Agromet Data Analysis using Python

B Sudarsan Patro
Scientist –D

INSAT AWS & RADIATION LAB
SID CRS IMD Pune

1. Is data processing necessary ?

2. Geopandas and Catropy difference?

Outline

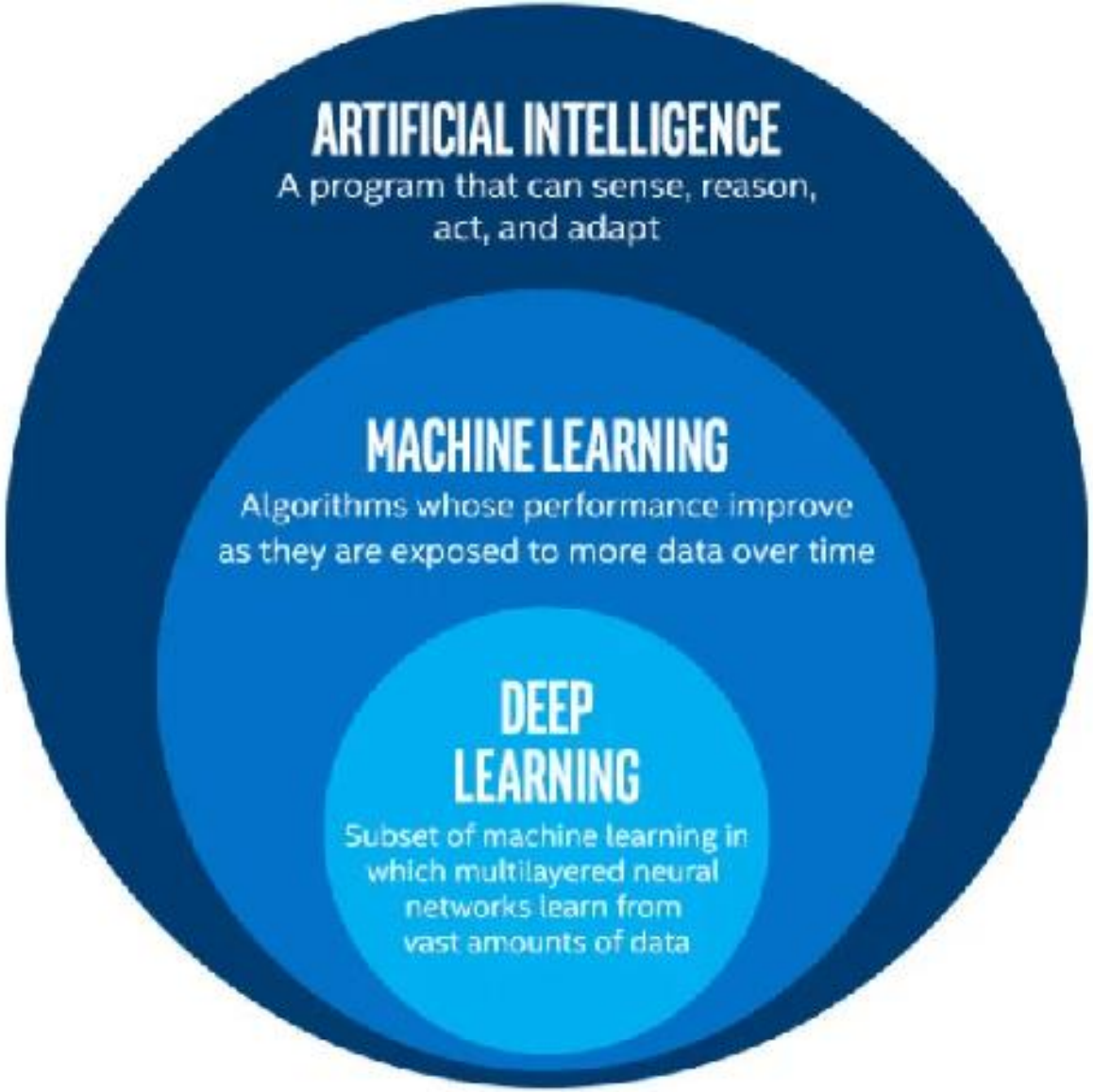
1. AIML in Weather using Python
2. Data Preprocessing & importance
3. Anantapur data
4. File processing using python
5. Case study

i. Anatapur data

ii. Soil Moisture sensor testing using ML/Python

iii. Pune Tmax using Geopandas

Introduction



Introduction.....

Artificial Intelligence

Machine Learning

Deep Learning

Data Pre-processing

Python

Statistics, Probability, Calculus, Linear algebra Probability Theory

Data Preprocessing techniques:

1. Handling Missing Values
2. Outlier Detection and Handling
3. Encoding Categorical Data
5. Feature Engineering
6. Dimensionality Reduction
7. Data Splitting (Training , Testing, Validation)
8. Data Normalization
9. Feature Scaling for Specific Algorithms

Documentation

https://pandas.pydata.org/docs

Installation

```
pip install pandas
```

Basic Pandas Operations

Importing Pandas

```
import pandas as pd
```

Creating a DataFrame

```
data = {'Column1': [value1, value2, ...], 'Column2': [value1, value2, ...]}
df = pd.DataFrame(data)
```

Reading Data from a File

```
df = pd.read_csv('data.csv')
```

Basic Pandas Operations….

