

Features Engineering & Selection

اختيار وهندسة المُدخلات (Feature Selection and Engineering)

بعد تجهيز البيانات، يتم الانتقال لمرحلة اختيار المُدخلات التي نريد استخراج المعلومات منها أو بناء نماذج التعلم الآلي، في بعض الأحيان لا يكفي اختيار مدخلات (Feature Selection) فقط وإنما نحتاج أيضا إلى هندسة هذه المدخلات (Feature Engineering) لكن مالفرق بين المصطلحين؟

أولاً: اختيار المُدخلات (Feature Selection)

- مرحلة اختيار المدخلات (features) ونقصد بها اختيار مجموعة محددة من الأعمدة التي نود استخراج معلومات منها واستبعاد المدخلات التي لا تعطي معلومات مفيدة لبناء نماذج التعلم الآلي.
- مرحلة نقفص فيها على اختيار المدخلات القيمة واستبعاد القيم الغير مفيدة مثل (noise data).

لماذا نحتاج (Feature Selection)؟

- تساهم هذه المرحلة في رفع كفاءة نماذج التعلم الآلي في مرحلة التدريب (Training Phase).
- رفع دقة نماذج التعلم الآلي عن طريق حذف irrelevant features، القيم المكررة أو highly correlated.

ثانياً: هندسة وتحويل المدخلات (Features Engineering)

- مرحلة يتم فيها نقل البيانات من شكلها الخام إلى شكل يقدم معلومة أفضل ويخدم الهدف المراد تحقيقه.
- مرحلة يتم فيها إنتاج و اكتشاف معلومات و مدخلات جديدة (new features) لم تكن موجودة في مجموعة البيانات الأصلية (original dataset)؛ لزيادة دقة التنبؤ (predictive power) في خوارزميات التعلم الآلي (machine learning model).

لماذا نحتاج هندسة المدخلات؟

- تساهم هذه المرحلة في رفع دقة النتائج التي نحصل عليها.
- تساعد بمعرفة أفضل تمثيل للبيانات لاستنتاج حل للمشكلة التي نعمل عليها.
- تحويل المدخلات إلى صورة نستطيع من خلالها فهم النتائج المستهدفة.

أمثلة على هندسة المدخلات:

- جدول المبيعات لمنتجين A و B في شركة ما.

Sale Amount	Product	Day	Client	Sales Representative
200	A	7 Dec 2022	Lama	Sara
400	A	7 Dec 2022	Omer	Nourah

200	A	12 Dec 2022	Sama	Nourah
150	B	7 Dec 2022	Asma	Khaled
300	B	29 Dec 2022	Fahad	Sara
150	B	7 Dec 2022	Salem	Ahmed
600	A	12 Dec 2022	Fatemah	Nourah
800	A	1 Jan 2023	Ali	Khaled

من الجدول السابق يمكننا استخلاص معلومات أو مدخلات جديدة (new feature) وهي (Product Sales Per Day) عن طريق تجميع البيانات كما في الشكل التالي.

Product Sales Per Day	Day	Product
600	7 Dec 2022	A
300	7 Dec 2022	B
800	12 Dec 2022	A
0	12 Dec 2022	B
0	29 Dec 2022	A
300	29 Dec 2022	B
800	1 Jan 2023	A
0	1 Jan 2023	B

• من الأمثلة أيضا، استخراج مدخلات إضافية من عمود التاريخ (date) مثل:

- Day of the week
- Day of the month
- Month
- Year
- Weekend or weekday
- Public holiday or working day

من أشكال هندسة المدخلات

• النوع الأول: إعادة ضبط المقياس **Scaling**، ويوجد عدة أمثلة عليه منها:

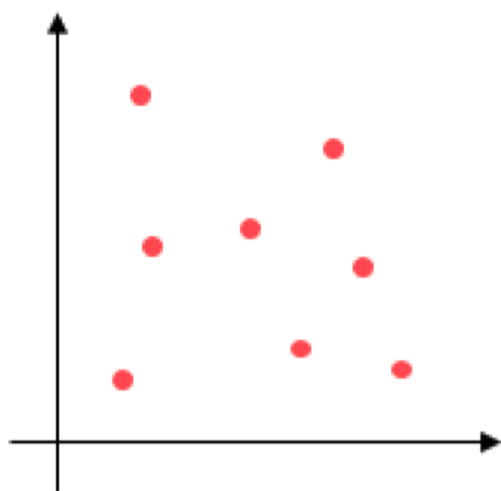
- استخدام Min-Max scaling.
- استخدام Standardization.
- استخدام (Capping (Floor& Ceiling.
- استخدام Quantile Transformers.

