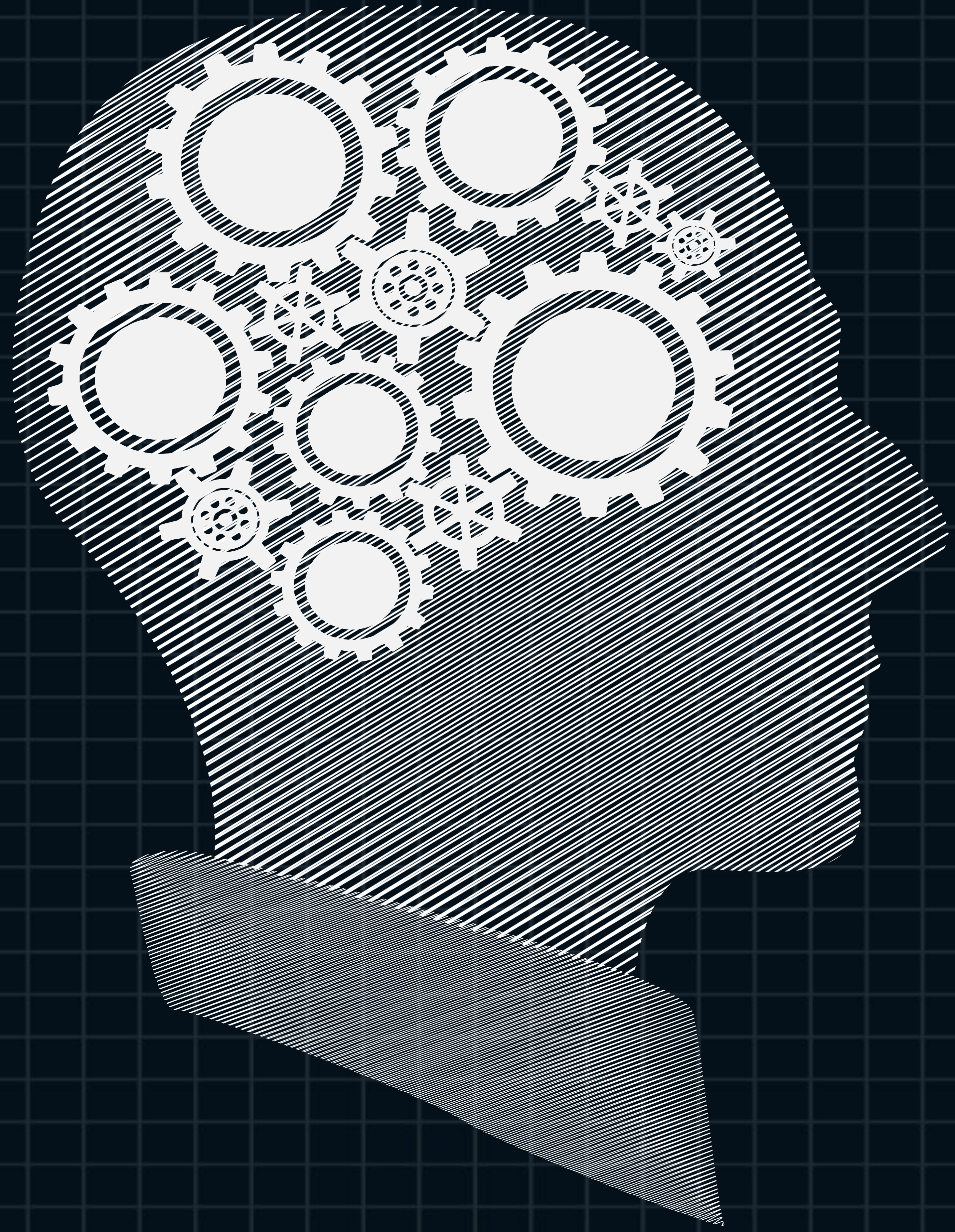


NumPy Master Class

Lecture.10
Rounding and Sorting

Rounding Functions

```
numpy.around(a, decimals=0, out=None)
numpy.round_(a, decimals=0, out=None)
ndarray.round(decimals=0, out=None)
```

```python
import numpy as np

x = np.random.uniform(-5, 5, (5, ))

np_around = np.around(x, decimals=2)
np_round_ = np.round_(x, decimals=2)
x_round = x.round(decimals=2)

print(f"x: \n {x}\n")

print(f"np_around: \n {np_around}")
print(f"np_round_: \n {np_round_}")
print(f"x_round: \n {x_round}")
```

```
x:
 [ 3.77130966 -4.06540544 -0.7369412  -0.26779326  0.80197106]

np_around:
 [ 3.77 -4.07 -0.74 -0.27  0.8 ]
np_round_:
 [ 3.77 -4.07 -0.74 -0.27  0.8 ]
x_round:
 [ 3.77 -4.07 -0.74 -0.27  0.8 ]
```

- Rounding Functions

Rounding Functions

```
import numpy as np

scores = np.random.uniform(0, 100, (100, 5))

means = scores.mean(axis=0)
stds = scores.std(axis=0)


print(f"class means: \n {means}")
print(f"class stds: \n {stds}\n")
```

```
class means:
 [51.14436919 49.20511642 45.83984329 51.13305663 52.77068665]
class stds:
 [27.51168138 31.52110869 31.08176805 26.21210195 27.49764483]
```

```
print(f"class means: \n {means.round(2)}")
print(f"class stds: \n {stds.round(2)}")
```

```
class means:
 [51.14 49.21 45.84 51.13 52.77]
class stds:
 [27.51 31.52 31.08 26.21 27.5 ]
```

- Rounding Functions

Rounding Functions

```python
import numpy as np

x = np.random.uniform(-5, 5, (5, ))


np_around = np.around(x)
np_round_ = np.round_(x)
x_round = x.round()


print(f"x: \n {x}\n")

print(f"np_around: \n {np_around}")
print(f"np_round_: \n {np_round_}")
print(f"x_round: \n {x_round}")
```

```
x:
