Internship Weekly Report – Week 2

◈ Title Page

- **Name:** Sandeep Ravaji Patel
- **Domain:** Data Science
- **Week Number:** Week 2

◈ Task Description

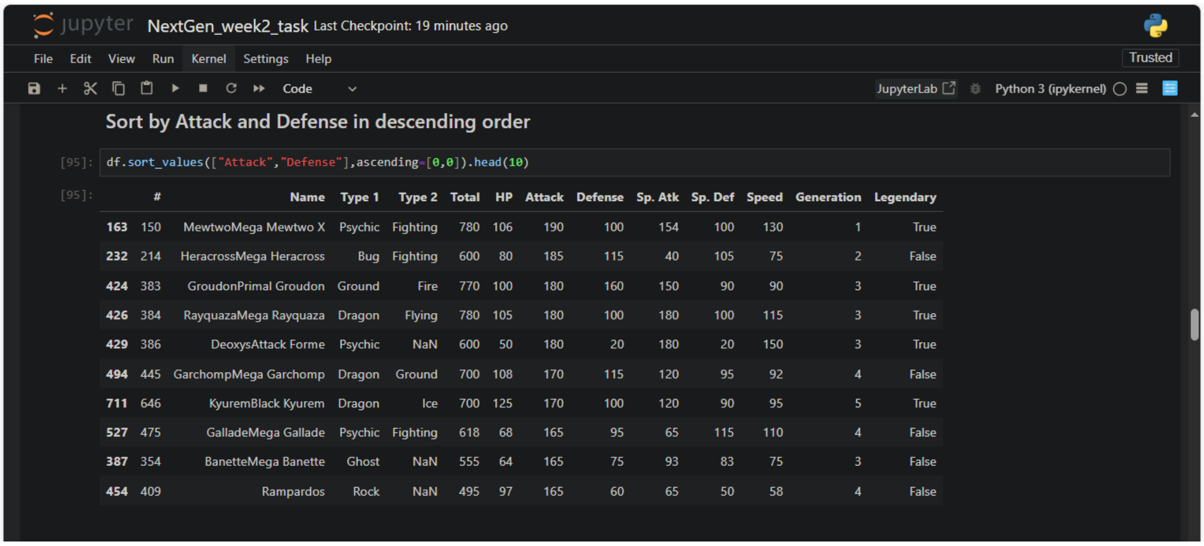
Objective:
To develop proficiency in data handling using Pandas and NumPy, focusing on operations like filtering, grouping, sorting, and analyzing datasets from real-world CSV files.

Tasks Completed:

1. **Data Import and Exploration:**
   - Imported CSV files using Pandas (read_csv) and NumPy (loadtxt).
   - Explored dataset structure (e.g., Pokémon stats) and displayed top rows.
2. **Data Filtering and Sorting:**
   - Filtered data based on conditions (e.g., Pokémon of type "Grass" or with high HP).
   - Sorted data by columns like "Attack," "Defense," and "Total" in ascending/descending order.
3. **Descriptive Statistics:**
   - Used describe() to summarize numerical columns (mean, min, max, etc.).
   - Grouped data by categories (e.g., Pokémon types) to calculate averages.
4. **Advanced Grouping and Aggregation:**
   - Grouped by "Type 1" and "Type 2" to analyze mean stats (e.g., highest average "Total" by type).
   - Counted Pokémon per type using groupby and custom aggregations.
5. **Data Conversion and Cleaning:**
   - Converted string data to numerical types in NumPy arrays.
   - Handled missing values (e.g., NaN in "Type 2" column).

◈ Code Snippets / Design Screenshots

Example 1: Sorting Pokémon by Attack and Defense


jupyter NextGen_week2_task Last Checkpoint: 19 minutes ago
File Edit View Run Kernel Settings Help
Trusted
Code
JupyterLab
Python 3 (ipykernel)

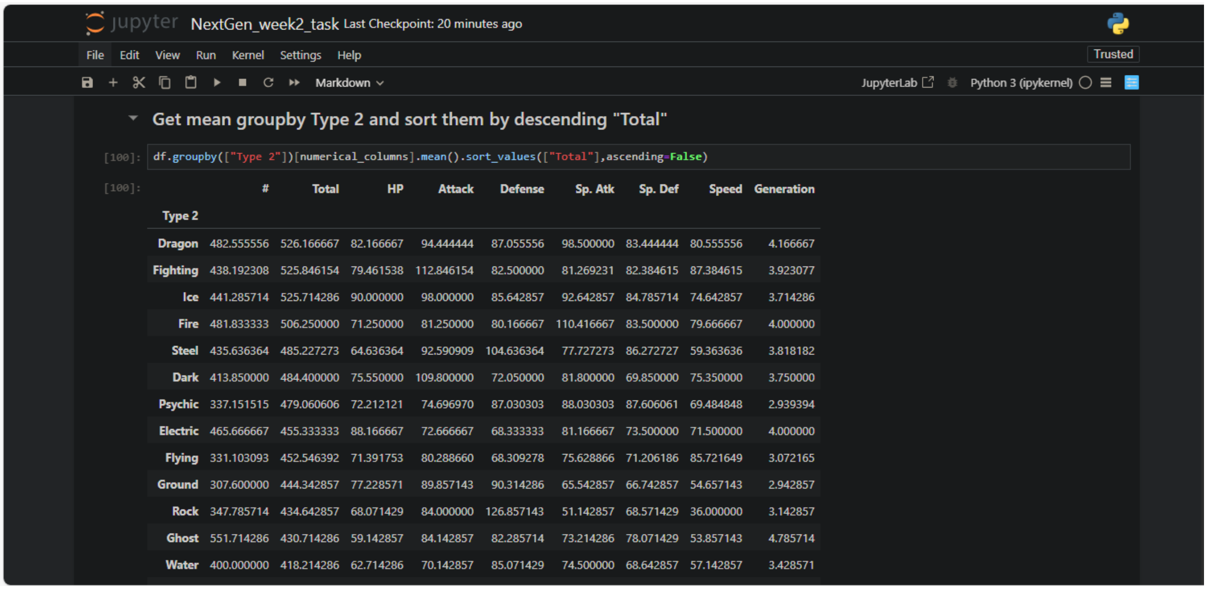
Sort by Attack and Defense in descending order

```
[95]: df.sort_values(["Attack","Defense"],ascending=[0,0]).head(10)
```

