جلسه اول آزمایشگاه هوش مصنوعی Scribing

جلسه اول آزمایشگاه هوش مصنوعی

.آشنا شدیم NumPy در این جلسه، با مفاهیم پایه ای برنامه نویسی پایتون و کتابخانه

Jupyter Lab آشنایی با

:را با دستور زیر باز کردیم Jupyter Lab ابتدا

```
python -m jupyterlab
```

print آشنایی با دستور

.نحوه چاپ متن در خروجی را یاد گرفتیم ("mmad")printسپس با دستور

NumPy آشنایی با کتابخانه

:در طول برنامه، آن را با دستور زیر فراخوانی کردیم NumPy با توجه به نیاز به استفاده از کتابخانه

```
import numpy as np
```

.استفاده کنیم npهر بار، می توانیم از نام کوتاه شده (NumPy) با این کار، به جای نوشتن نام کامل کتابخانه

ساخت ماتریس

استفاده می کنیم. به عنوان مثال، برای ساخت یک ماتریس یک array، از تابع NumPy برای ساخت ماتریس در کتابخانه :بعدی با سه عنصر، از دستور زیر استفاده می کنیم

```
arr = np.array([12, 324, 6])
```

.نام متغیری است که ماتریس را در خود ذخیره می کند arrدر این مثال،

ساخت ماتریس های چند بعدی

:بسازیم. برای مثال arrayبه همین ترتیب، می توانیم ماتریس های دو بعدی، سه بعدی و ... را نیز با استفاده از تابع

:ماتریس دو بعدی

```
arr = np.array([[12, 34], [56, 78]])
```

:ماتریس سه بعدی

```
arr = np.array([[[12, 34], [56, 78], [323, 67, 8]]])
```

:ماتریس چهار بعدی

```
arr = np.array([[[[12, 34], [56, 78], [323, 67, 8], [2, 34]]]])
```

مشاهده تعداد بعدهای ماتریس

:استفاده کنیم. به عنوان مثال ndimبرای مشاهده تعداد بعدهای ماتریس، می توانیم از خاصیت

```
arr = np.array([[[[12, 34], [56, 78], [323, 67, 8], [2, 34]]]])
arr.ndim
```

.دارای چهار بعد است arrخواهد بود، زیرا ماتریس 4خروجی این کد

تعیین تعداد بعدهای ماتریس

:تعداد بعدهای ماتریس را به طور دلخواه تعیین کنیم. به عنوان مثال arrayدر تابع ndminمی توانیم با استفاده از آرگومان

```
arr = np.array([12, 34], ndmin=2)
```

.با دو بعد، حتی با وجود فقط دو عنصر، ساخته خواهد شد arrبا این کد، ماتریس