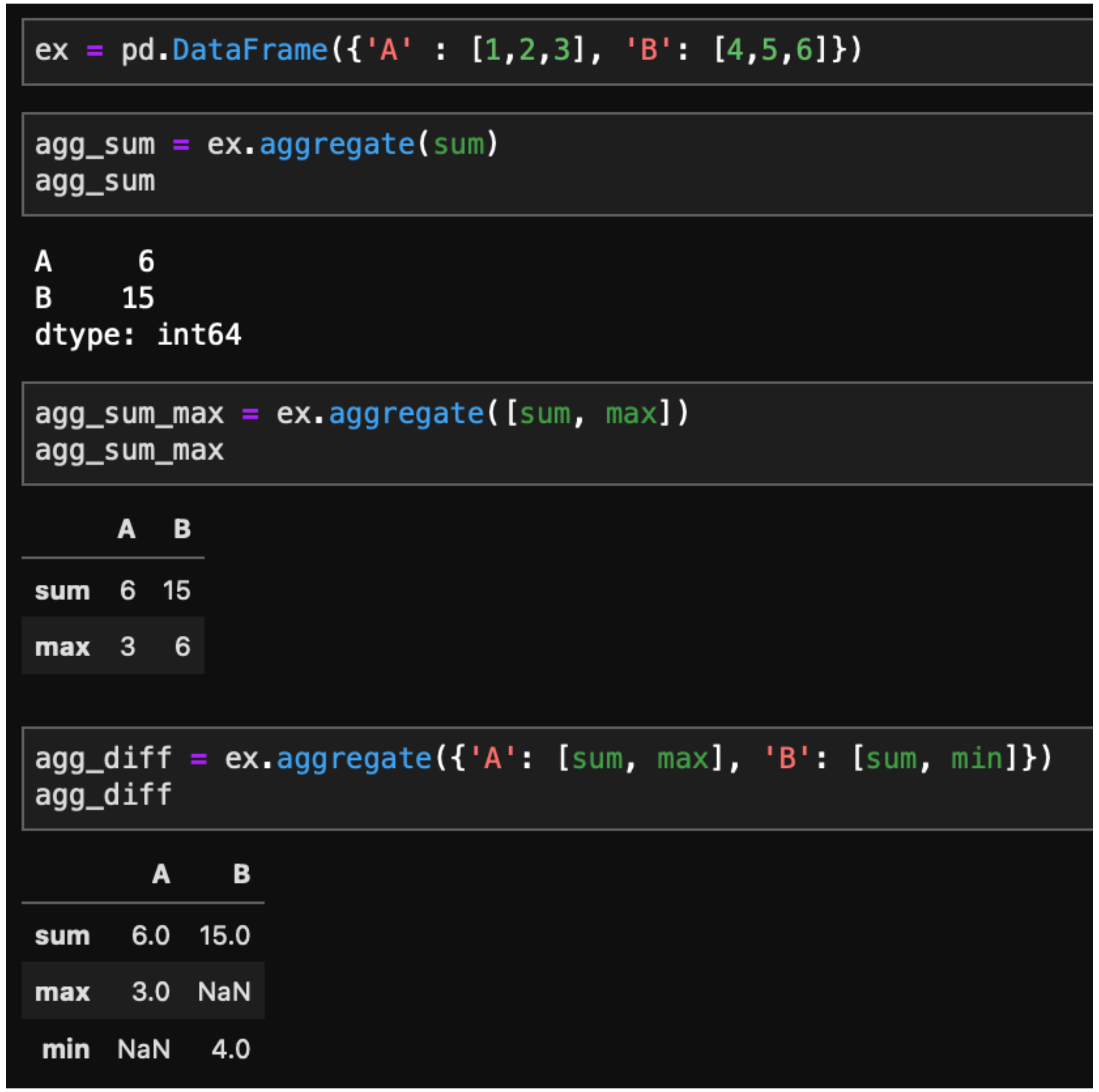
Study 4

GDSC AI_ML[6]

2021.11.20

pd.DataFrame.aggregate()

= pd.DataFrame.agg() (alias)



```
ex = pd.DataFrame({'A' : [1,2,3], 'B': [4,5,6]})
```

```
agg_sum = ex.aggregate(sum)
agg_sum
```

```
A     6
B    15
dtype: int64
```

