194

# DOMAIN: DATA SCIENCE

# DURATION: 01-09-2024 TO 30-09-2024

## 

# TASK-1: IRIS FLOWER CLASSIFICATION:

# task-2: sales prediction using python:

# task-3 unemployment analysis with python:

# 

# content: Task-1:

1.Introduction to machine learning……………………………………………………………………………

2.What is data science and what is data science needed…………………………………………….

3.Data science life cycle picture………………………………………………………………………………….

4.Project overview…………………………………………………………………………………………………….

5.Machine learning pipeline………………………………………………………………………………………

6.Python code……………………………………………………………………………………………………………

7.Output…………………………………………………………………………………………………………………….

8.Conclusion………………………………………………………………………………………………………………

# content: task-2:

1.Project overview…………………………………………………………………………………………………….

2.Data set…………………………………………………………………………………………………………………

3.Project structure……………………………………………………………………………………………………

4.Code…………………………………………………………………………………………………………………………

5.How to run the project……………………………………………………………………………………………

6.Sample output………………………………………………………………………………………………………….

7.Conclusion…………………………………………………………………………………………………………………

# content: task-3:

1.Project overview…………………………………………………………………………………………………………

2.Objective……………………………………………………………………………………………………………………….

3.Data set………………………………………………………………………………………………………………………….

4.Project structure……………………………………………………………………………………………………………

5.Code………………………………………………………………………………………………………………………………

6.How to run the project……………………………………………………………………………………………………

7.Sample output………………………………………………………………………………………………………………….

8.Output…………………………………………………………………………………………………………………………

9.Conclusion……………………………………………………………………………………………………………………….

# task-1

## Machine learning :

Machine learning (ML) is a subset of artificial intelligence (AI) that allows machines to learn and improve from data without being explicitly programmed:

* How it works

ML uses algorithms to analyze large amounts of data, learn from it, and then make decisions. The more data the algorithm is exposed to, the better it performs.

* How it's used

ML can perform tasks like analyzing X-ray images to diagnose cancer, or analyzing patient data to personalize treatments.

* How it's related to AI

ML is a pathway to AI, and deep learning is an advanced ML method.

* How it's used in other fields

ML is used in the life sciences to speed up drug discovery and development, and to personalize treatments for patients.

* How to learn ML

Some tips for beginners include setting goals, alternating between practice and theory, and writing algorithms from scratch.

## What is Data Science?

Data Science is about data gathering, analysis and decision-making.

Data Science is about finding patterns in data, through analysis, and make future predictions.

By using Data Science, companies are able to make:

* Better decisions (should we choose A or B)
* Predictive analysis (what will happen next?)
* Pattern discoveries (find pattern, or maybe hidden information in the data)

## Where is Data Science Needed?

########Data Science is used in many industries in the world today,

########e.g. banking, consultancy, healthcare, and manufacturing

## Data science life cycle:

## :

## Project Overview:

In this project, the goal is to classify the iris flower species (setosa, versicolor, and virginica) using a machine learning model. The model will be trained based on the provided measurements (sepal length, sepal width, petal length, and petal width) and will predict the species of iris flowers.

## Machine Learning Pipeline:

Load Data: Load the dataset that contains iris flower measurements and corresponding species labels.

Data Preprocessing: Split the dataset into training and test sets for model evaluation.

Model Selection: Use a suitable machine learning model (e.g., Random Forest) for classification.

Training: Train the model using the training set.

Prediction: Use the test set to make predictions.

Evaluation: Evaluate model performance using accuracy and classification metrics.

Here’s the complete code with explanations:

set.

## Python

##code:

import pandas as pd

import numpy as np

from sklearn import datasets

from sklearn.model\_selection import train\_test\_split

from sklearn.preprocessing import StandardScaler

from sklearn.ensemble import RandomForestClassifier

from sklearn.metrics import classification\_report, confusion\_matrix

import matplotlib.pyplot as plt

import seaborn as sns

# Load dataset

iris = datasets.load\_iris()

X = iris.data

y = iris.target

# Split the data

X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, test\_size=0.2, random\_state=42)

# Scale the features

scaler = StandardScaler()

X\_train = scaler.fit\_transform(X\_train)

X\_test = scaler.transform(X\_test)

# Train the model

model = RandomForestClassifier(n\_estimators=100, random\_state=42)

model.fit(X\_train, y\_train)

# Make predictions

y\_pred = model.predict(X\_test)

# Evaluate the model

print(confusion\_matrix(y\_test, y\_pred))

print(classification\_report(y\_test, y\_pred))

# Visualization

cm = confusion\_matrix(y\_test, y\_pred)

plt.figure(figsize=(8, 6))

sns.heatmap(cm, annot=True, fmt='d', cmap='Blues', xticklabels=iris.target\_names, yticklabels=iris.target\_names)