 [ 3.48438515 -1.9036614   2.63927546 -0.51517886  3.05421167]

np_around:
 [ 3. -2.  3. -1.  3.]
np_round_:
 [ 3. -2.  3. -1.  3.]
x_round:
 [ 3. -2.  3. -1.  3.]
```

- Rounding Functions

Ceiling and Flooring Functions

```
numpy.ceil(x, /, out=None, *, where=True, casting='same_kind', order='K', dtype=None, subok=True[, signature, extobj])
numpy.floor(x, /, out=None, *, where=True, casting='same_kind', order='K', dtype=None, subok=True[, signature, extobj])
```

```python
import numpy as np

x = np.random.uniform(-5, 5, (5, ))

ceil = np.ceil(x)
floor = np.floor(x)

print(f"x: \n {x}\n")

print(f"ceil: \n {ceil}")
print(f"floor: \n {floor}")
```

```
x:
 [-4.02531311  1.21678491  3.63022617 -3.42367724  4.92429523]

ceil:
 [-4.  2.  4. -3.  5.]
floor:
 [-5.  1.  3. -4.  4.]
```

- Rounding Functions

Truncation Functions

`numpy.trunc`*(x, /, out=None, *, where=True, casting='same_kind', order='K', dtype=None, subok=True*[*, signature, extobj*])

```python
import numpy as np

x = np.random.uniform(-5, 5, (5, ))

trunc = np.trunc(x)

print(f"x: \n {x}\n")

print(f"trunc: \n {trunc}")
```

```
x:
 [ 1.49831974  0.44590877 -4.67534704  0.80151716  2.71089048]

trunc:
 [ 1.  0. -4.  0.  2.]
```

Lecture.10
Rounding and Sorting

- Rounding Functions

Truncation Functions

```python
import numpy as np

x = np.random.uniform(-5, 5, (5, ))


trunc_where = np.where(x >= 0, np.floor(x), np.ceil(x))
trunc = np.trunc(x)


print(f"x: \n {x}\n")

print(f"trunc_where: \n {trunc_where}")
print(f"trunc: \n{trunc}")
```

```
x:
 [-1.23773426 -0.08975259  4.8163968  -2.55348586 -1.25676803]

trunc_where:
 [-1. -0.  4. -2. -1.]
trunc:
[-1. -0.  4. -2. -1.]
```