Displaying DataFrame Information

```
print(df.head())  # Displays the first few rows
print(df.info())  # Displays information about the DataFrame
```

Selecting Columns

```
column = df['Column1']
```

Filtering Data

```
filtered_df = df[df['Column1'] > 35]
```

Basic Pandas Operations….

Grouping and Aggregation

```
grouped = df.groupby('Category')['Value'].mean()
```

Handling Missing Data

```
df.dropna()  # Drop rows with missing values
df.fillna(value)  # Fill missing values with a specific value
```

Data Visualization

```
import matplotlib.pyplot as plt
df.plot(kind='scatter', x='X', y='Y')
plt.show()
```

Basic Pandas Operations….

Grouping and Aggregation

```
grouped = df.groupby('Category')['Value'].mean()
```

Handling Missing Data

```
df.dropna()  # Drop rows with missing values
df.fillna(value)  # Fill missing values with a specific value
```

Data Visualization

```
import matplotlib.pyplot as plt
df.plot(kind='scatter', x='X', y='Y')
plt.show()
```

Operations….

-Time Series Analysis

- Merging and Joining DataFrames

- Reshaping Data

- Working with Multi-Index DataFrames

- Custom Functions

…… >

Python Libraries

Additional Libraries & Pandas works well with other libraries like

NumPy: NumPy is a Python library for numerical computations, particularly for handling arrays and matrices efficiently. It provides a wide range of mathematical functions and operations.

Matplotlib: Matplotlib is a Python library for creating static, animated, and interactive visualizations in a wide range of formats, including 2D and 3D plots, charts, and graphs. It is often used in scientific and data analysis applications to visualize data and results.

Seaborn: Seaborn is a Python data visualization library that is built on top of Matplotlib. It provides a high-level interface for creating informative and attractive statistical graphics. Seaborn is particularly useful for visualizing complex datasets and statistical relationships in a simple and aesthetically pleasing manner. It offers built-in themes and color palettes to enhance the visual appeal of plots.

and

Scikit-Learn often referred to as sklearn, is a popular machine learning library in Python. It provides a wide range of tools and algorithms for machine learning tasks such as classification, regression, clustering, dimensionality reduction, and more. Scikit-Learn is widely used in both academia and industry for building and evaluating machine learning models, as it offers a consistent and user-friendly API that makes it easier to work with various machine learning algorithms and datasets.

GeoPandas

1. GeoPandas:

Purpose: GeoPandas is primarily used for working with geospatial vector data, such as shapefiles, GeoJSON, and other vector formats.

Functionality: It provides a convenient and efficient way to read, manipulate, analyze, and visualize geospatial data. You can perform operations like spatial joins, attribute queries, and geometry transformations on vector data.

Integration with Pandas: GeoPandas is built on top of the Pandas library, allowing users to leverage Pandas DataFrame functionality for geospatial data.

Data Structures: GeoPandas mainly deals with two primary data structures: GeoDataFrame (for vector data) and GeoSeries (for individual geometries).

Visualization: It provides basic plotting and visualization capabilities for geospatial data but may not be as extensive as dedicated mapping libraries like Cartopy.

Cartopy

2. Cartopy:

Purpose: Cartopy is focused on cartographic projections and mapping. It is used for creating maps and visualizations of geospatial data on different map projections.

Functionality: It offers tools for defining and customizing map projections, adding geospatial data (such as points, lines, or polygons) to maps, and creating map layouts.

Map Projections: Cartopy provides a wide range of map projections for various regions of the Earth.

Integration with Matplotlib: It seamlessly integrates with Matplotlib, allowing you to create complex and customized map visualizations.

Data Structures: Cartopy doesn't have its own data structures for geospatial data; instead, it relies on other libraries like Matplotlib or GeoPandas to handle the data.

Agromet Data Analysis: Anantapur 43238(1983-2020)

- **index**: This column might represent an index or identifier for each row in the dataset.
- **year**: The year when the data was recorded.
- **week**: The week number within the year when the data was recorded.
- **date**: The specific date when the data was recorded.
- **month**: The month when the data was recorded.
- **db1, wb1, db2, wb2**: These columns may represent different temperature measurements (e.g., dry bulb and wet bulb temperatures) at two different locations or conditions (1 and 2).
- **max**: Maximum temperature recorded.
- **min**: Minimum temperature recorded.
- **gmt**: Some kind of time-related measurement, possibly Greenwich Mean Time (GMT).
- **st5hr1, st10hr1, st20hr1, st5hr2, st10hr2, st20hr2**: These columns could represent various soil temperature measurements at different depths (e.g., 5 cm, 10 cm, 20 cm) and possibly at two different locations or conditions.
- **vp1, vp2**: These columns might represent vapor pressure measurements at two different locations or conditions.
- **rh1, rh2**: Relative humidity measurements at two different locations or conditions.
- **ws**: Wind speed.
- **wdhr1, wdhr2**: Wind direction at two different hours or conditions.
- **bss**: Possibly a measurement related to sunshine duration.
- **rain**: Amount of rainfall.
- **pievp**: Possibly another measurement related to evaporation or vapor pressure.
- **panevp**: Another measurement related to evaporation or vapor pressure.