

INT303 W9

NUMERICAL FEATURES

对数字型 feature 的处理。

Binarization

Transform discrete or continuous numeric features in binary features

Example: Number of user views of the same document

document_id	uuid	views_count
25792	6d82e412aa0f0d	8
25792	571016386ffee7	6
25792	6a91157d820e37	6
25792	ad45fc764587b0	6
25792	a743b03f2b8ddc	3



document_id	uuid	viewed
25792	6d82e412aa0f0d	1
25792	571016386ffee7	1
25792	6a91157d820e37	1
25792	ad45fc764587b0	1
25792	8d87becfb35857	1
25792	abcdefg1234567	0

```
>>> from sklearn import preprocessing
>>> X = [[ 1., -1., 2.],
...      [ 2., 0., 0.],
...      [ 0., 1., -1.]]

>>> binarizer =
preprocessing.Binarizer(threshold=1.0)
>>> binarizer.transform(X)
array([[ 1., 0., 1.],
       [ 1., 0., 0.],
       [ 0., 1., 0.]])
```

Binarization with scikit-learn

注：上面讲观看人数二值化 (binarization) 成了 0 (没人看) 和 1 (有人看)。

Binning

Binning is to group data according to specific rules

- Achieve discretization of data (实现数据的离散化)
- enhance data stability (增强数据稳定性)
- reduce the risk of overfitting

Binning is very necessary in logistic regression

Tree models do not need to be binned

这里把连续的数据变成不连续的数据，可以增加鲁棒性。

Age	After Binning
2	Under 18
10	Under 18
99	Order than 60
49	30-60
23	18-30
1	Under 18

From the theoretical stand-point, there are several possible methods of discretization (binning) a continuous variable as follows

- Equal width discretization (等宽离散化)
- Equal Frequency discretization (等频离散化)
- Discretization using decision trees

Equal width discretization

The bins or interval limits are determined so that each interval is of the same width.

- Dividing that range into the amount of bins desired (假如年龄是0-100，等距分就是：假如每20年用一个 bin 表示，就有 5 个 bin，0-20，..., 80-100)
- Note: if the distribution is skewed, this technique does not improve the spread of the values (这个技术不会改变数据分布，即：如果 0-20 岁的人特别多，画出来的柱状图上，代表 0-20 的 bin 就特别高)

Equal Frequency discretization

The boundaries of the intervals are determined so that each bin contains the same number of observations (假如有 100 个人年龄在 0-100，等频分就是：假如分成 5 组，那每组都要 20 个人。例如第一组的20个人，可能在 0-37 岁)

This is a better solution

- We want to spread the values evenly across all bins.

The usual approach is:

- The percentiles
- Quartiles to determine the intervals

Discretization using decision trees

Sorting the observations into the tree end leaves, after training a decision tree.

- Different leaves will contain different number of observations
- It does not preserve frequency like equal frequency discretization (它不会像等频离散化那样保留频率)

For some datasets, discretization with decision trees can improve model performance

- Creating monotonic relationships (创建单调关系) (already capture some of the predictive power of the variable)

Discretization using decision trees 包括使用决策树来确定条柱 (bin) 或连续间隔 (interval) 的最佳拆分点，决策树是一个监督学习。