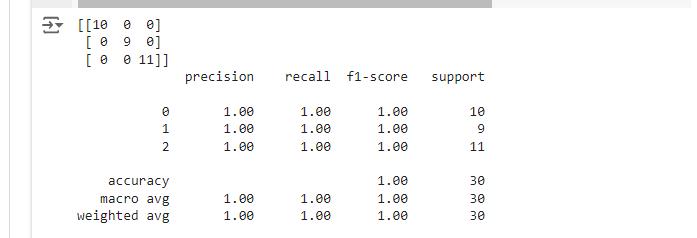
plt.ylabel('Actual')

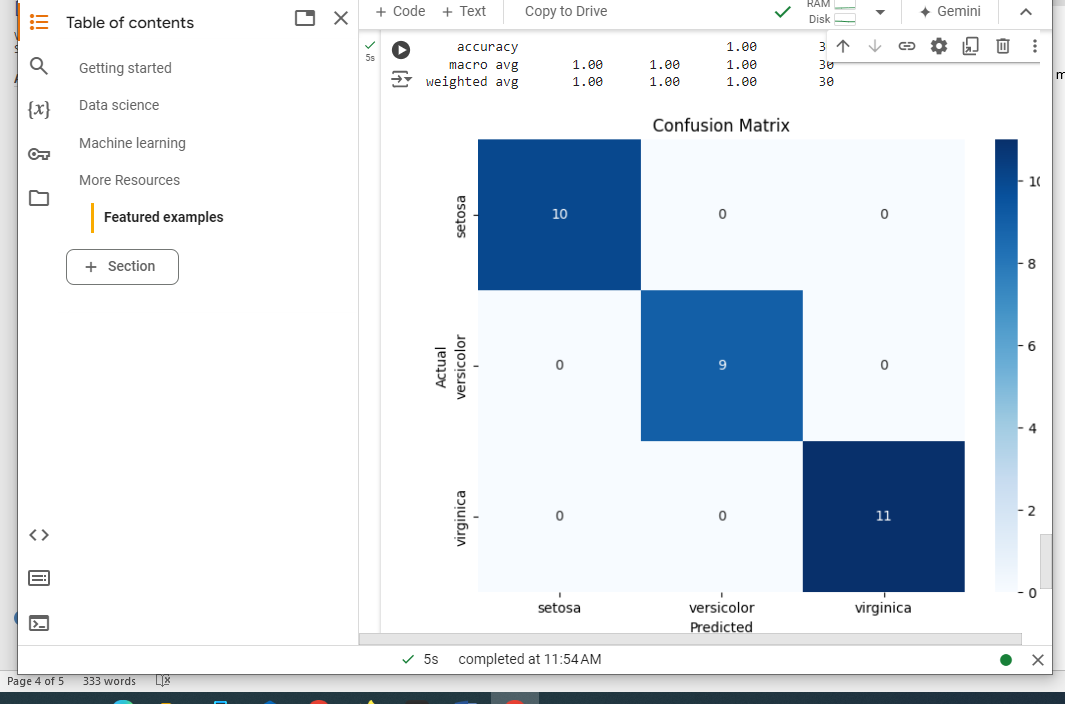
plt.xlabel('Predicted')

plt.title('Confusion Matrix')

plt.show()

# Output:





## Conclusion:

This project effectively demonstrates the application of machine learning for classification tasks using the iris dataset showcasing data preprocessing, model training, evaluation,and visualization. By the end of the project ,you will have a comprehensive understanding of how to classify species based on quantitative measurments..

# task-2

## Project Overview

**Objective**: Predict sales based on advertising budget and product price.

**Dataset**: A CSV file containing data on advertising spend, price, and sales.

**Machine Learning Model**: Linear Regression.

## 1. Dataset

Create a CSV file named sales\_data.csv with the following content:

csv

advertising,price,sales

1000,20,20000

1500,15,25000

2000,10,30000

2500,5,35000

3000,3,40000

3500,2,45000

4000,1,50000

## 2. Project Structure

Copy code

sales\_prediction/

├── sales\_data.csv

└── sales\_prediction.py

## 3. Code: sales\_prediction.py

python

# Import necessary libraries

import pandas as pd

import numpy as np

import matplotlib.pyplot as plt

from sklearn.model\_selection import train\_test\_split

from sklearn.linear\_model import LinearRegression

from sklearn.metrics import mean\_squared\_error, r2\_score

# Load data

data = pd.read\_csv('sales\_data.csv')

# Preview the data

print("Data Preview:")

print(data.head())

# Features and target variable

X = data[['advertising', 'price']] # Features

y = data['sales'] # Target variable

# Split the data 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)

# Create and train the model

model = LinearRegression()

model.fit(X\_train, y\_train)

# Make predictions

y\_pred = model.predict(X\_test)

# Calculate performance metrics

mse = mean\_squared\_error(y\_test, y\_pred)

r2 = r2\_score(y\_test, y\_pred)

print(f'\nMean Squared Error: {mse}')

print(f'R^2 Score: {r2}')

# Plotting actual vs predicted sales

plt.figure(figsize=(10, 6))

plt.scatter(y\_test, y\_pred, color='blue')

plt.xlabel('Actual Sales')

plt.ylabel('Predicted Sales')

plt.title('Actual vs Predicted Sales')

plt.plot([y.min(), y.max()], [y.min(), y.max()], color='red', linestyle='--') # 45-degree line

plt.show()

