Use Case Tutorial

Gary Chen
2018/10/23

Agenda

- Overall Workflow
- Exploratory Data Analysis
- Feature Engineering + Training

Agenda

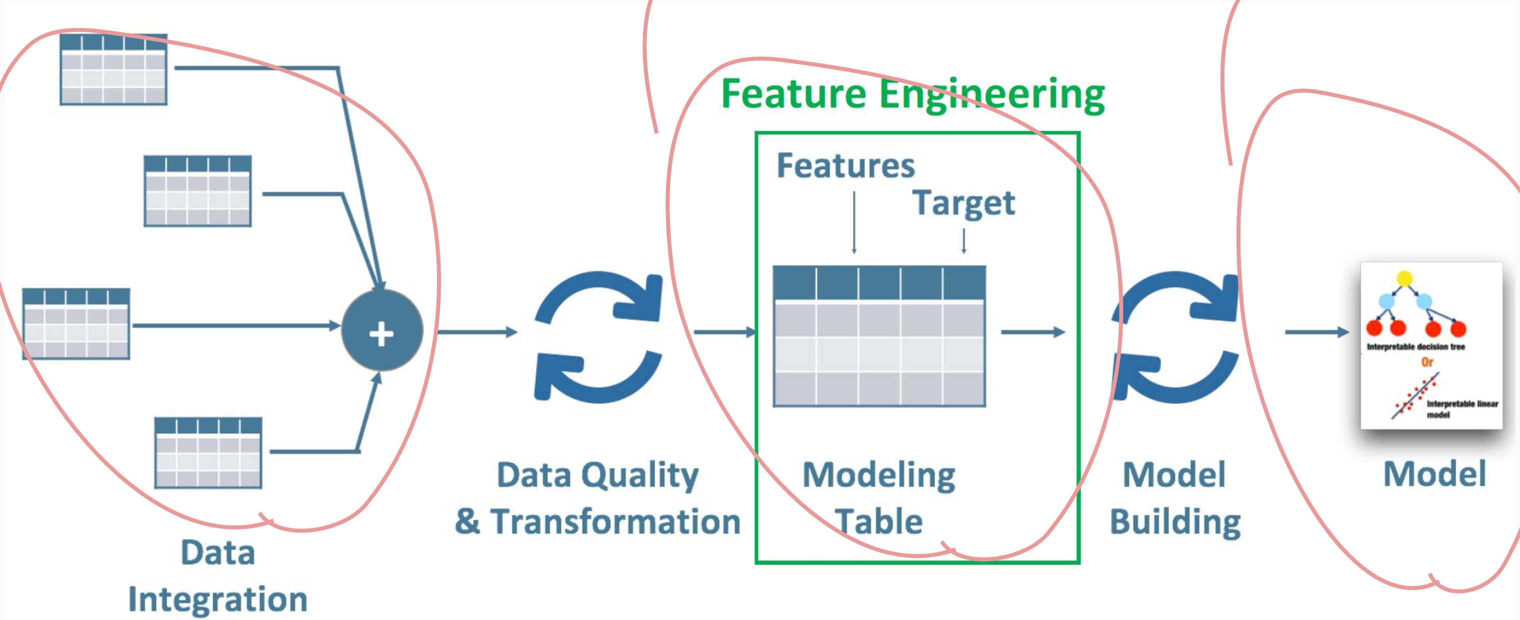
- Overall Workflow
- Exploratory Data Analysis
- Feature Engineering + Training

Typical Enterprice Machine Learning Workflow

CSV(GCS), Cloud SQL, Big Query

Python Script, Data Flow...

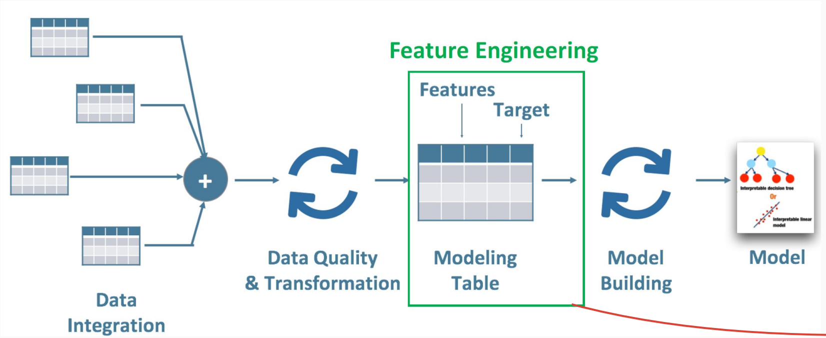
Deploy to ML-Engine,
Put Back to GCS...