python implementation

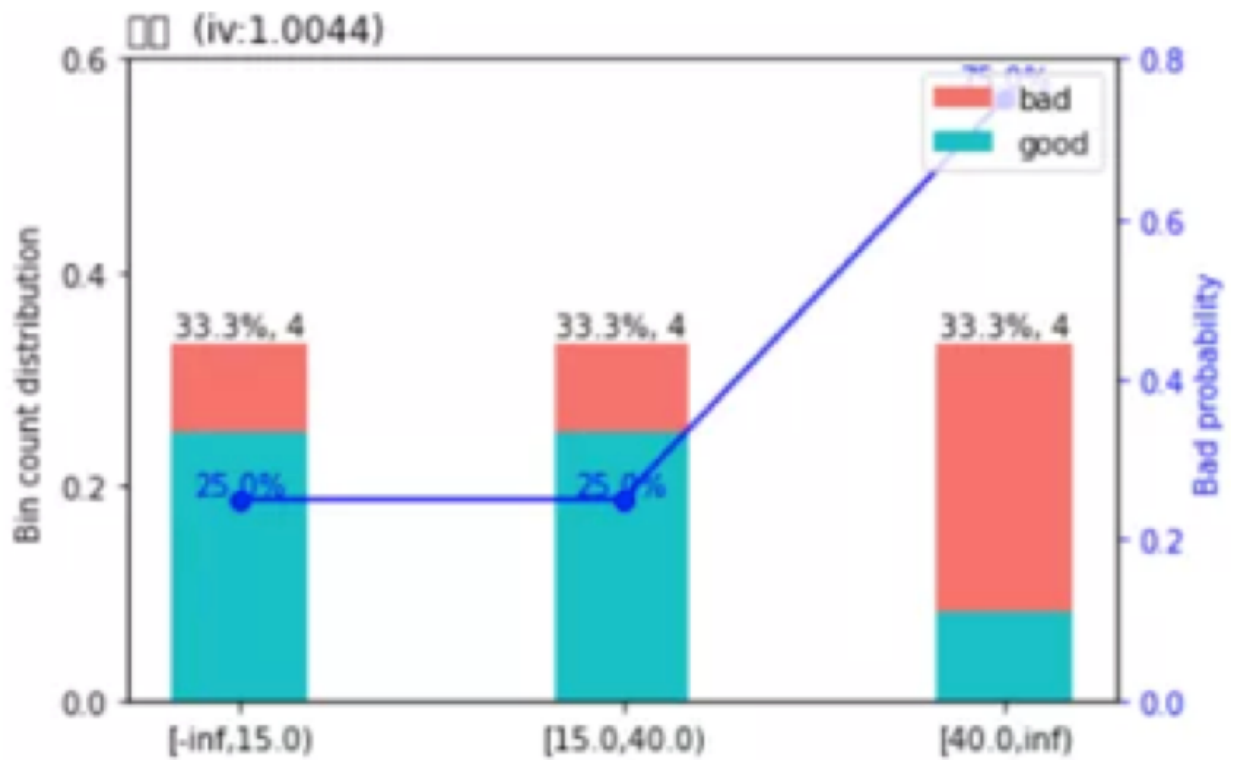
- Equal width/frequency discretization: `pd.cut()/pd.qcut()`

```
1 import pandas as pd
2
3 # 导入一列数据 (Age)
4 df = pd.DataFrame({'年龄': [29, 7, 49, 12, 50, 34, 36, 75, 61, 20, 3, 11]})
5
6 (Equal width discretization)
7 df['等距分箱'] = pd.cut(df['年龄'], 4) # 实现等距分箱, 分为4个箱
8 df['等频分箱'] = pd.qcut(df['年龄'], 4) # 实现等频分箱, 分为4个箱
9 (Equal frequency discretization)
10 df
```

- Discretization with decision trees

```
1 import pandas as pd
2 import scorecardpy as sc
3
4 # 导入两列数据 (Age)
5 df = pd.DataFrame({'年龄': [29, 7, 49, 12, 50, 34, 36, 75, 61, 20, 3, 11],
6                    'Y' : [0, 0, 1, 1, 0, 1, 0, 1, 1, 0, 0, 0]})
7
8 bins = sc.woebin(df, y='Y', method='tree') # 决策树分箱
9 sc.woebin_plot(bins)
```

注：上面的 Y 相等于 label



注：上图只是对 df 和 Y 的统计，其中 0 代表 good，1 代表 bad

Log Transformation

Compresses the range of large numbers and expand the range of small numbers (压缩大数字的范围并扩展小数字的范围). Eg. The larger x is, the slower $\log(x)$ increments.

