Feature Engineering

Tram Nghi Pham, Hong ONG Xuan

December 8, 2018

Feature Engineering Overview

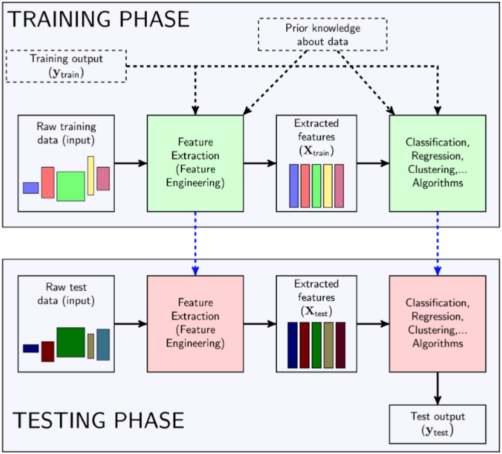

Figure: A standard machine learning pipeline (source: Machine Learning Co Ban, Vu Huu Tien)

Data Overview

Problem Statement: Based on daily grab usage data, can we predict income of a customer, namely over or under 10 million VND per month?

	user_id	birthday	from_district	to_district	service	promo	data_time	gender	status	label
0	A	10-25-1995	1	1	car	1	2018-09-21 00:00:15	female	plantium	1
1	A	10-25-1995	1	1	car	0	2018-10-09 17:55:52	female	plantitum	1
2	B	20-04-1989	2	3	bike	0	2018-10-28 21:54:15	male	silver	1
3	C	11-18-1990	11	11	car	0	2018-10-28 01:52:27	female	gold	0
4	B	20-04-1989	2	1	car	1	2018-10-28 02:10:50	male	silver	1
5	D	15-03-1980	8	10	bike	1	2018-10-11 20:25:26	female	silver	0
6	E	27-08-1970	go_vap	go_vap	bike	0	2018-11-08 17:51:18	male	silver	0
7	F	11-12-2000	phu_nhuan	1	bike	1	2018-11-08 10:48:43	male	silver	0
8	C	11-18-1990	2	12	bike	1	2018-11-02 14:55:33	female	gold	0
9	A	10-25-1995	1	thu_duc	delivery	1	2018-11-02 22:29:04	female	plantium	1
10	D	15-03-1980	8	2	food	0	2018-09-01 10:39:23	female	silver	0
11	A	10-25-1995	1	1	food	0	2018-09-01 08:29:46	female	plantium	1
12	B	20-04-1989	2	1	car	1	2018-09-01 15:05:59	male	silver	1

What is feature engineering?

- "Coming up with features is difficult, time-consuming, requires expert knowledge. 'Applied machine learning' is basically feature engineering." — Prof. Andrew Ng.
- "Feature engineering is the process of transforming raw data into features that better represent the underlying problem to the predictive models, resulting in improved model accuracy on unseen data." — Dr. Jason Brownlee

Feature Engineering: Demo

- We CANNOT throw raw data directly to the model as input
- Our goal is to find features that are highly relevant to income

Feature Engineering: Demo

- We CANNOT throw raw data directly to the model as input
- Our goal is to find features that are highly relevant to income
    1. Age of user
    2. Time using grab
    3. Most visited dictrict in working hour
    4. Total payment
    5. ...
    6. ...

Feature Engineering

- Features can come from two major types based on the raw dataset
  1. Inherent **raw features** are features that can be obtained directly from raw data without any transformation or engineering
  2. **Derived features** are features that can be obtained from feature engineering, i.e through manipulating and transforming raw data.
     For example, feature *age* is derived from *birthday* column of raw data

Different types of variable in statistics

Numerical (quantitative)

- **Discrete:** integer values, typically counts. E.g. age, day sick per year.
- **Continuous:** takes any value in a range of values. E.g. weight, height.