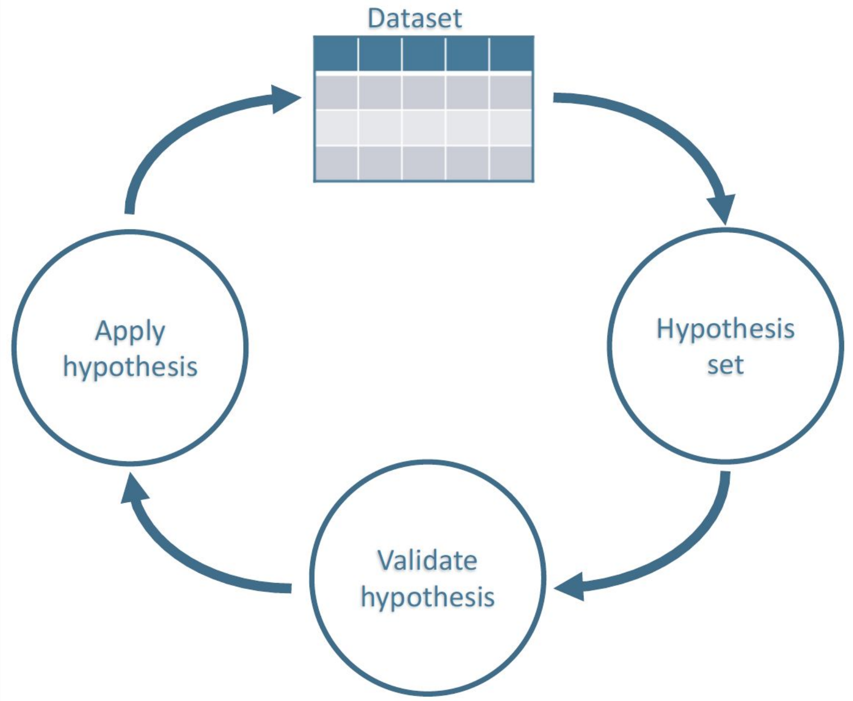


CloudMile

Typical Enterprice Machine Learning Workflow

Feature Engineering cycle

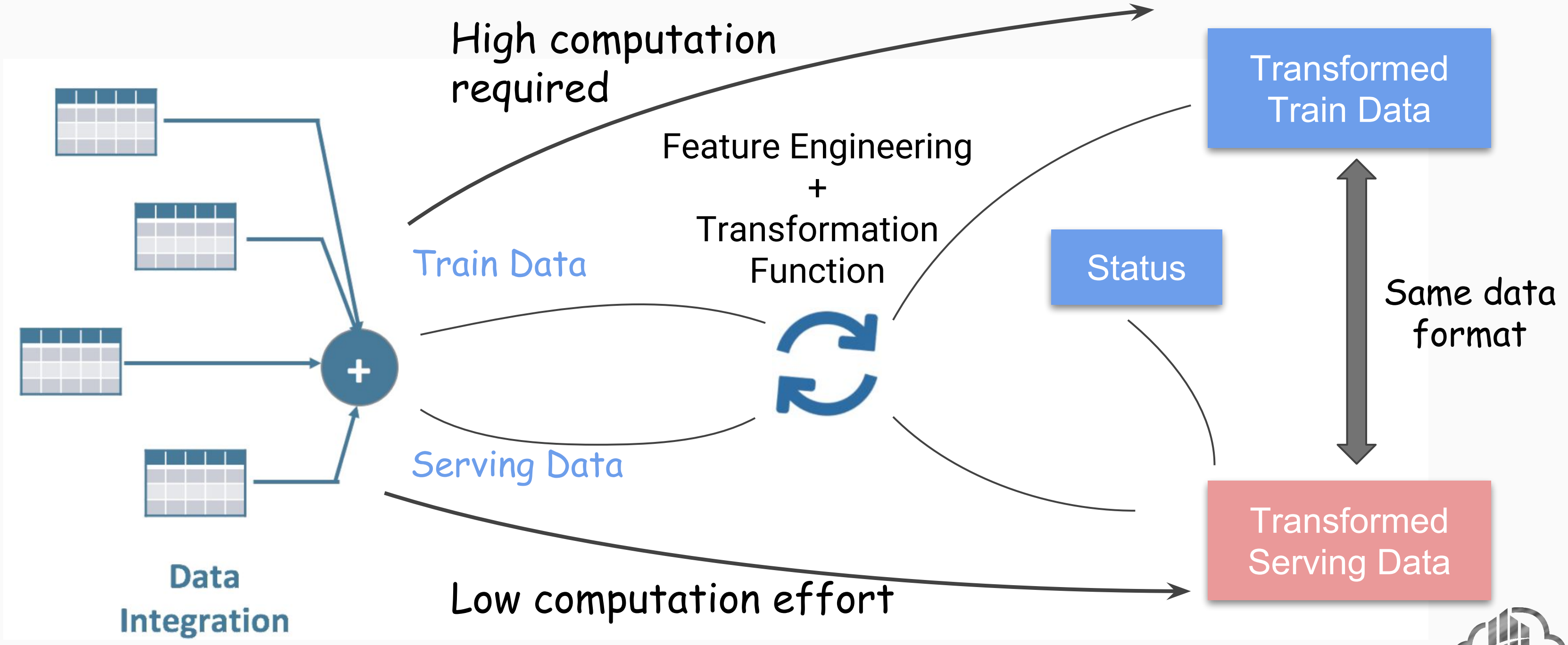




CloudMile

Typical Enterprice Machine Learning Workflow

The Training and Serving



Agenda

Overall Workflow

Exploratory Data Analysis

Feature Engineering + Training

Type of Variable

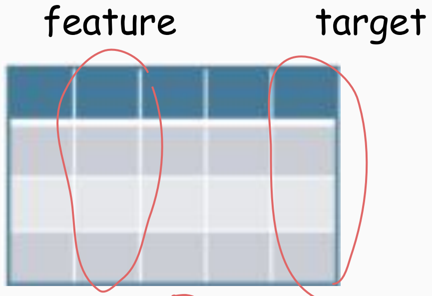
- Numerical variable: “Float, Integer”
  - Discrete numerical variable
    e.g: `[1, 2, 3, 4]`
  - Continuous numerical variable
    e.g: `[1.23, 0.87, 1.5498, -0.3146]`
- Nominal (Categorical) variable: “String, Integer”
  e.g: Geography: `['France', 'Germany', 'Spain']`
  e.g: Address:
  1F., No.1, Bilong Ln., Zhongzheng 1st Rd., Yingge Dist., New Taipei City 239, Taiwan (R.O.C.)
  - Ordinal nominal variable: “String, Integer”
    e.g: Size of clothes: `['S', 'M', 'L', 'XL']`

What We Want to Explore

- Numerical variables:
  mean, std, median, quartiles, deciles
  data distribution (histogram)

- Nominal variables: frequency distribution

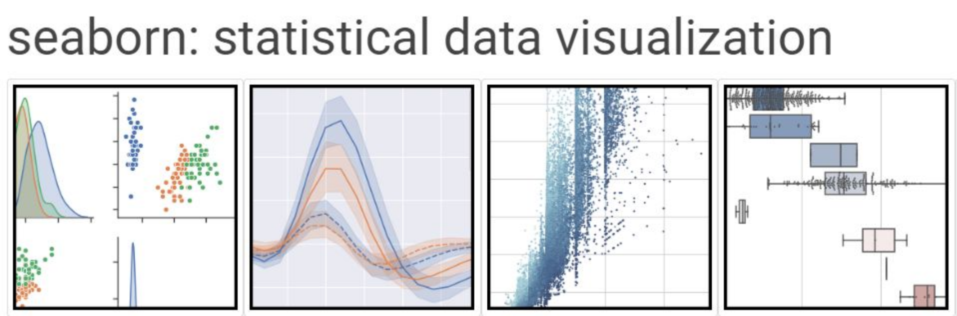
- Relationship between variables
  - Numerical x Numerical
  - Numerical x Nominal
  - Nominal x Nominal



What we care is the relation between **Feature and Label**

EDA Tools

Data read write, data clean, data transformation, data join ...