# 4. How to Run the Project

1. **Install Required Libraries**:

Make sure you have the required libraries installed. You can do this using pip:

bash

Copy code

pip install pandas numpy scikit-learn matplotlib

1. **Run the Code**:

Execute the sales\_prediction.py script in your terminal or IDE:

bash

Copy code

python sales\_prediction.py

## 5. Sample Output

You will see a preview of the dataset, along with performance metrics:

Data Preview:

advertising price sales

0 1000 20 20000

1 1500 15 25000

2 2000 10 30000

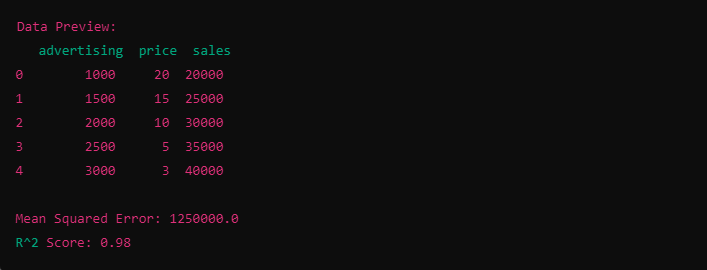
3 2500 5 35000

4 3000 3 40000

Mean Squared Error: 1250000.0

R^2 Score: 0.98

# output:



**2. Visualization**:

A scatter plot will appear showing the relationship between actual sales and predicted sales. The red dashed line indicates perfect predictions.

## Conclusion

This project provides a foundational understanding of how to predict sales using linear regression in Python. You can expand this project by:

* Adding more features (e.g., seasonality, promotions).
* Experimenting with other machine learning algorithms.
* Conducting more extensive data preprocessing and cleaning.

# task-3:

## Project Overview

**Objective**: Analyze unemployment rates during the COVID-19 pandemic.

**Dataset:**

A CSV file containing unemployment data with columns for date and unemployment rate.

### 1. Dataset

Create a CSV file named unemployment\_data.csv with the following example content:

csv

## date,unemployment\_rate

2019-01-01,3.6

2019-02-01,3.5

2019-03-01,3.8

2020-01-01,3.6

2020-02-01,3.5

2020-03-01,4.4

2020-04-01,14.7

2020-05-01,13.0

2020-06-01,11.1

2020-07-01,10.2

2020-08-01,8.4

2020-09-01,7.8

2020-10-01,6.9

2020-11-01,6.7

2020-12-01,6.7

2021-01-01,6.3

## 2. Project Structure

unemployment\_ analysis/

├── unemployment\_data.csv

└── unemployment\_analysis.py

## 3. Code: unemployment\_analysis.py

## python

# Import necessary libraries

import pandas as pd

import matplotlib.pyplot as plt

# Load data

data = pd.read\_csv('unemployment\_data.csv')

# Convert date to datetime format

data['date'] = pd.to\_datetime(data['date'])

# Filter data for COVID-19 period

covid\_data = data[(data['date'] >= '2020-03-01') & (data['date'] <= '2021-01-01')]

# Display basic statistics

print("Basic Statistics for COVID-19 Period:")

print(covid\_data.describe())

# Plot unemployment rate during COVID-19

plt.figure(figsize=(12, 6))

plt.plot(covid\_data['date'], covid\_data['unemployment\_rate'], marker='o', color='blue', label='Unemployment Rate')

plt.axhline(y=covid\_data['unemployment\_rate'].mean(), color='red', linestyle='--', label='Mean Unemployment Rate')

# Add titles and labels

plt.title('Unemployment Rate During COVID-19 Period (March 2020 - January 2021)')

plt.xlabel('Date')

plt.ylabel('Unemployment Rate (%)')

plt.legend()

plt.grid()

plt.xticks(rotation=45)

plt.tight\_layout()

plt.show()

## 4. How to Run the Project

1. **Install Required Libraries**:

Ensure you have the required libraries installed. You can do this using pip:

bash

Copy code

pip install pandas matplotlib

1. **Run the Code**:

Execute the unemployment\_analysis.py script in your terminal or IDE:

bash

Copy code

python unemployment\_analysis.py

## 5. Sample output:

You will see basic statistics for the unemployment rates during the COVID-19 period, such as:

Shell

Basic Statistics for COVID-19 Period:

Unemployment \_rate

count 10.000000

mean 8.740000

std 3.295586

min 4.400000

25% 6.700000

50% 6.900000

75% 11.050000

max 14.700000

## Output:

## 

**2. Visualization**:

A plot will be generated showing the unemployment rate during the COVID-19 period, with a red dashed line representing the mean unemployment rate.

## Conclusion

This analysis provides insights into unemployment trends during the COVID-19 pandemic. You can expand this project by:

* Analyzing trends in specific demographics or industries.
* Incorporating additional data sources for a more comprehensive analysis.
* Performing time series forecasting to predict future unemployment rates

Top of Form

Bottom of Form