Feature Scaling

STANDARDSCALER & MINMAXSCALER

StandardScaler

```python
from sklearn.preprocessing import StandardScaler
standard_df=complete_df.copy()
scaler=StandardScaler()
standard_df[num_cols]=scaler.fit_transform(complete_df[num_cols])
standard_df[num_cols].head()
```

✓ 27.7s Python

	order_id	product_id	add_to_cart_order	reordered	aisle_id	department_id	user_id	order_number	order_dow	order_hour_of_day	days_since_prior_order
0	-1.732751	0.535137	-1.031488	0.834137	0.387029	0.967671	1.670551	-0.806502	1.08188	-1.042062	-0.353583
1	-1.732751	0.241806	-0.891170	0.834137	0.308502	-0.942805	1.670551	-0.806502	1.08188	-1.042062	-0.353583
2	-1.732751	-1.152706	-0.750852	-1.198844	0.858196	0.490052	1.670551	-0.806502	1.08188	-1.042062	-0.353583
3	-1.732751	1.443010	-0.610534	0.834137	-1.366758	0.490052	1.670551	-0.806502	1.08188	-1.042062	-0.353583
4	-1.732751	0.316291	-0.470216	-1.198844	-1.419110	0.490052	1.670551	-0.806502	1.08188	-1.042062	-0.353583

قمنا ب استخدام **StandardScaler** على **num-cols** فقط لتوحيد مقياس القيم وجعلها قابلة للمقارنة بين **features** الأخرى

يعتمد **StandardScaler**على طرح المتوسط من كل قيمة ثم القسمة على الانحراف المعياري مما يجعل متوسط القيم **0** والانحراف المعياري **1**

ماذا نلاحظ في الجدول؟

ان بعض القيم اصبحت سالبة وهذا امر طبيعي ويعني ان هذه القيم اقل من متوسط العمود بينما القيم الموجبة تدل على انها اعلى من المتوسط

هذا الاسلوب يقلل تأثير اختلاف المقاييس بين الخصائص ويساعد النماذج الحساسة للمقياس على التعلم بشكل ادق

MinMaxScaler

```python
from sklearn.preprocessing import MinMaxScaler
min_max_df=complete_df.copy()
MMS=MinMaxScaler()
min_max_df[num_cols]=MMS.fit_transform(complete_df[num_cols])
min_max_df[num_cols].head()
```

24.1s Python

	order_id	product_id	add_to_cart_order	reordered	aisle_id	department_id	user_id	order_number	order_dow	order_hour_of_day	days_since_prior_order
0	0.0	0.666553	0.000000	1.0	0.639098	0.75	0.980942	0.020408	0.833333	0.391304	0.266667
1	0.0	0.583332	0.006944	1.0	0.616541	0.15	0.980942	0.020408	0.833333	0.391304	0.266667
2	0.0	0.187695	0.013889	0.0	0.774436	0.60	0.980942	0.020408	0.833333	0.391304	0.266667
3	0.0	0.924125	0.020833	1.0	0.135338	0.60	0.980942	0.020408	0.833333	0.391304	0.266667
4	0.0	0.604464	0.027778	0.0	0.120301	0.60	0.980942	0.020408	0.833333	0.391304	0.266667

قمنا ب استخدام **MinMaxScaler** على **num-col** فقط لجعل القيم بين **0** و **1**

يعتمد هذا الاسلوب على تحويل القيم بحيث تصبح اصغر قيمة =**0** واكبر قيمة =**1** ,بينما باقي القيم بتطلع ارقام بين **0** و **1** حسب قربها من الاصغر او الاكبر

ماذا نلاحظ في الجدول؟

ان جميع القيم أصبحت ضمن نفس النطاق هاد الاشي بساعد لنموذج على التعامل مع البيانات بسهولة

لكن هذا الاسلوب قد يتأثر بالقيم الشاذه **(outlier)** لانه بيعتمد مباشرة على على اقل واكبر قيمة في العمود

Compare StandardScaler & MinMaxScaler

قمنا بتجربة طريقتي **StandardScaler & MinMaxScaler** لمقارنة تأثير كل منهما على البيانات

لاحضنا ان **MinMaxScaler** أكثر حساسية للقيم الشاذة حيث يمكن لقيمة كبيرة واحدة ان تؤثر على توزيع باقي القيم

بالمقابل **StandardScaler** اقل تأثرا بالقيم الشاذه ويعطي توزيعا اكثر توازنا للخصائص العددية

بناءا على ذلك, اعتمدنا انا و عون تطبيق الداتا سيت يلي عنا على **StandardScaler** لانه اكثر استقرارا ومناسب للداتا سيت يلي منستخدمها.