user_id	views_count		$\log(1+\text{views_count})$
a	1000	➡	6.91
b	500		6.22
c	300		5.71
d	200		5.30
e	150		5.02
f	100		4.62
g	70		4.26
h	50		3.93
i	30		3.43
j	20		3.04
k	10		2.40
l	5		1.79
m	1		0.69

Scaling

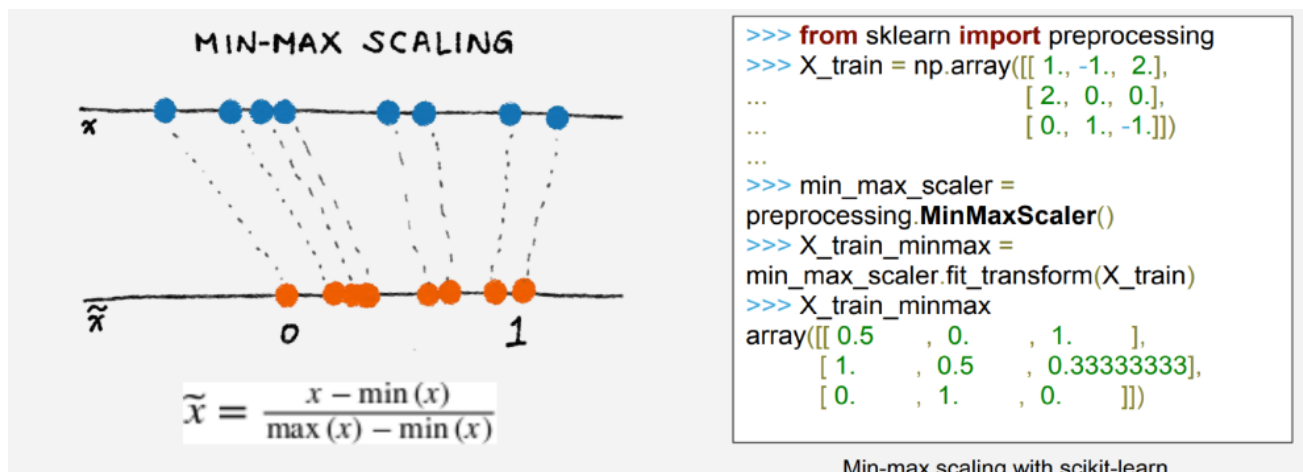
Models that are smooth functions of input features are sensitive to the scale of the input (eg. Linear Regression) (模型对输入的比例敏感). Scale numerical variables into a certain range, dividing values by a normalization constant (no changes in single-feature distribution) (将数值变量缩放到某个范围, 将值除以规范化常量, 即正则化), 让不同的 feature 拥有相同的 scaling.

Popular techniques:

- Min-Max Scaling
- Standard (Z) Scaling

Min-Max Scaling

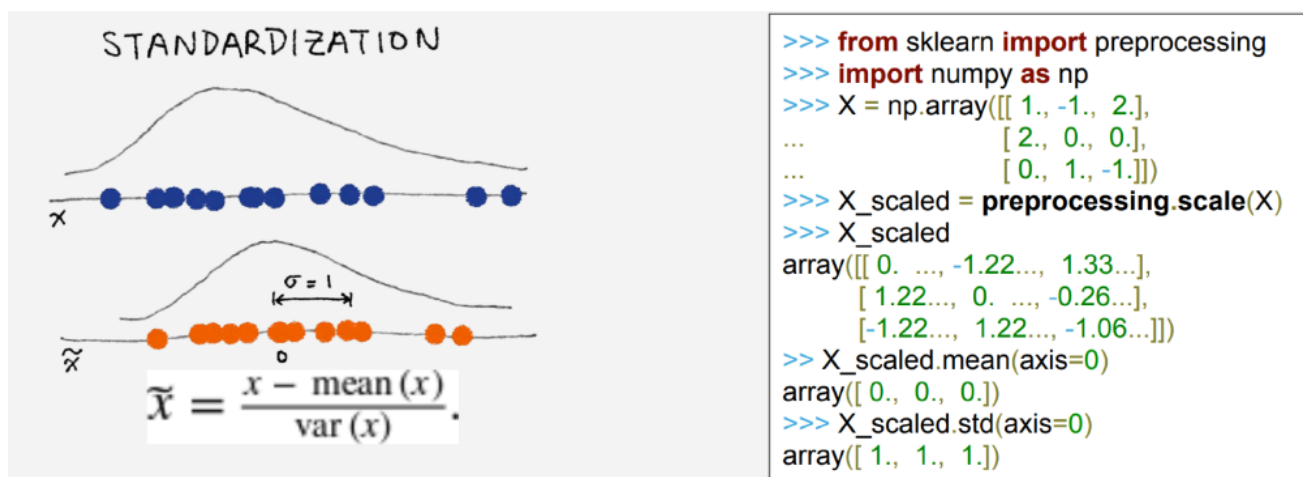
Squeezes (or stretches) all values within the range of [0, 1] to add robustness to very small standard deviations and preserving zeros for sparse data.



注: sparse data, 稀疏数据, 指具有大部分未使用元素 (不携带任何信息的元素, 即 0) 的数据。

Standard (Z) Scaling

After Standardization, a feature has mean of 0 and variance of 1 (assumption of many learning algorithms)



注: $\text{var}(x)$ 是 x 的方差

Interaction Features

Simple linear models use a linear combination of the individual input features, x_1, x_2, \dots, x_n to predict the outcome y .

$$y = w_1x_1 + w_2x_2 + \dots + w_nx_n$$

An easy way to increase the complexity of the linear model is to create feature combinations (nonlinear features).

Example: Degree 2 interaction features for vector $x = (x_1, x_2)$ (degree 2 代表最高 2 次方)

$$y = w_1x_1 + w_2x_2 + w_3x_1x_2 + w_4x_1^2 + w_5x_2^2$$

$(X_1, X_2) \Rightarrow (1, X_1, X_2, X_1^2, X_1X_2, X_2^2)$


```
>>> import numpy as np
>>> from sklearn.preprocessing import PolynomialFeatures
>>> X = np.arange(6).reshape(3, 2)
>>> X
array([[0, 1],
       [2, 3],
       [4, 5]])
>>> poly = PolynomialFeatures(degree=2, interaction_only=False,
include_bias=True)
>>> poly.fit_transform(X)
array([[ 1.,  0.,  1.,  0.,  0.,  1.],
       [ 1.,  2.,  3.,  4.,  6.,  9.],
       [ 1.,  4.,  5., 16., 20., 25.]])
```

CATEGORICAL FEATURES

High cardinality can create very sparse data (cardinality, 基数, 就是集合中元素的个数)

One-hot Encoding (OHE)

Transform a categorical feature with m possible values into m binary features.

platform		platform=desktop	platform=mobile	platform=tablet
desktop		1	0	0
mobile		0	1	0
tablet		0	0	1

Sparse format 对内存友好 (memory-friendly), example: "platform=tablet" can be sparsely encoded as "2:1"

Feature Hashing

Hashes categorical values into vectors with fixed-length (把 categorical value 变成长度固定的向量).