Data visualization

SM StatsModels
Statistics in Python

Statistical test, basic model

Numerical Variable Visualized

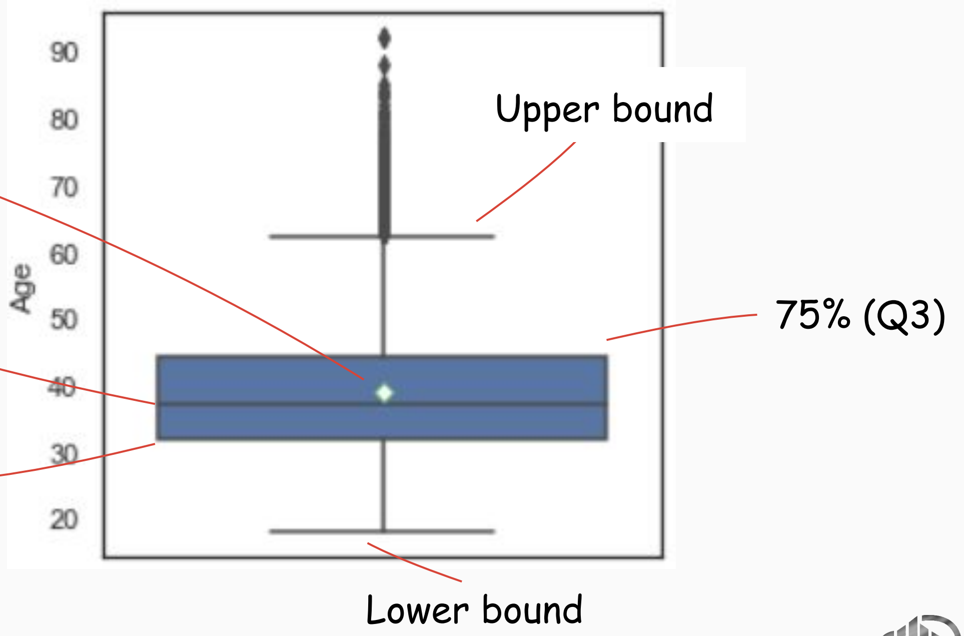
Visualization - Numerical Variable (Boxplot)

pandas.Series.describe()

```
raw.Age.describe()

count    7500.000000
mean       39.004267
std        10.500007
min        18.000000
25%        32.000000
50%        37.000000
75%        44.000000
max        92.000000
Name: Age, dtype: float64
```

```
sns.boxplot(raw.Age,
      showmeans=True, orient='v',
      meanprops={'marker':'D','markerfacecolor':'white'})
```



Visualization - Numerical Variable (Boxplot)

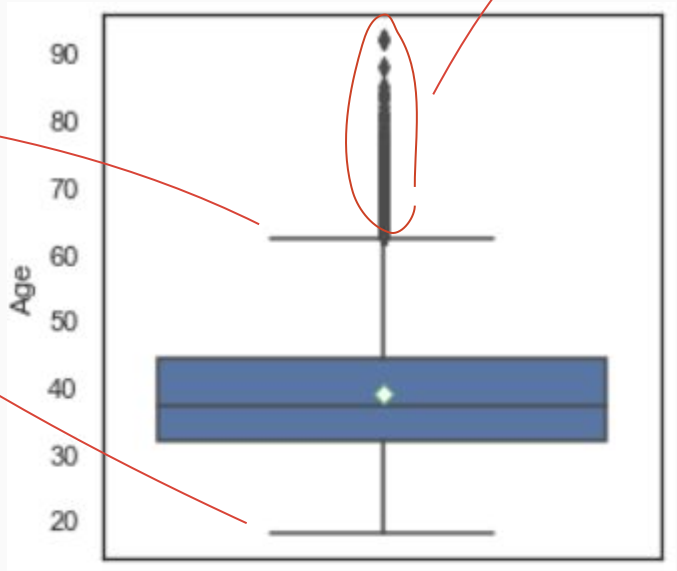
Interquatile range

IQR = Q3 - Q1

Upper bound = Q3 + IQR x 1.5

Lower bound = Q1 - IQR x 1.5

Outlier

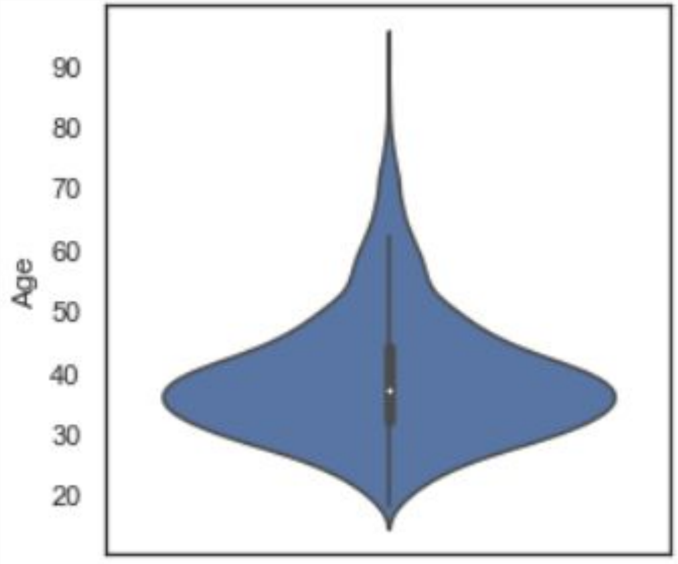


Visualization - Numerical Variable (Violinplot)

```
sns.violinplot(raw.Age, orient='v')
```