- Rounding Functions

Truncation Functions

```python
import numpy as np

x = np.random.uniform(-5, 5, (5, ))


trunc = 0.1*np.trunc(10*x)


print(f"x: \n {x}\n")
print(f"trunc: \n {trunc}")
```

```
x:
 [ 2.51346428 -1.0792005  -3.0255869   4.70130872 -2.43689857]

trunc:
 [ 2.5 -1.  -3.   4.7 -2.4]
```

- Rounding Functions

Truncation Functions

```python
import numpy as np

x = np.random.uniform(-5, 5, (5, ))

int_part = np.trunc(x)
frac_part = x - int_part

print(f"x: \n {x}\n")

print(f"int_part: \n {int_part}")
print(f"frac_part: \n {frac_part}")
```

```
x:
 [-1.31897598 -1.87246704  3.0183615  -4.29552814  1.8357296 ]

int_part:
 [-1. -1.  3. -4.  1.]
frac_part:
 [-0.31897598 -0.87246704  0.0183615  -0.29552814  0.8357296 ]
```

Lecture.10 Rounding and Sorting

- Sorting Functions

Sorting APIs

```
numpy.sort(a, axis=-1, kind=None, order=None)
numpy.argsort(a, axis=-1, kind=None, order=None)
```

- Sorting Functions

Sorting Vectors

```
import numpy as np

x = np.random.randint(0, 100, (10, ))

sort = np.sort(x)
argsort = np.argsort(x)

print(f"x: \n{x}\n”)

print(f"sort: \n{sort}”)

print(f"argsort: \n{argsort}")
```

```
x:
[35 17 72 37 32 65  4 81 92 59]

sort:
[ 4 17 32 35 37 59 65 72 81 92]

argsort:
[6 1 4 0 3 9 5 2 7 8]
```

Lecture.10
Rounding and Sorting

- Sorting Functions

Sorting Vectors

```
import numpy as np

x = np.random.randint(0, 100, (10, ))

sort = np.sort(x)[::-1]
argsort = np.argsort(x)[::-1]

print(f"x: \n{x}\n")

print(f"sort: \n{sort}”)

print(f"argsort: \n{argsort}")
```

```
x:
[95 81 36 64 83 25 29  7 90  6]

sort:
[95 90 83 81 64 36 29 25  7  6]

argsort:
[0 8 4 1 3 2 6 5 7 9]
```

Lecture.10
Rounding and Sorting

- Sorting Functions

Sorting Vectors

```
import numpy as np

pred = np.random.uniform(0, 100, (5, ))
pred /= pred.sum()

top3_pred = np.sort(pred)[::-1][:3]
top3_indices = np.argsort(pred)[::-1][:3]

print(f"pred: \n{pred.round(3)}\n”)

print(f"top-3 pred: {top3_pred.round(3)}")

print(f"top-3 indices: {top3_indices}")
```

```
pred:
[0.113 0.183 0.228 0.378 0.098]

top-3 pred: [0.378 0.228 0.183]

top-3 indices: [3 2 1]
```

Lecture.10
Rounding and Sorting

- Sorting Functions

Sorting Matrices

```
import numpy as np

x = np.random.randint(0, 100, (4, 5))

sort = np.sort(x, axis=0)
argsort = np.argsort(x, axis=0)

print(f"x: \n{x}\n")

print(f"sort: \n{sort}")

print(f"argsort: \n{argsort}")
```

```
x:
[[99 59 19  4 75]
 [50 37 78 47 76]
 [42 49 28 84 96]
 [ 6 78 56 56 45]]

sort:
[[ 6 37 19  4 45]
 [42 49 28 47 75]
 [50 59 56 56 76]
 [99 78 78 84 96]]

argsort:
[[3 1 0 0 3]
 [2 2 2 1 0]
 [1 0 3 3 1]
 [0 3 1 2 2]]
```