Categorical (qualititive)

- **Nominal:** categories mutually exclusive and unordered. E.g. sex(male/female), blood group (A/B/AB/O).
- **Ordinal:** categories mutually exclusive and ordered. E.g. disease stage (mild/moderate/severe).

Feature Engineering on Numeric Data

- Let try out some examples of feature engineering on our numeric data

	user_id	num_usage	age
0	A	4	23
1	B	3	29
2	C	2	28
3	D	2	38
4	E	1	48
5	F	1	18

Figure: Two new features: num_usage and age derived from numeric columns

Feature Engineering on Numeric Data

Some types of feature you might figure out

Indicator features

- thresholds: You can create an indicator variable for $age \geq 21$.
- special events: Tet, Black Friday, Christmas.

Statistics features

- the count or number of values
- mean
- standard deviation
- minimum, maximum
- 25%, 50%, and 75% quantile

Numerical to Categorical Variable

We can transform numerical variable to categorical variable by using the following techniques: bins numerical, ranging, percentile, threshold, etc.

Example time slot mappings. Define time slot:

- $t \in [0, 23]$: time of day.
- night: $6 \leq t$.
- working: $9 \leq t \leq 12$ and $13 \leq t \leq 17$.
- evening: $t \geq 19$.

Time slot mappings

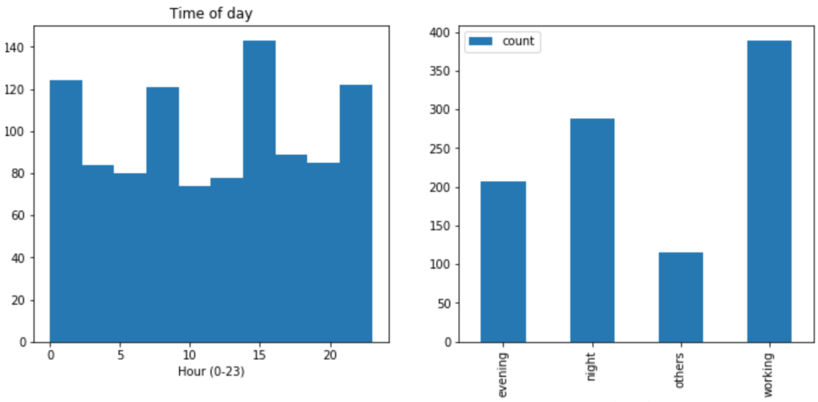

Figure: Time slot mappings from time of day

Standardizing Numerical Variables

Z-score normalization are features that are re-scaled to have a mean of zero and a standard deviation of one. By doing this, we allow models such as Gradient Descent to learn optimally and not skew towards larger scaled features (e.g. age vs income).

$$z = \frac{x - \mu}{\sigma}$$

Where:

- z: z-score.
- x: previous feature value.
- μ: mean of feature value.
- σ: standard deviation of feature value.

Z-score standardization example

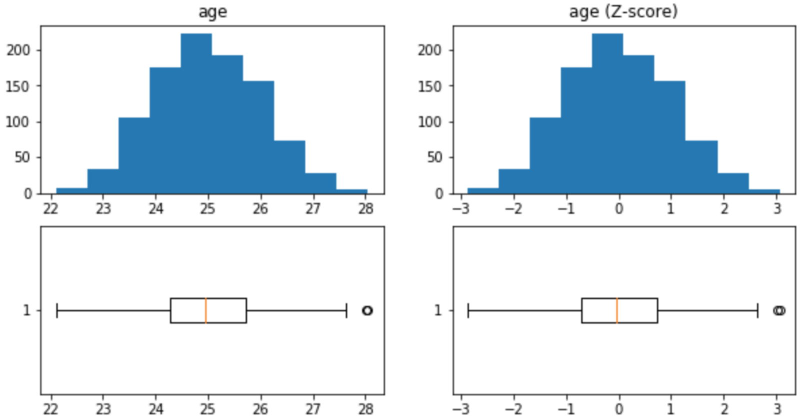

Figure: Z-score transformation on age feature

Standardizing Numerical Variables

The idea is to get every input feature into approximately a $[-1, 1]$ range. The name comes from the use of min and max functions, namely the smallest and greatest values in your dataset. It requires dividing the input values by the range (i.e. the maximum value minus the minimum value) of the input variable:

$$x_i' = \frac{x_i - min(x_i)}{max(x_i) - min(x_i)}$$

Where:

- x_i: is the original i-th input value.
- x_i': normalize feature.