• النوع الثاني: التجميع والتقطيع **Merging & Discretization**، ويوجد عدة أمثلة عليه منها:

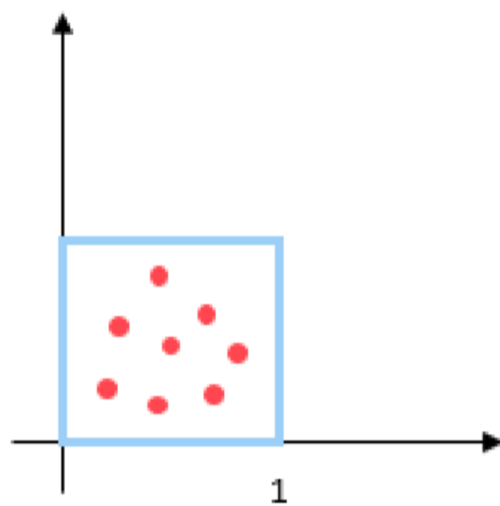
- استخدام Binning.
- استخدام Dimensionality Reduction.
- استخدام One-hot encoding.
- استخدام Aggregation Functions.

استخدام **Min-Max scaling** ويسمى **Normalization**.

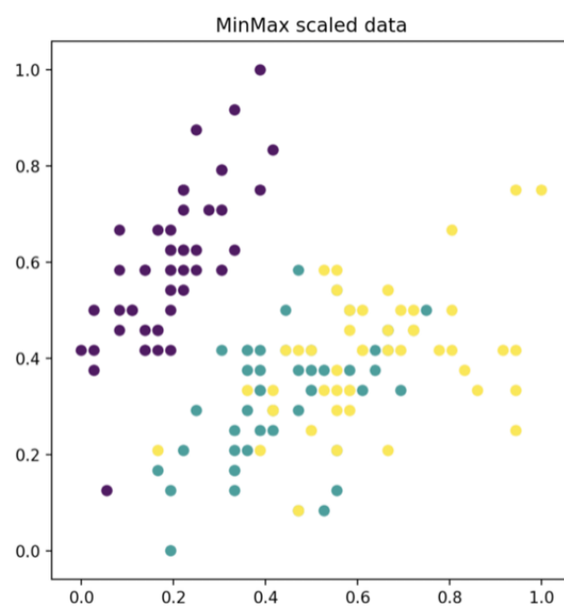
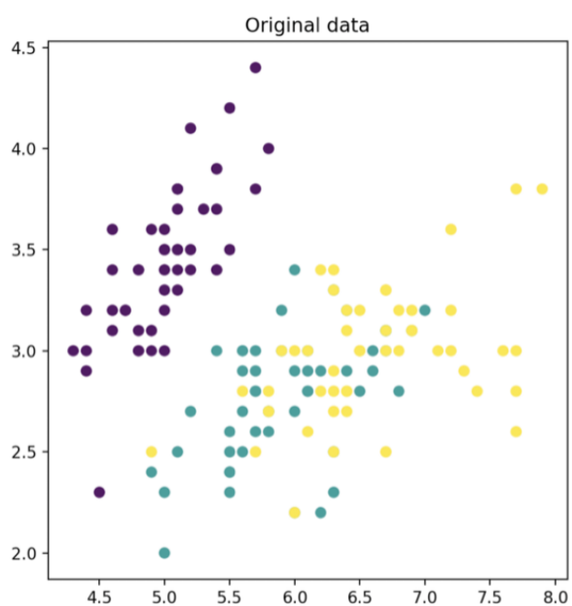
المقصود به تحويل القيم الرقمية وإعادة ضبطها لتكون القيم محدودة بين 0 و 1.