Include information

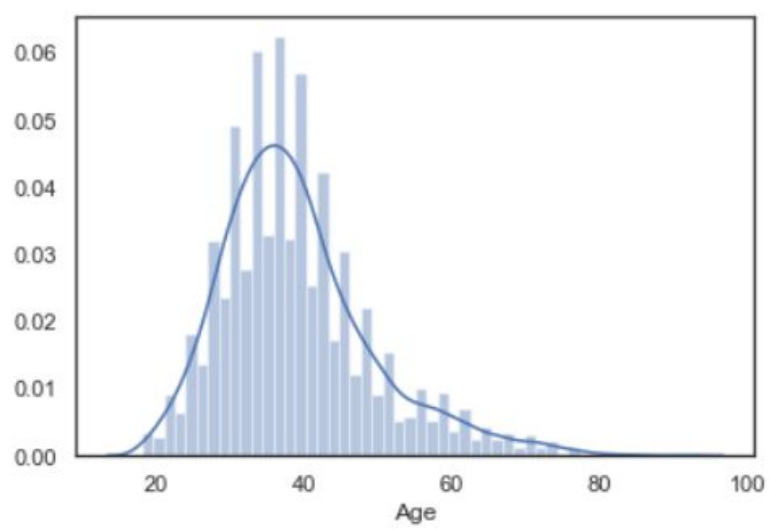
- Boxplot
- Data distribution



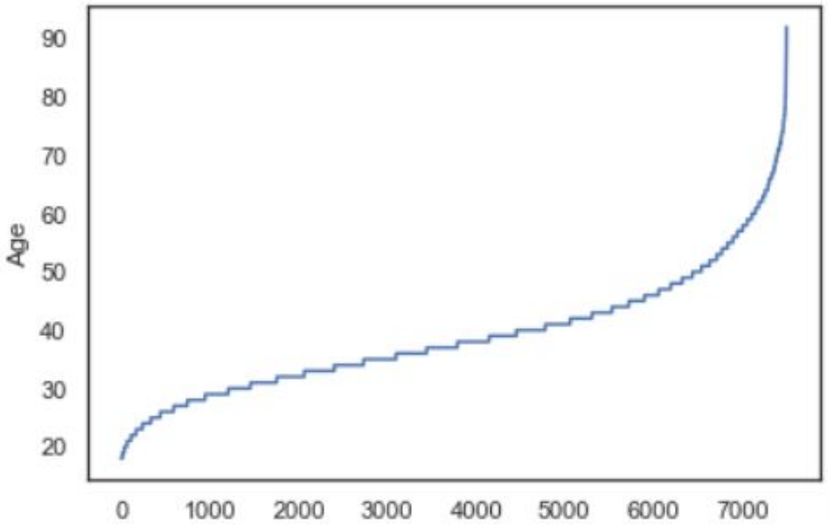
CloudMile

Visualization - Numerical Variable (Distplot, Lineplot)

```
sns.distplot(raw.Age)
```



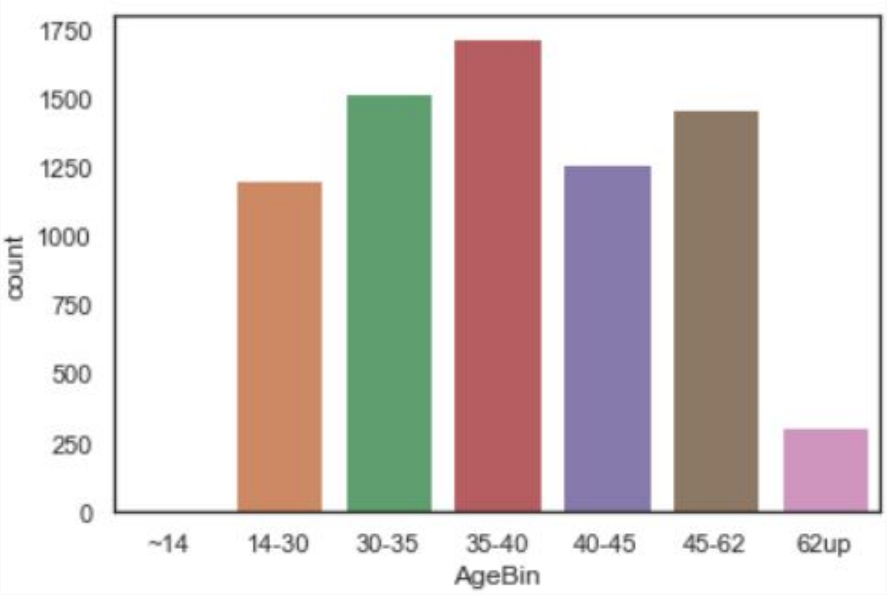

```
sns.lineplot(
    np.arange(len(raw)),
    raw.Age.sort_values())
```



Nominal Variable Visualized

Visualization - Nominal Variable (Countplot)

```
sns.countplot(
    x="AgeBin",
    data=raw)
```



`pandas.Series.value_counts`

```
raw.AgeBin.value_counts()

14-30    1209.0
30-35    1525.0
35-40    1721.0
40-45    1267.0
45-62    1465.0
62up      313.0
Name: AgeBin, dtype:
float64
```