Lecture.10
Rounding and Sorting

- Sorting Functions

Sorting Matrices

```
import numpy as np

x = np.random.randint(0, 100, (4, 5))

sort = np.sort(x, axis=0)[::-1, :]
argsort = np.argsort(x, axis=0)[::-1, :]

print(f"x: \n{x}\n")
print(f"sort: \n{sort}")
print(f"argsort: \n{argsort}")
```

```
x:
[[72 89 99 46 91]
 [98 45 65 54 72]
 [79 24  8 17 39]
 [84  6 46 82 26]]

sort:
[[98 89 99 82 91]
 [84 45 65 54 72]
 [79 24 46 46 39]
 [72  6  8 17 26]]

argsort:
[[1 0 0 3 0]
 [3 1 1 1 1]
 [2 2 3 0 2]
 [0 3 2 2 3]]
```

Lecture.10
Rounding and Sorting

- Sorting Functions

Sorting Matrices

```python
import numpy as np

scores = np.random.randint(0, 100, (5, 3))

sort = np.sort(scores, axis=0)[::-1, :]
argsort = np.argsort(scores, axis=0)[::-1, :]

top2_scores = sort[:2, :]
top2_students = argsort[:2, :]

print(f"scores: \n{scores}\n")
```

```
scores:
[[79  7  1]
 [28 16 29]
 [76 97  1]
 [29 46 94]
 [81  5 21]]
```

```python
print(f"sort: \n{sort}")
print(f"argsort: \n{argsort}\n")
```

```
sort:            argsort:
[[81 97 94]      [[4 2 3]
 [79 46 29]       [0 3 1]
 [76 16 21]       [2 1 4]
 [29  7  1]       [3 0 2]
 [28  5  1]]      [1 4 0]]
```

```python
print(f"top-2 scores: \n{top2_scores}")
print(f"top-2 students: \n{top2_students}")
```

```
top-2 scores:    top-2 students:
[[81 97 94]      [[4 2 3]
 [79 46 29]]      [0 3 1]]
```

- Sorting Functions

Sorting Matrices

```python
import numpy as np

x = np.random.randint(0, 100, (4, 5))

sort_ascending = np.sort(x, axis=1)
argsort_ascending = np.argsort(x, axis=1)

sort_descending = np.sort(x, axis=1)[:, ::-1]
argsort_descending = np.argsort(x, axis=1)[:, ::-1]


print(f"x: \n{x}\n")

print(f"sort(ascending): \n{sort_ascending}")
print(f"argsort(ascending): \n{argsort_ascending}\n")

print(f"sort(descending): \n{sort_descending}")
print(f"argsort(descending): \n{argsort_descending}\n")
```

```
x:
[[79  9 24 49 50]
 [52 83 14 31 70]
 [29 37 45 92 80]
 [58  5 82 49 81]]
```

```
sort(ascending):
[[ 9 24 49 50 79]
 [14 31 52 70 83]
 [29 37 45 80 92]
 [ 5 49 58 81 82]]
argsort(ascending):
[[1 2 3 4 0]
 [2 3 0 4 1]
 [0 1 2 4 3]
 [1 3 0 4 2]]
```

```
sort(descending):
[[79 50 49 24  9]
 [83 70 52 31 14]
 [92 80 45 37 29]
 [82 81 58 49  5]]
argsort(descending):
[[0 4 3 2 1]
 [1 4 0 3 2]
 [3 4 2 1 0]
 [2 4 0 3 1]]
```