Actual Data



After normalizing



الصيغة الرياضية كالتالي:

$$x_{scaled} = \frac{x - x_{min}}{x_{max} - x_{min}}$$

python - Can someone explain to me how

MinMaxScaler() works? - Stack Overflow

ما فائدة Min-Max scaling؟

- تقادي التحيز عن طريق توحيد القياس للخواص العددية لأن العديد من خوارزميات التعلم الآلي تميل لإيجاد trends في البيانات عن طريق المقارنة بين قيم البيانات وهذا يؤثر على عملية تعلم الآلة عند وجود اختلاف في مقاييس البيانات.
- من أفضل الأمثلة التي توضح أهمية تطبيق Min-Max scaling عندما يكون لدينا عمودين يعبران عن المسافة أحدهما يحوي المسافة بوحدات kms والآخر بوحدات Miles، بالرغم من أنهما يعبران عن المسافة إلا أن كل واحد منهما له مقياس scale مختلف عن الآخر.

تطبيق Min-Max scaling رياضياً:

	Age	Salary
0	22	12000
1	23	11000
2	24	18000
3	25	17000
4	26	15000
5	27	16000
6	28	20000
7	29	25000
8	30	30000

عند تطبيق الصيغة السابقة على الصف الأول (العمر 22 والراتب 12,000) نحصل على التالي:

$$0 = (22-22)/(22-30) = \text{العمر}$$

$$0.05 = (12,000-30,000)/(30,000-11,000) = \text{الراتب}$$

يوضح الجدول التالي القيم بعد تطبيق Min-Max scaling:

	Age	Salary
0	0.000	0.052632
1	0.125	0.000000
2	0.250	0.368421
3	0.375	0.315789
4	0.500	0.210526
5	0.625	0.263158
6	0.750	0.473684
7	0.875	0.736842
8	1.000	1.000000

تطبيق Min-Max scaling عملياً:

توفر مكتبة Scikit-Learn بلغة python خاصية تسمى `MinMaxScaler` و تحتوي `feature_range` hyperparameter يمكن من خلالها تعديل القيم 0-1 لأي قيم أخرى نريدها.

استخدام Standardization ويسمى z-score normalization

- يعمل Standardization بطريقة مختلفة عن Min-Max scaling حيث أنه لا يُقيد القيم بنطاق معين.
- يحول البيانات بحيث يكون المتوسط لها (mean) يساوي الصفر و الانحراف المعياري (standard deviation) يساوي 1 بحيث يعمل بشكل مشابه للتوزيع الطبيعي (Standard Normal Distribution).
- يقوم بطرح المتوسط (mean) ثم القسمة على الانحراف المعياري (standard deviation).

الصيغة الرياضية كالتالي:

$$x_{scaled} = \frac{x - mean}{sd}$$

z score standardization -
Hands-On Machine
Learning on Google Cloud
Platform [Book]

ما فائدة استخدام **Standardization**؟

- يتميز بأنه يتأثر بشكل أقل عند وجود outliers على عكس Min-max scaling.

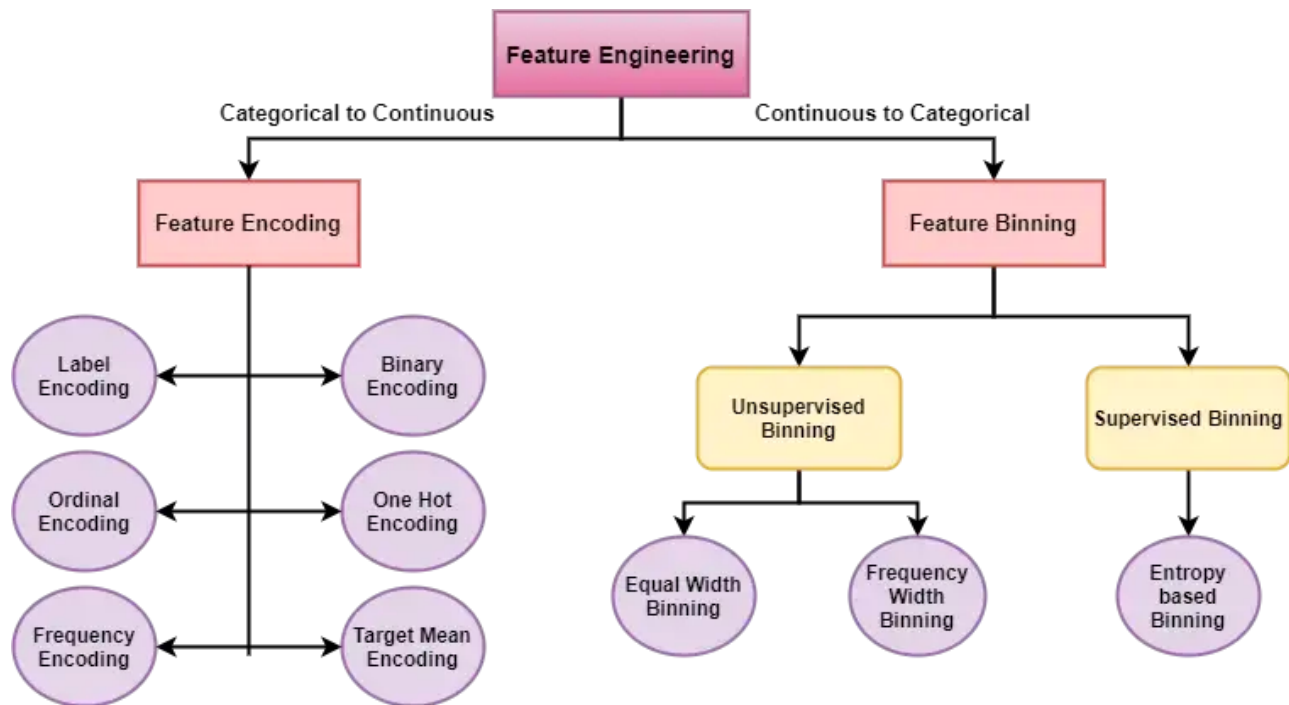
تطبيق **Standardization** عملياً:

توفر مكتبة Scikit-Learn بلغة python خاصية تسمى `StandardScaler` يمكن من خلالها تطبيق **Standardization**.

هل نستخدم **Min-max scaling** أو **Standardization**؟

- كلا النوعين يتم استخدامهم عند استخدام خوارزميات التعلم الآلي التي تعتمد على حساب الانحدار (gradient) مثل:
 - خوارزميات الانحدار (Linear and Logistic Regression).
 - خوارزميات الشبكات العصبية (Artificial Neural Networks)
- النماذج الشجرية (tree-based) مثل: tree و النماذج التي تعتمد على المسافة (distance-based) مثل: SVM و k-nearest لا تتطلب عمل scaling أو Standardization.
- لا يمكن القول أن أحد هذه الطرق هو الصحيح دائماً، على سبيل المثال في خوارزميات الشبكات العصبية:
 - يُفضل استخدام Min-max scaling عندما لا نفترض وجود توزيع معين للبيانات لأن عدم تقييد القيم بنطاق معين يسبب مشاكل لأن القيم المدخلة لها بالغالب تتراوح بين 0 و 1.
 - يُفضل استخدام Standardization عندما يكون توزيع البيانات (Normal or Gaussian distribution).
 - يُفضل استخدام Standardization عند وجود الكثير من outliers لأن Standardization يتأثر بشكل أقل عند وجود outliers على عكس Min-max scaling.

استخدام **Binning**.



تعريف Feature Encoding

- هو عملية تحويل الخواص المُصنفة (categorical features) إلى خواص عددية (numerical features).

تعريف Feature Binning

- هو عملية تحويل الخواص العددية (continuous numeric features) إلى تصنيفات (discrete categories).

أولاً: استخدام Binning أو Discretization

- يتم تقسيم البيانات لمجموعات لها نفس الحجم.
- يتم استخدامه عندما يكون توزيع البيانات مائل (skewed).
- يستخدم لتحديد missing values و outliers.

أنواع Binning:

- تجزئة غير موجهة Unsupervised Binning.

تقطيع الأرقام إلى فئات متساوية من دون الرجوع إلى target class label. هذا النوع يمكن تقسيمه لنوعين:

○ النوع الأول Equal Width Binning

تقطيع الأرقام إلى فئات متساوية بحيث يكون bins أو range لها نفس width.

○ النوع الثاني Equal Frequency Binning

تقطيع الأرقام إلى فئات متساوية بحيث يكون bins أو range لها نفس Frequency.

• تجزئة موجهة Supervised Binning

تقطيع مبني على Entropy.