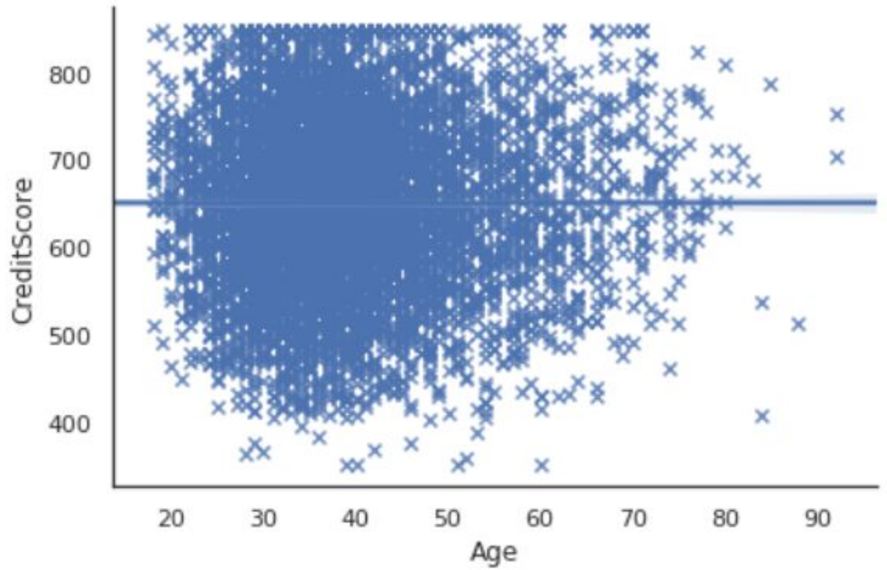
Multivariate Visualized

Multivariate Visualized - Numerical x Numerical

Regression Plot

⇒ sns.lmplot

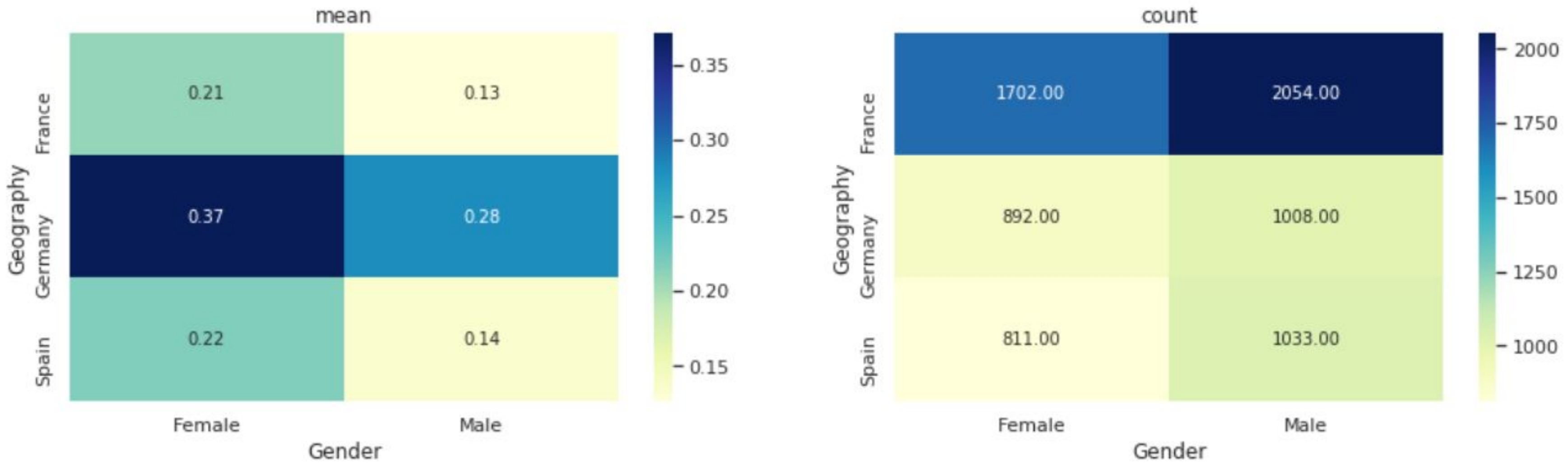
```
g = sns.lmplot(x='Age', y='CreditScore',
               # hue='Exited',
               data=raw, height=4, aspect=1.5, markers='x')
```



Multivariate Visualized - Nominal x Nominal

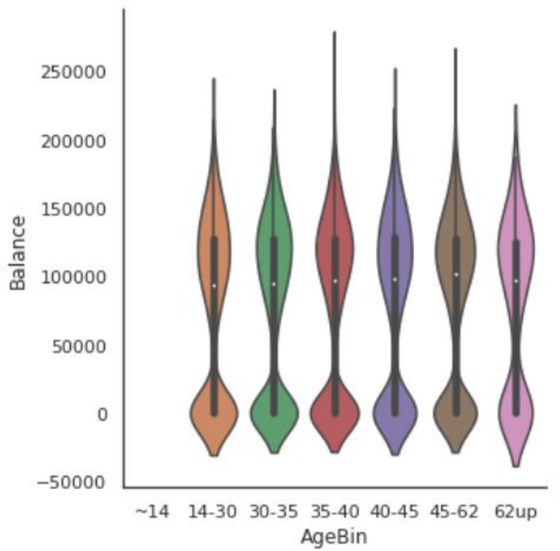
Heatmap ⇒ sns.heatmap

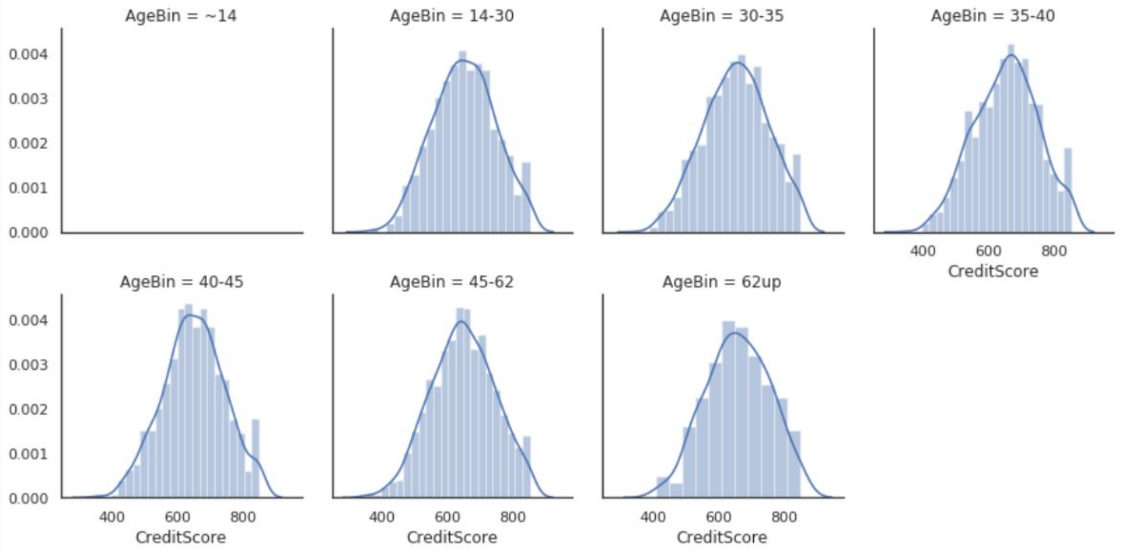
```
heatmap(raw, 'Geography', 'Gender', target='Exited')
```



Multivariate Visualized - Nominal x Numerical

Divide numerical variable by nominal variable !
⇒ Violinplot, Boxplot, Distplot, Lineplot ...