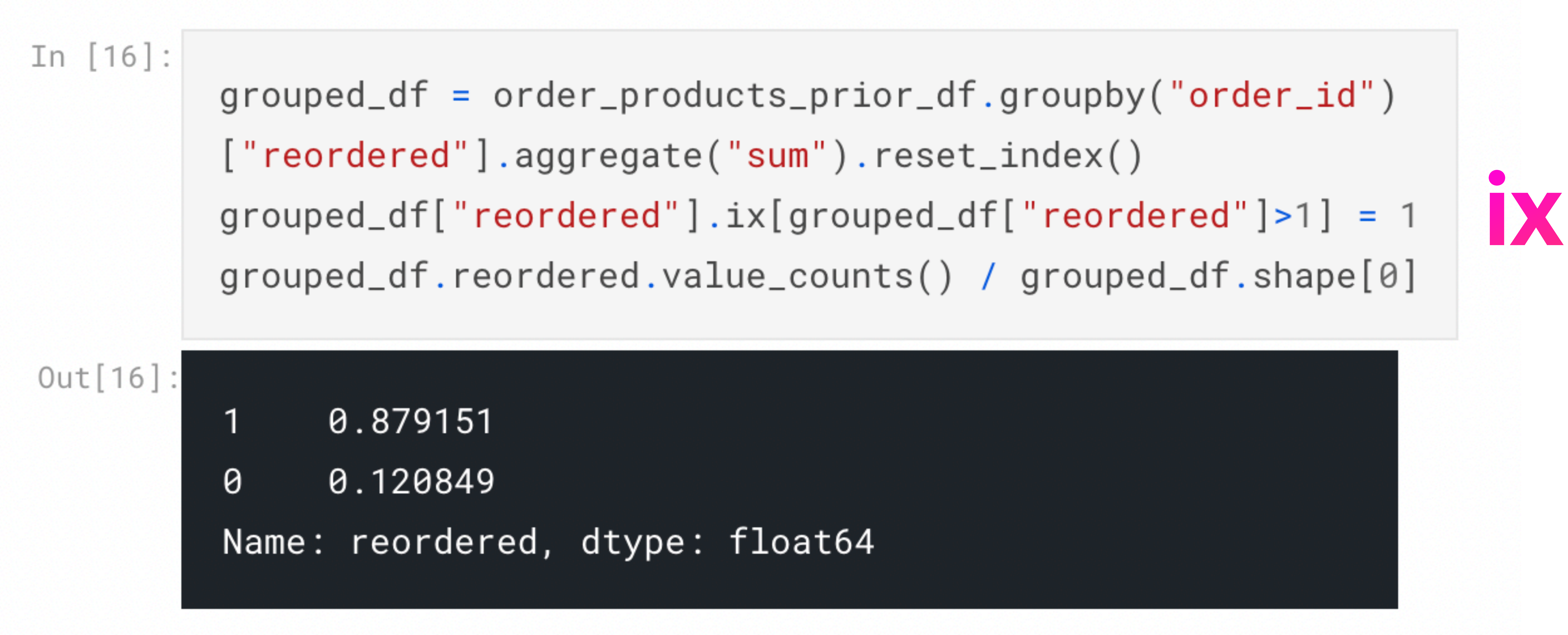
```
agg_sum_max = ex.aggregate([sum, max])
agg_sum_max
```

	A	B
sum	6	15
max	3	6

```
agg_diff = ex.aggregate({'A': [sum, max], 'B': [sum, min]})
agg_diff
```

	A	B
sum	6.0	15.0
max	3.0	NaN
min	NaN	4.0

Example Kernel 5



In [16]:

```
grouped_df = order_products_prior_df.groupby("order_id")
["reordered"].aggregate("sum").reset_index()
grouped_df["reordered"].ix[grouped_df["reordered"]>1] = 1
grouped_df.reordered.value_counts() / grouped_df.shape[0]
```

Out[16]:

```
1    0.879151
0    0.120849
Name: reordered, dtype: float64
```

ix -> iloc

pd.merge()

Useful if columns are not the same

pd.concat()

Useful if columns are the same