التجزئة باستخدام Equal Width Binning

$$w = \left\lfloor \frac{\max - \min}{x} \right\rfloor$$

حيث يُمثل:

- الرمز max, min أعلى وأقل قيمة في مجموعة البيانات.
- الرمز x عدد المجموعات.
- الرمز w = width of a category

مثال:

```
age_lst = [10, 15, 16, 18, 20, 30, 35, 42, 48, 50, 52, 55]
x = 4, min = 10, max = 55
w = 11
```

عند تطبيق المعادلة السابقة سوف نحصل على المجموعات التالية:

Numbers in group	Group [start-end]
10, 15, 16, 18, 20	[10-21]
30	[22-33]
35, 42	[34-45]
48, 50, 52, 55	[46-55]

التجزئة باستخدام Equal Frequency Binning

$$freq = \frac{n}{x}$$

حيث يُمثل:

- الرمز n عدد البيانات.
 - الرمز x عدد المجموعات.
 - الرمز freq = frequency of a category
- عند تطبيق المعادلة على المثال السابق سوف نحصل على المجموعات التالية: (3 = 12/4)

Numbers in group	Group [start-end]
10, 15, 16	[10-16]
18, 20, 30	[17-30]
35, 42, 48	[31-48]
50, 52, 55	[49-55]

ثانياً: استخدام Encoding

يعتبر تطبيق Encoding مهم، لأن أغلب خوارزميات التعلم الآلي لا تستطيع معالجة categorical variables

استخدام Label Encoding.

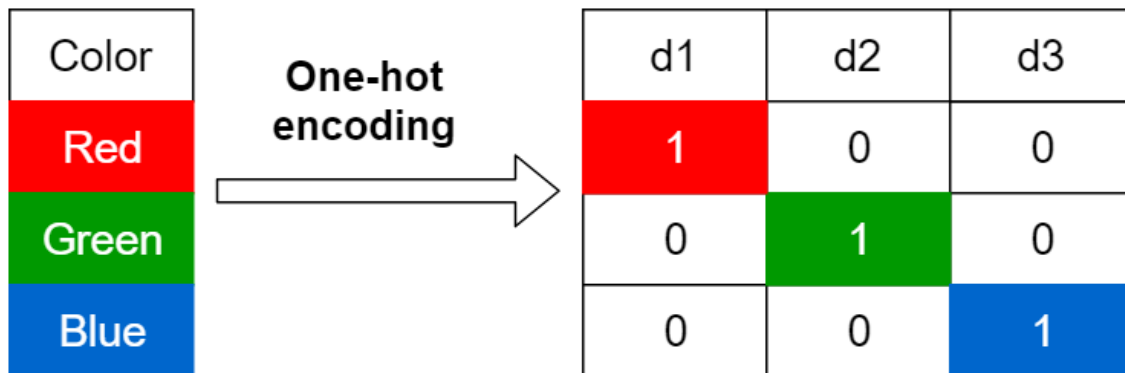
Original Data		Label Encoded Data	
Team	Points	Team	Points
A	25	0	25
A	12	0	12
B	15	1	15
B	14	1	14
B	19	1	19
B	23	1	23
C	25	2	25
C	29	2	29

How to Perform Label Encoding in Python (With Example) - Statology

استخدام Ordinal encoding.

Original Encoding	Ordinal Encoding
Poor	1
Good	2
Very Good	3
Excellent	4

استخدام One-hot encoding.



Encoding Categorical Variables: One-hot vs Dummy Encoding | by Rukshan Pramoditha | Towards Data Science

استخدام Frequency encoding

Column	Freq_Encoding
red	5
green	3
red	5
green	3
blue	4
red	5
red	5
blue	4
red	5
blue	4
blue	4
green	3

استخدام Binary encoding

Column	Label Enc	Binary enc1	Binary enc2	Binary enc3
red	1	0	0	1
green	2	0	1	0
red	1	0	0	1
green	2	0	1	0
blue	3	0	1	1
red	1	0	0	1
grey	4	1	0	0
blue	3	0	1	1
red	1	0	0	1
blue	3	0	1	1
blue	3	0	1	1
green	2	0	1	0
grey	4	1	0	0

- Chapter 2: <https://learning.oreilly.com/library/view/exam-ref-ai-900/9780137358076/>
- http://rasbt.github.io/mlxtend/user_guide/preprocessing/minmax_scaling/
- Chapter 2: <https://learning.oreilly.com/library/view/hands-on-machine-learning/9781491962282/>
- Chapter2, <https://learning.oreilly.com/library/view/hands-on-machine-learning/9781098125967/>