About The Statistical

Numerical x Numerical (Linear)

Pearson Correlation Coefficient

$$\rho = \frac{cov(X,Y)}{\sigma_x \sigma_y} = \frac{\sum (x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum (x_i - \bar{x})^2 (y_i - \bar{y})^2}}$$

$-1 < \rho < 1$

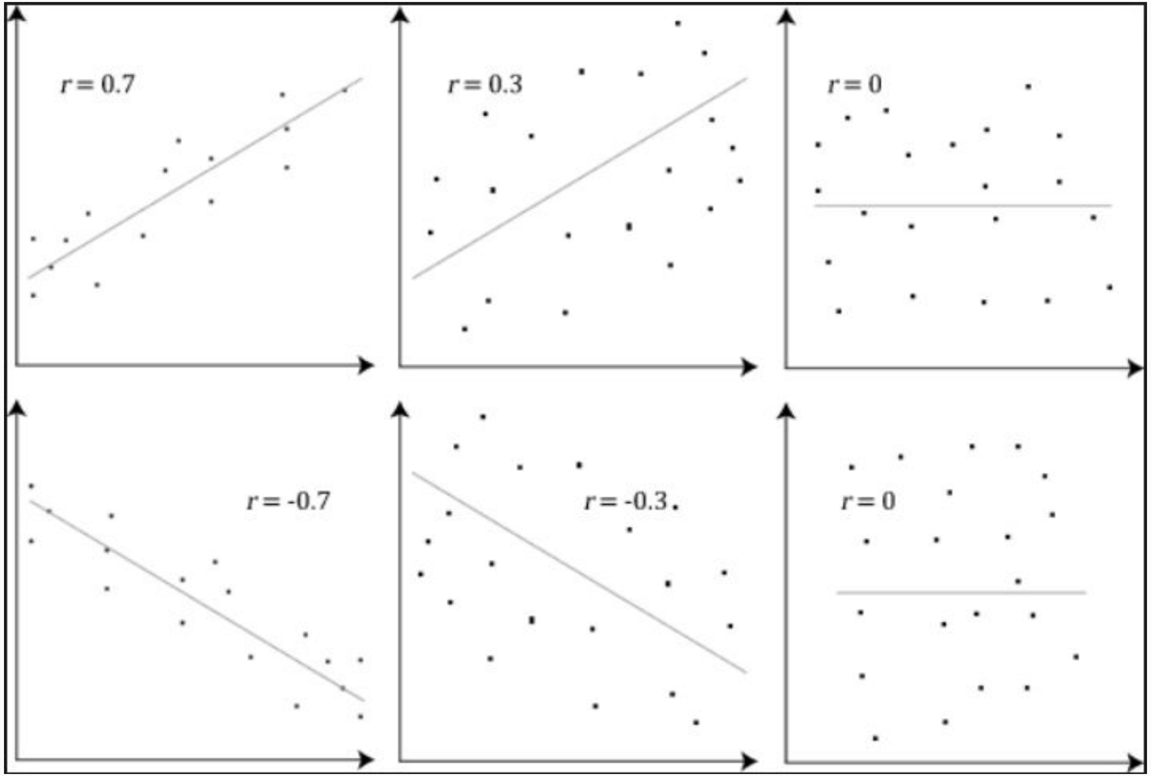
```
pd.DataFrame.corr()

data = data[
       ['Tenure', 'Balance', 'EstimatedSalary']]
data.corr(method='pearson')
```

	Tenure	Balance	EstimatedSalary
Tenure	1.000000	-0.012194	0.014443
Balance	-0.012194	1.000000	0.010461
EstimatedSalary	0.014443	0.010461	1.000000

Numerical x Numerical (Linear)

Pearson Correlation Coefficient



Numerical x Numerical (Linear)

Spearman's rank correlation coefficient

$$\rho = 1 - \frac{6 \sum d_i^2}{n(n^2 - 1)}$$

Difference between rank of two variables

$-1 < \rho < 1$

Data size

```
pd.DataFrame.corr()

data = data[
        ['Tenure', 'Balance', 'EstimatedSalary']]
data.corr(method='spearman')
```

	Tenure	Balance	EstimatedSalary
Tenure	1.000000	-0.008285	0.014451
Balance	-0.008285	1.000000	0.006840
EstimatedSalary	0.014451	0.006840	1.000000

Numerical x Numerical (Linear)

Spearman's rank correlation coefficient

	x	y	x_level	y_level	d_i	d_i^2
0	86	0	1	1	0	0
1	97	20	2	6	-4	16
2	99	28	3	8	-5	25
3	100	27	4	7	-3	9
4	101	50	5	10	-5	25
5	103	29	6	9	-3	9
6	106	7	7	3	4	16
7	110	17	8	5	3	9
8	112	6	9	2	7	49
9	113	12	10	4	6	36

$$\rho = 1 - \frac{6 \times 194}{10(10^2 - 1)} = -0.175175$$

Negative correlation, but not significant

Beware the **scaler information lost**, good for ordinal variable

CloudMile

Chi-Square Test

About The Statistical Test

Before The Chi-Square Test ..., About The Statistical Experiment

1. Hypothesis
   a. H_0: Null Hypothesis
      e.g: Vaiable $\mathbf{X_1}$**,** $\mathbf{X_2}$ independent
   b. H_1: Alternative Hypothesis
      e.g: Vaiable $\mathbf{X_1}$ association with $\mathbf{X_2}$

2. Significant Level $\boldsymbol{\alpha}$, Confidence Interval (1 - $\boldsymbol{\alpha}$)
   e.g: Given $\boldsymbol{\alpha} = \mathbf{0.05}$**,** means confidence interval = **0.95**