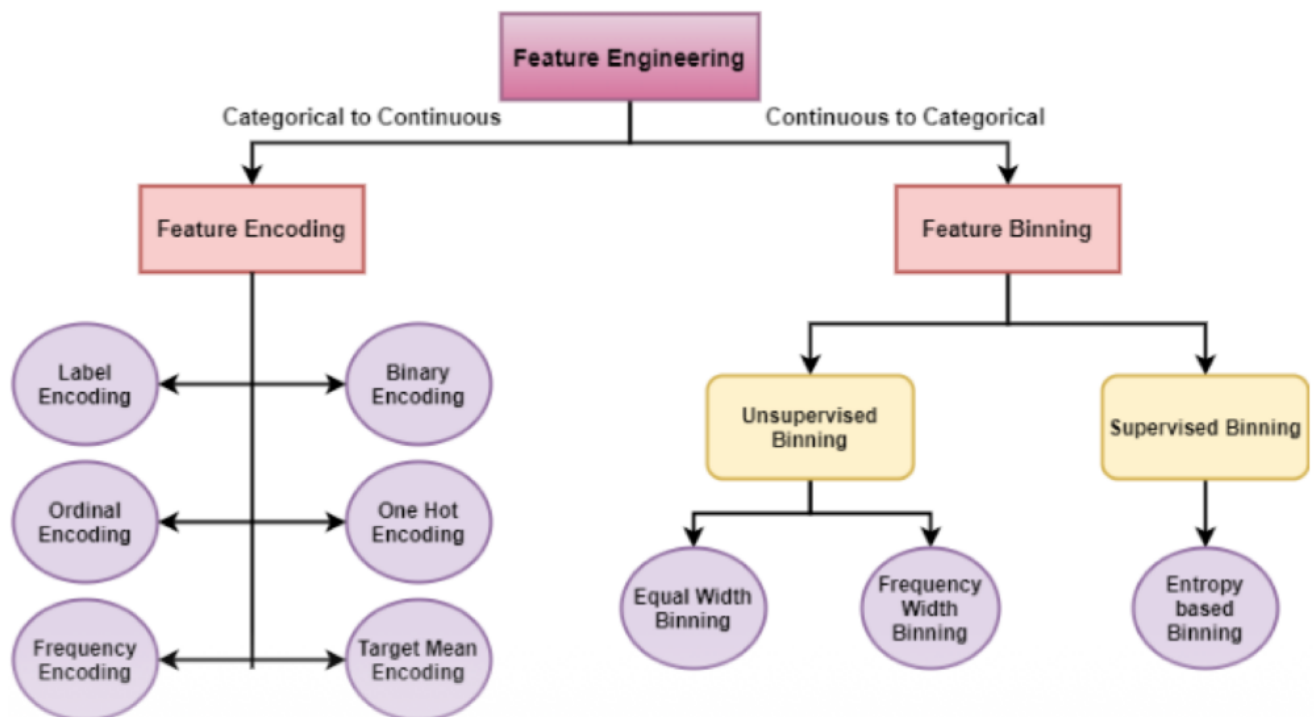
Lower sparsity and higher compression compared to OHE (与 OHE 相比，稀疏度更低，压缩率更高).

Deals with new and rare categorical values (eg: new user-agents)

May introduce collisions (可能出现碰撞)

100 hashed columns					
country	country_hashed_1	country_hashed_2	country_hashed_3	country_hashed_4	...
brazil	1	0	0	0	...
chile	0	0	0	1	...
venezuela	0	0	1	0	...
colombia	0	0	1	0	...
... 222 countries