[95]:

	#	Name	Type 1	Type 2	Total	HP	Attack	Defense	Sp. Atk	Sp. Def	Speed	Generation	Legendary
163	150	MewtwoMega Mewtwo X	Psychic	Fighting	780	106	190	100	154	100	130	1	True
232	214	HeracrossMega Heracross	Bug	Fighting	600	80	185	115	40	105	75	2	False
424	383	GroudonPrimal Groudon	Ground	Fire	770	100	180	160	150	90	90	3	True
426	384	RayquazaMega Rayquaza	Dragon	Flying	780	105	180	100	180	100	115	3	True
429	386	DeoxysAttack Forme	Psychic	NaN	600	50	180	20	180	20	150	3	True
494	445	GarchompMega Garchomp	Dragon	Ground	700	108	170	115	120	95	92	4	False
711	646	KyuremBlack Kyurem	Dragon	Ice	700	125	170	100	120	90	95	5	True
527	475	GalladeMega Gallade	Psychic	Fighting	618	68	165	95	65	115	110	4	False
387	354	BanetteMega Banette	Ghost	NaN	555	64	165	75	93	83	75	3	False
454	409	Rampardos	Rock	NaN	495	97	165	60	65	50	58	4	False

Example 2: Grouping by Type and Calculating Averages


jupyter NextGen_week2_task Last Checkpoint: 20 minutes ago
File Edit View Run Kernel Settings Help
Trusted
Markdown
JupyterLab
Python 3 (ipykernel)

Get mean groupby Type 2 and sort them by descending "Total"

```
[100]: df.groupby(["Type 2"])[numerical_columns].mean().sort_values(["Total"],ascending=False)
```

[100]:

	#	Total	HP	Attack	Defense	Sp. Atk	Sp. Def	Speed	Generation
Type 2									
Dragon	482.555556	526.166667	82.166667	94.444444	87.055556	98.500000	83.444444	80.555556	4.166667
Fighting	438.192308	525.846154	79.461538	112.846154	82.500000	81.269231	82.384615	87.384615	3.923077
Ice	441.285714	525.714286	90.000000	98.000000	85.642857	92.642857	84.785714	74.642857	3.714286
Fire	481.833333	506.250000	71.250000	81.250000	80.166667	110.416667	83.500000	79.666667	4.000000
Steel	435.636364	485.227273	64.636364	92.590909	104.636364	77.727273	86.272727	59.363636	3.818182
Dark	413.850000	484.400000	75.550000	109.800000	72.050000	81.800000	69.850000	75.350000	3.750000
Psychic	337.151515	479.060606	72.212121	74.696970	87.030303	88.030303	87.606061	69.484848	2.939394
Electric	465.666667	455.333333	88.166667	72.666667	68.333333	81.166667	73.500000	71.500000	4.000000
Flying	331.103093	452.546392	71.391753	80.288660	68.309278	75.628866	71.206186	85.721649	3.072165
Ground	307.600000	444.342857	77.228571	89.857143	90.314286	65.542857	66.742857	54.657143	2.942857
Rock	347.785714	434.642857	68.071429	84.000000	126.857143	51.142857	68.571429	36.000000	3.142857
Ghost	551.714286	430.714286	59.142857	84.142857	82.285714	73.214286	78.071429	53.857143	4.785714
Water	400.000000	418.214286	62.714286	70.142857	85.071429	74.500000	68.642857	57.142857	3.428571

◈ Challenges Faced

1. **Type casting of Ndarray:**

   - Struggled with typecasting of Ndarray as it contains alphabetic and numeric values in object format.

   - **Resolution:** Observed first row carefully and type casted each column using astype(int) method.

2. **Complex Grouping Operations:**
    - Confusion when aggregating multiple columns (e.g., mean vs. count).
    - **Resolution:** Referred to Pandas documentation for groupby syntax.
3. **Performance Issues:**
    - Slow operations on large datasets.
    - **Resolution:** Optimized by selecting specific columns (df[['col1', 'col2']]).

◈ Learning Outcome

- **Pandas Proficiency:** Mastered data filtering (loc), sorting, and grouping.
- **Statistical Analysis:** Used descriptive stats (describe()) and custom aggregations.
- **Real-world Data Handling:** Worked with CSV files and cleaned messy data.
- **Debugging Skills:** Improved error resolution (e.g., TypeError during conversions).

◈ Next Steps

For **Week 3**, the focus will be on:

- **Data Visualization:** Creating plots with Matplotlib/Seaborn.
- **Advanced Pandas:** Merging datasets, handling time-series data.
- **Machine Learning Prep:** Feature engineering and correlation analysis.

◈ Resources:

- Pandas: Pandas Documentation
- NumPy: NumPy Guide
- Dataset: https://www.kaggle.com/datasets/abcsds/pokemon
