TRAINING 103 DAILY DIARY

Creating a daily diary focused on Python and Machine Learning (ML) over 20 working days is an excellent way to structure learning and track progress

Day 1: Introduction to Python

Create and Run your First Python Program on Terminal
Once you have Python installed, you can run the program by following these steps:

1. Open a text editor (e.g., Notepad on Windows, TextEdit on macOS, or any code editor like VS Code, PyCharm, etc.).
2. Copy the code:- **print('Hello World')** above and paste it into the text editor.
3. Save the file with .**py extension** (e.g., **Hello.py**).
4. Open the terminal.
5. Run the program by **pressing Enter**.

Topics Covered:

- Python syntax, variables, and data types.
- Basic operations: arithmetic, logical, and comparison operators.

Key Notes:

- Variables are dynamically typed in Python.
- Common data types: int, float, str, bool, list, tuple, dict, set.
- Example:
- x = 10
- y = 5
  print(x + y) # Output: 15

Day 2: Control Structures

Topics Covered:

- If-else statements.
- Loops: for and while.

Key Notes:

- Use if-elif-else for decision making.
- Example:
- for i in range(1, 6):
- if i % 2 == 0:
  print(f"{i} is even")

Day 3: Functions

Topics Covered:

- Creating and using functions.
- Passing arguments and returning values.

Key Notes:

- Example:
- def greet(name):
- return f"Hello, {name}!"
print(greet("Alice"))

Day 4: File Handling

Topics Covered:

- Reading and writing files.

Key Notes:

- Example:
- with open('sample.txt', 'w') as file:
file.write("Hello, File Handling!")

Day 5: Error and Exception Handling

Topics Covered:

- try, except, and finally blocks.

Key Notes:

- Example:
- try:
- result = 10 / 0
- except ZeroDivisionError:
print("Cannot divide by zero!")

Intermediate Python and Data Analysis

Day 6: NumPy and Pandas

Topics Covered:

- NumPy: Arrays and basic operations.
- Pandas: Series and DataFrames.

Key Notes:

- Example:
- import numpy as np
- arr = np.array([1, 2, 3])
print(arr.sum())
import pandas as pd

```
df = pd.DataFrame({'A': [1, 2], 'B': [3, 4]})
print(df)
```

Day 7: Data Visualization

Topics Covered:

- Matplotlib: Line plots, bar charts.
- Seaborn: Advanced plots.

```
[5]: import numpy as np
     import pandas as pd
     import matplotlib.pyplot as plt
     import seaborn as sns

[6]: df=pd.read_csv("latest Covid-19 India Status1 (2).xls")

[7]: df.head()
```

[7]:

	State/UTs	Total Cases	Active	Deaths	Active Ratio (%)	Death Ratio (%)
0	Maharashtra	6122893	117869	123857	1.93	2.02
1	Kerala	3011694	108400	14108	3.60	0.47
2	Karnataka	2862338	39626	35601	1.38	1.24
3	Tamil Nadu	2506848	34076	33196	1.36	1.32
4	Andhra Pradesh	1911231	32356	12919	1.69	0.68

```
[8]: df.shape
```

Key Notes:

- Example:
- import matplotlib.pyplot as plt
- plt.plot([1, 2, 3], [4, 5, 6])
  plt.show()

Day 8: Data Preprocessing

Topics Covered:

- Handling missing values and categorical encoding.

Key Notes:

- Example:

```
df.fillna(0, inplace=True)
```

Day 9: Exploratory Data Analysis (EDA)

Topics Covered:

- Summarizing datasets.

Key Notes:

- Example:

df.describe()

Day 10: Debugging

Topics Covered:

- Debugging tools like pdb.

Key Notes:

- Example:

import pdb; pdb.set_trace()

Machine Learning Foundations

Day 11: ML Concepts

Topics Covered:

- Supervised vs. Unsupervised Learning.

Key Notes:

- Supervised Learning: Predict target variables.
- Unsupervised Learning: Find patterns in data.

Day 12: Linear Regression

Topics Covered:

- Fit and evaluate linear models.

Key Notes:

- Example:
- from sklearn.linear_model import LinearRegression
- model = LinearRegression()
  model.fit(X_train, y_train)

Day 13: Logistic Regression

Topics Covered:

- Binary classification.

Key Notes:

- Example:
- from sklearn.linear_model import LogisticRegression
  model = LogisticRegression()

Day 14: Evaluation Metrics

Topics Covered:

- Accuracy, Precision, Recall, F1-Score.

Key Notes:

- Use classification_report in scikit-learn.

Day 15: Decision Trees and Random Forests

Topics Covered:

- Tree-based models.

Key Notes:

- Example:
- from sklearn.ensemble import RandomForestClassifier
- clf = RandomForestClassifier()
  clf.fit(X_train, y_train)

Advanced ML and Web Integration

Day 16: Unsupervised Learning

Topics Covered:

- Clustering (K-Means).

Key Notes:

- Example:
- from sklearn.cluster import KMeans
- kmeans = KMeans(n_clusters=3)
  kmeans.fit(X)

Day 17: Dimensionality Reduction

Topics Covered:

- PCA for reducing feature dimensions.

Key Notes:

- Example:
- from sklearn.decomposition import PCA
- pca = PCA(n_components=2)
  reduced_data = pca.fit_transform(X)

Day 18: Neural Networks

Topics Covered:

- TensorFlow/Keras basics.

Key Notes:

- Example:
- from keras.models import Sequential
- model = Sequential([...])
  model.compile(...)

Day 19: Flask Basics

Topics Covered:

- Web app framework.

Key Notes:

- Example:
- from flask import Flask, render_template
- app = Flask(__name__)
-
- @app.route('/')
- def home():
- return "Hello, Flask!"
-

app.run()

Day 20: Pickle Library

Topics Covered:

- Save and load models.

Key Notes:

- Example:

```
import pickle

with open('model.pkl', 'wb') as file:
    pickle.dump(model, file)


with open('model.pkl', 'rb') as file:
    loaded_model = pickle.load(file)
```

What is Flask?
Flask is a web framework that allows developers to build lightweight web applications quickly and easily with Flask Libraries. It was developed by Armin Ronacher, leader of the International Group of Python Enthusiasts(POCCO). It is basically based on the WSGI toolkit and Jinja2 templating engine.

Should I learn HTML for Flask?
Both are highly recommended in case you are learning to develop web applications.

Is Flask open source?
Yes, Flask and Django both are Free Open Source, Python-based web frameworks that are used for building web applications.

How do I start a Flask project?
Here's a simplified version of the steps to start a Flask project:

1. Install Python.
2. Install Flask by running "pip install flask" in the terminal or command prompt.
3. Create a virtual environment by running "python -m venv myenv".
4. Activate the virtual environment by running "myenv\Scripts\activate.bat" on Windows or "source myenv/bin/activate" on macOS or Linux.

5. Create a Flask app by importing Flask and creating a new instance of the Flask class.
6. Define routes by decorating a function with "@app.route('/myurl')" and defining the function to return data or HTML.
7. Run the app by running "flask run" in the terminal or command prompt.

What is the default port of Flask?
By default, many Flask applications run on port 5000.