Above Summary

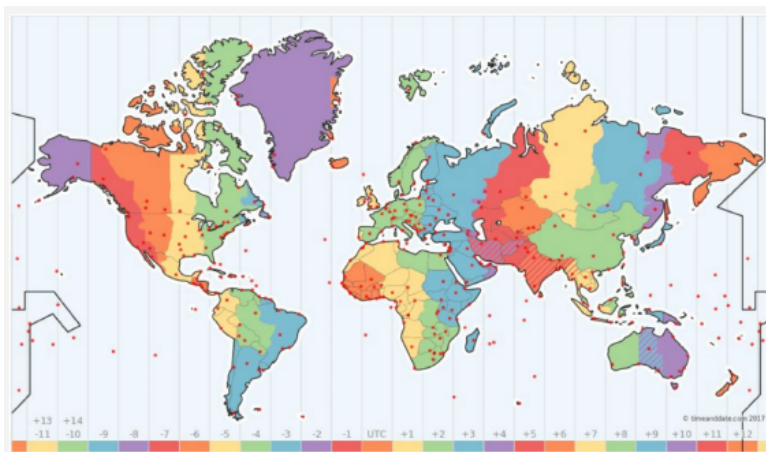


TEMPORAL FEATURES

Time zone conversion

Factors to consider:

- 一个国家可能横跨很多时区
- Daylight Saving Time (DST, 夏令时)
 - Start and end DST dates



	country_name	utc_time_offset	dst_time_offset
0	Afghanistan	+04:30	-
1	Aaland Islands	+02:00	+03:00
2	Albania	+01:00	+02:00
3	Algeria	+01:00	-
4	Samoa (American)	-11:00	-
5	Andorra	+01:00	+02:00
6	Angola	+01:00	-
7	Anguilla (UK)	-04:00	-
8	Antigua & Barbuda	-04:00	-
9	Argentina	-03:00	-

Time binning & Trendlines

Time binning

- Apply binning on time data to make it categorial and more general

Trendlines

- 按时间进行 encoding, 比如: 上个月的支出, 上周的支出.....
- 显示变化的趋势

SPATIAL FEATURES

Spatial variables

Spatial variables encode a location in space, like:

- GPS-coordinates (lat. / long.) – sometimes require projection to a different coordinate system
- Street Addresses – require geocoding
- ZipCodes, Cities, States, Countries – usually enriched with the centroid coordinate of the polygon (from external GIS data)

Derived features (派生特征)

- Distance between a user location and searched hotels (Expedia competition)
- Impossible travel speed (fraud detection)

TEXTUAL FEATURE

Natural language processing

Cleaning

- Lowercasing
- Convert accented characters
- Removing non-alphanumeric
- Repairing

Tokenizing

- Encode punctuation marks
- Tokenize
- N-Grams
- Skip-grams
- Char-grams
- Affixes

Removing

- Stopwords
- Rare words
- Common words

Roots

- Spelling correction
- Chop
- Stem
- Lemmatize

Enrich

- Entity Insertion / Extraction
- Parse Trees
- Reading Level

Text Vectorization

Represent each document as a feature vector in the vector space, where each position represents a word (token) and the contained value is its presence in the document (将每个文档表示为矢量空间中的特征向量，其中每个位置表示一个单词，包含的值是其在文档中的存在)。

- BoW (Bag of words)
- TF-IDF (Term Frequency - Inverse Document Frequency)
- Embeddings (eg. Word2Vec, Glove)
- Topic models (e.g. LDA)

	linux	modern	the	system	steering	petrol
D1	3	4	3	0	2	0
D2	4	3	4	1	0	1
D3	1	0	4	1	0	1
D4	0	1	3	3	3	4

FEATURE SELECTION

Reduces model complexity and training time

- Filtering - Eg. Correlation our Mutual Information between each feature and the response variable (在每个特征和响应变量之间关联共有的信息)
- Wrapper methods - Expensive, trying to optimize the best subset of features (eg. Stepwise Regression)
- Embedded methods - Feature selection as part of model training process (eg. Feature Importances of Decision Trees or Trees Ensembles)

Random forest tree

Each tree of the random forest can calculate the importance of a feature according to its ability to increase the pureness of the leaves.

It's a topic related to how Classification And Regression Trees (CART) work.

```
rf = RandomForestRegressor(random_state=0)

rf.fit(X_train,y_train)
```

```
f_i = list(zip(features,rf.feature_importances_))
f_i.sort(key = lambda x : x[1])
plt.barh([x[0] for x in f_i],[x[1] for x in f_i])

plt.show()
```