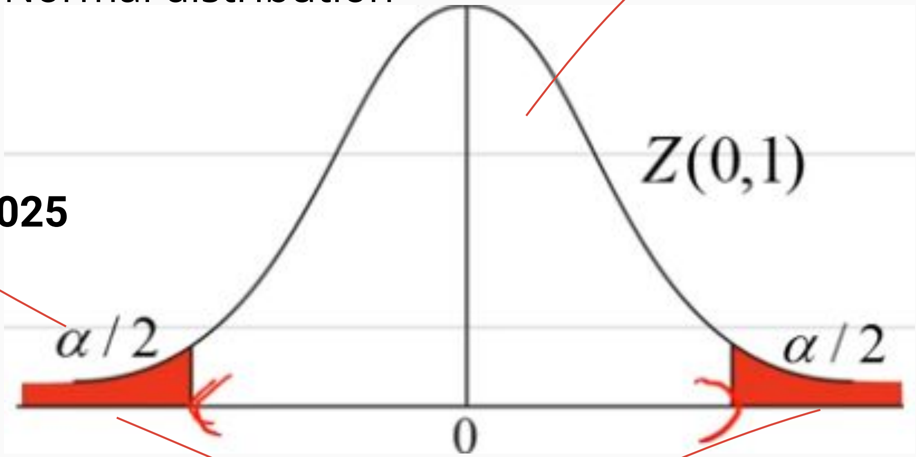
3. The p-value

4. When to reject H_0? (Means the result is significant)
   H_0: Negative event
   H_1: Positive event

About The Statistical Test

95% Confidence Interval

Normal distribution



α / 2 = 0.025

Reject Region

Prediction

		H_0	H_1
Ground Truth	H_0	TN	FP
	H_1	FN	TP

Type I error => FP

Type II error => FN

α: Significant level

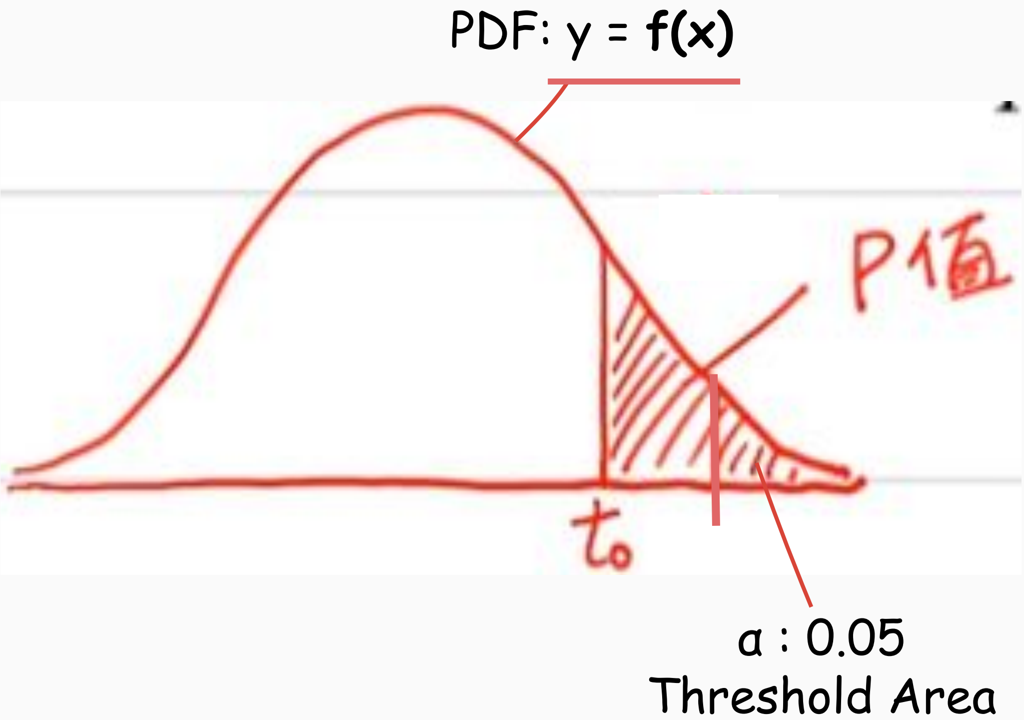
⇒ The probability of type I error occured

[Example of Type I II error]

CloudMile

About The Statistical Test

When is the experiment significant?



PDF: Probability Density Function

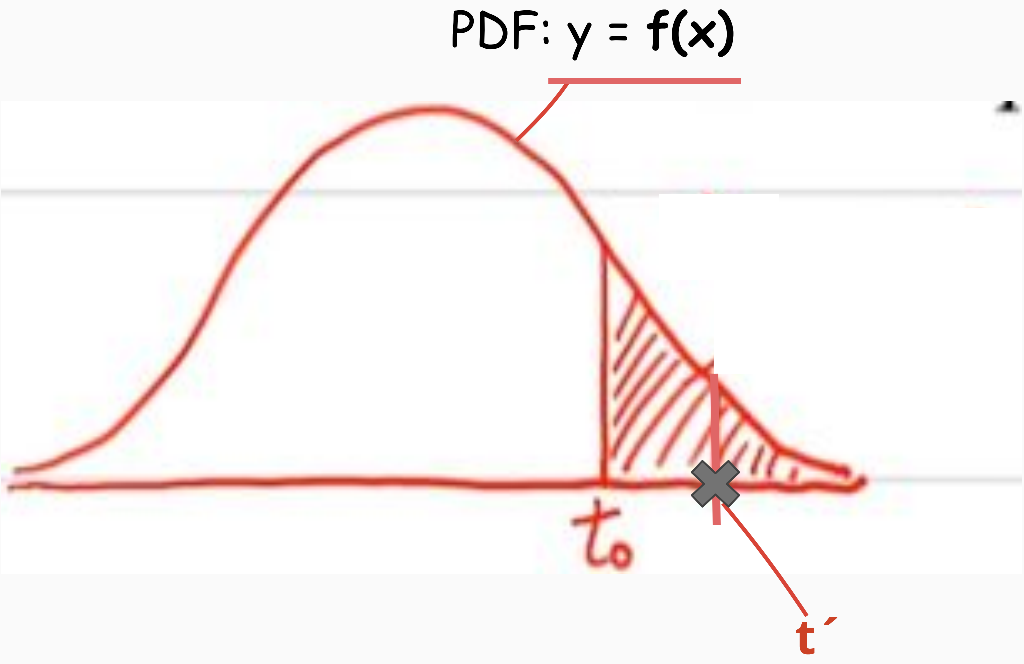
t_0: Statistic value

p-value: Area under **PDF** and **t_0**

$P < \alpha$: Reject H_0
$P >= \alpha$: Not reject H_0

About The Statistical Test

When is the experiment significant?