Lecture.10
Rounding and Sorting

- Sorting Functions

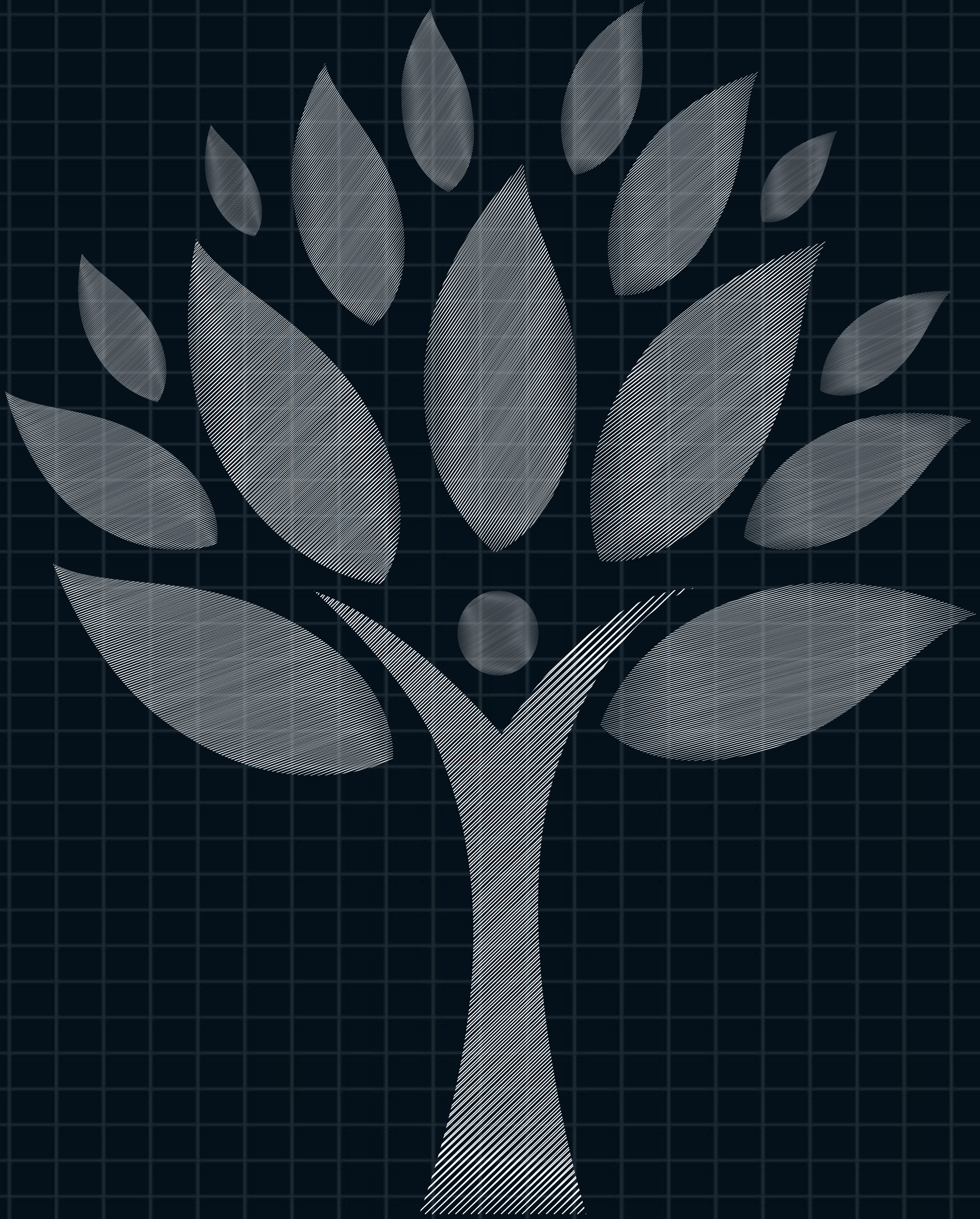
Sorting Matrices

```python
import numpy as np

scores = np.random.randint(0, 100, (5, 3))

sort = np.sort(scores, axis=1)
argsort = np.argsort(scores, axis=1)

bottom2_scores = sort[:, :2]
bottom2_subjects = argsort[:, :2]

print(f"scores: \n{scores}\n")

print(f"sort: \n{sort}")
print(f"argsort: \n{argsort}\n")

print(f"bottom-2 scores: \n{bottom2_scores}")
print(f"bottom-2 subjects: \n{bottom2_subjects}")
```

```
scores:
[[51 47 61]
 [17 66 54]
 [ 9 42 70]
 [21 49 45]
 [ 7 85 73]]
```

```
sort:
[[47 51 61]
 [17 54 66]
 [ 9 42 70]
 [21 45 49]
 [ 7 73 85]]
argsort:
[[1 0 2]
 [0 2 1]
 [0 1 2]
 [0 2 1]
 [0 2 1]]
```

```
bottom-2 scores:
[[47 51]
 [17 54]
 [ 9 42]
 [21 45]
 [ 7 73]]
bottom-2 subjects:
[[1 0]
 [0 2]
 [0 1]
 [0 2]
 [0 2]]
```



NumPy Master Class

Lecture.10
Rounding and Sorting