Min-max scaling

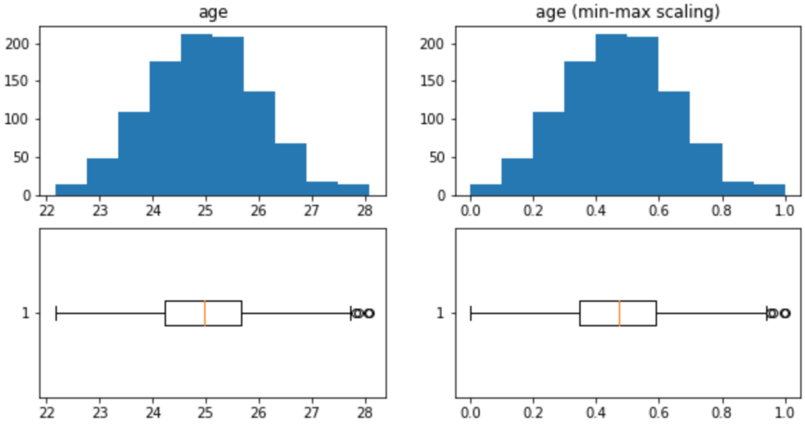

Figure: Min-Max scaling on age feature

Feature Engineering on Categorical Data

- What is the difference between nominal and ordinal features and how we can generate new features from them?
- A nominal variable is one that has two or more categories, but there is no intrinsic ordering to the categories. For example, gender (female and male) has no ordering meaning

Feature Engineering on Categorical Data

- What is the difference between nominal and ordinal features and how we can generate new features from them?
- A nominal variable is one that has two or more categories, but there is no intrinsic ordering to the categories. For example, gender (female and male) has no ordering meaning
- An ordinal variable is similar to a categorical variable. The difference between the two is that there is a clear ordering of the variables. For example, status (platinum, gold, silver) has meaning order
- Why does it matter whether a variable is categorical, or ordinal?
- How do we present information of categorical variables?

Feature Engineering on Categorical and Ordinal Data

- Let consider categorical columns in our dataset

	service	gender	status
0	car	female	plantium
1	car	female	plantitum
2	bike	male	silver
3	car	female	gold
4	car	male	silver
5	bike	female	silver
6	bike	male	silver
7	bike	male	silver
8	bike	female	gold
9	delivery	female	plantium
10	food	female	silver
11	food	female	plantium
12	car	male	silver

Figure: Categorical columns from grab dataset

Feature Engineering on Categorical and Ordinal Data

- Using **one-hot-encoding** for categorical features
- "One hot encoding is a process by which categorical variables are converted into a form that could be provided to ML algorithms to do a better job in prediction" -onlineSource
- Some intuitive features: number of car/bike/food/delivery usage, status
- Basically, we will count how many times user used car, bike, food, or delivery service and represent those information in a meaningful way

Feature engineering on different domain

- Computer Vision: pixels, contours, textures, etc.
- Natural Language Processing: words, grammatical classes and relations, word2vec, doc2vec, etc.
- Signal processing: samples, spectrograms, etc.
- Time series: ticks, trends, reversals, etc.
- Biological: dna, marker sequences, genes, etc.

References

[1] Machine Learning co ban, Bai 11: Gioi thieu ve Feature Engineering.
[2] Princeton COS 424, Feature engineering.
[3] Auckland, Lecture 7: Data Preprocessing and Feature Engineering.