Féidearthachtaí as Cuimse
Infinite Possibilities

Programming for Analytics

Lecture 9: NumPy

Bojan Božić
School of Computer Science
TU Dublin, Grangegorman

bojan.bozic@tudublin.ie

OLLSCOIL TEICNEOLAÍOCHTA BHAILE ÁTHA CLIATH
TU DUBLIN
TECHNOLOGICAL UNIVERSITY DUBLIN

Overview

- What is NumPy?
- Arrays
- Indexing and Slicing
- Random Numbers
- Filtering

Dataset Overview

- `malazan_characters_stats.csv`
- Includes Strength, Morale, Loyalty, Magic Power
- We'll analyse characters numerically using NumPy

What is NumPy?

- High-performance numerical library
- Used for mathematical and array operations
- Core dependency of pandas, SciPy, scikit-learn

Why NumPy?

- Much faster than Python lists
- Supports vectorised operations
- Efficient memory use

Importing NumPy

- `import numpy as np`
- `np.__version__`

Creating Arrays

- `np.array([1,2,3])`
- Multi-dimensional arrays
- `np.zeros()` – array of 0s
- `np.ones()` – array of 1s
- `np.arange()` – array of range values
- `np.linspace()` – evenly spaced values between start and end

NumPy + Pandas

- `df['Strength'].to_numpy()`
- `df[['Strength','Magic_Power']].to_numpy()`

Activity: Create Arrays

- Load the dataset into a `DataFrame` and create arrays for the following columns:
  - `strength = np.array(df['Strength'])`
  - `magic = np.array(df['Magic_Power'])`
  - `morale = np.array(df['Morale'])`

Array Attributes

- `shape`: shows instances in dimensions of the array
- `ndim`: shows number of dimensions
- `size`: number of elements in the array
- `dtype`: type of elements in the array
- `astype()`: change type of elements

Indexing Characters

- `strength[0] - # Anomander`
- `morale[:5] -`

`# first 5 characters`

Slicing Example

- Top magic users: `magic[magic>80]`
- High morale: `morale[morale>85]`

Vectorised Operations

- `win_rate = victories / battles`
- `power = (strength + magic + morale) / 3`

Activity: Example Calculation

- Compute Karsa's win rate
- Compute Quick Ben's magic/morale ratio

Aggregations

- `np.mean()`: mean of array
- `np.median()`: median of array
- `np.std()`: standard deviation
- `np.max(), np.min()`: maximum and minimum

Activity: Statistics on Characters

- Average Strength
- Median Intelligence
- Std deviation of Morale

Reshaping Arrays

- `arr.reshape(3,2)` – change shape of array without altering data
- Flattening with `flatten()` – convert multi-dimensional array to one dimension

Stacking Arrays

- `np.vstack([strength, morale])` – stack arrays vertically
- `np.hstack([…])` – stack arrays horizontally

Broadcasting

- `morale + 5`
- `strength * 1.1`

Smaller array broadcast across larger array. Most often calculations with one constant.

Activity: Broadcasting

- Add loyalty bonus +3
- Scale Magic Power

Random Numbers

- `np.random.randint()` – generate random integers from specified range
- `np.random.rand()` – random values in given shape

Activity: Luck Simulation

```
luck =
np.random.randint(1,101,size=14)
```

- Compare luck to morale by creating a ratio.

Correlation

- `np.corrcoef(strength, victories)`
- Find strongest relationships

Activity: Character Analytics

- Who is strongest overall?
- Who has highest power index?

Boolean Filtering

- `df[df['Magic_Power']>80]`
- `df[(df['Strength']>90) & (df['Loyalty']>85)]`

Activity: Top Champions

- Add the power index to your dataframe and find out who the top champions are by sorting the dataframe by power index and printing out the top five.

Bonus Activity

- Simulate a duel: Rake vs Icarium
- Use NumPy for random damage

- Each round, **both characters attack once**
- Damage is generated with `np.random.randint`, ensuring NumPy controls all randomness
- HP drops until one (or both!) reach zero
- Prints round-by-round combat logs

Full Workflow

1. Load dataset
2. Convert to NumPy arrays
3. Compute stats
4. Rank characters

Summary

- NumPy is essential for analytics
- Arrays enable fast computations
- Foundation for ML and pandas

Questions?