PDF: Probability Density Function

t_0: Statistic value

$t_0 > t'$: Reject H_0
$t_0 <= t'$: Not reject H_0

About The Statistical Test

- Goodness of Fit
  - H_0: There is no difference between the observed and expected frequencies
  - H_1: There is a difference between the observed and the expected frequencies

- Test for homogeneity
  - H_0: Populations follow the same probability distribution
  - H_1: One of populations dosen't follow the specific probability distribution

- Test for Independent
  - H_0: Nominal vaiables $\mathbf{X_1}$, $\mathbf{X_2}$ independent
  - H_1: Nominal vaiables $\mathbf{X_1}$ association with $\mathbf{X_2}$

CloudMile

About The Statistical Test

H_0: Vaiable $\mathbf{X_1}$, $\mathbf{X_2}$ independent H_1: Vaiable $\mathbf{X_1}$ association with $\mathbf{X_2}$

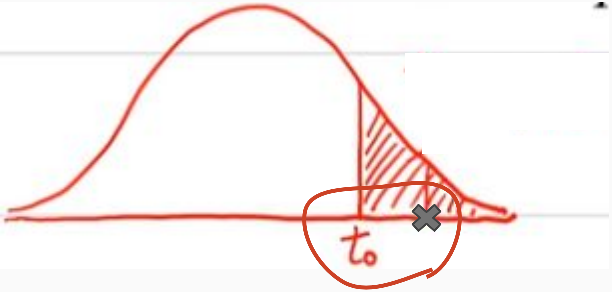
marit	educ
Never married	PhD or higher
Married	Middle school or lower
Divorced	Bachelor's
Widowed	PhD or higher
Married	PhD or higher

$$\chi^2 = \sum_i \frac{(Actual_i - Expected_i)^2}{Expected_i}$$

t_0: Chi-Square Statistic Value

Marital Status by Education | n = 300

	Middle school or lower	High school	Bachelor's	Master's	PhD or higher	Total
Never married	18	36	21	9	6	90
Married	12	36	45	36	21	150
Divorced	6	9	9	3	3	30
Widowed	3	9	9	6	3	30
Total	39	90	84	54	33	300



Nominal x Nominal (Non-Linear): Chi-Square Test

H_0: Vaiable $\mathbf{X_1}$, $\mathbf{X_2}$ independent H_1: Vaiable $\mathbf{X_1}$ association with $\mathbf{X_2}$

Marital Status by Education | n = 300

	Middle school or lower	High school	Bachelor's	Master's	PhD or higher	Total
Never married	18	36	21	9	6	90
Married	12	36	45	36	21	150
Divorced	6	9	9	3	3	30
Widowed	3	9	9	6	3	30
Total	39	90	84	54	33	300

$$\chi^2 = \sum_i \frac{(Actual_i - Expected_i)^2}{Expected_i}$$

Assume the events between **Education** and **Marital** status are mutual independent

$$P(A \cap B) = P(A)P(B)$$

P(**Education** and **Marital**) = P(**Education**) x P(**Marital**)

Nominal x Nominal (Non-Linear): Chi-Square Test

H_0: Vaiable $\mathbf{X_1}$, $\mathbf{X_2}$ independent H_1: Vaiable $\mathbf{X_1}$ association with $\mathbf{X_2}$

Marital Status by Education | n = 300

	Middle school or lower	High school	Bachelor's	Master's	PhD or higher	Total
Never married	18	36	21	9	6	90
Married	12	36	45	36	21	150
Divorced	6	9	9	3	3	30
Widowed	3	9	9	6	3	30
Total	39	90	84	54	33	300

$$\chi^2 = \sum_i \frac{(Actual_i - Expected_i)^2}{Expected_i}$$

Actual(**Master's** x **Married**) = 36
Expected(Master's x Married) = P(**Master's**) x P(**Married**) x Length(data)
= (54 / 300) x (150 / 300) x 300
= 0.18 x 0.5 x 300
= 27

⇒ $(36 - 27)^2 / 27 = 3.0$

CloudMile

Nominal x Nominal (Non-Linear): Chi-Square Test

statsmodels package example

```
# Calculate chi-square value, p-value, degree of freedom, expected value
chi, pv, df, expected = stats.chi2_contingency(observed=data)

# check if chi-square value > criterion(95% confidence interval)
crit = stats.chi2.ppf(q=0.95, df=df)

print(f'chi-square value: {chi}, criterion: {crit}')
```

```
chi-square value: 1022.025, criterion: 11.070
result: True
```

Significant! Reject H_0

There is a relationship between X_1, X_2

ANOVA (Analysis of Variance)

Nominal x Numerical (Non-Linear): ANOVA

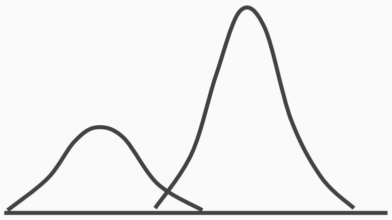
Assume we have k group, k > 1

H_0: $\mu 1 = \mu 2 \ \ldots = \mu k$

H_1: Means are not all equal.

Prerequisite

- ➔ Each groups approximately follow **normal distribution (Guassian distribution)**
- ➔ Independent cases
- ➔ Equality (or "homogeneity") of variances



Welch test, Brown Forsythe test...

??% Confidence in ANOVA test

CloudMile

Nominal x Numerical (Non-Linear): ANOVA

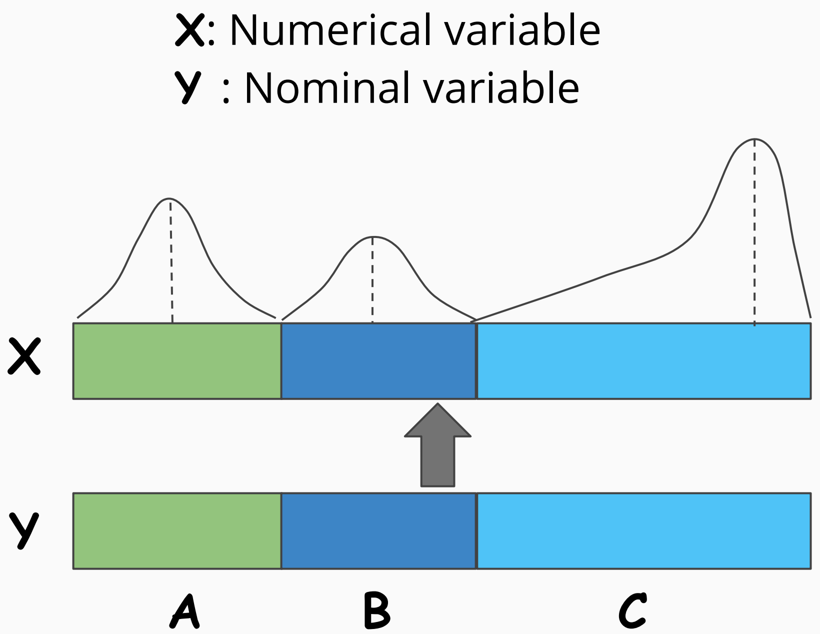
Assume we have k group, k > 1

H_0: $\mu 1 = \mu 2 \ \ldots = \mu k$

H_1: Means are not all equal.

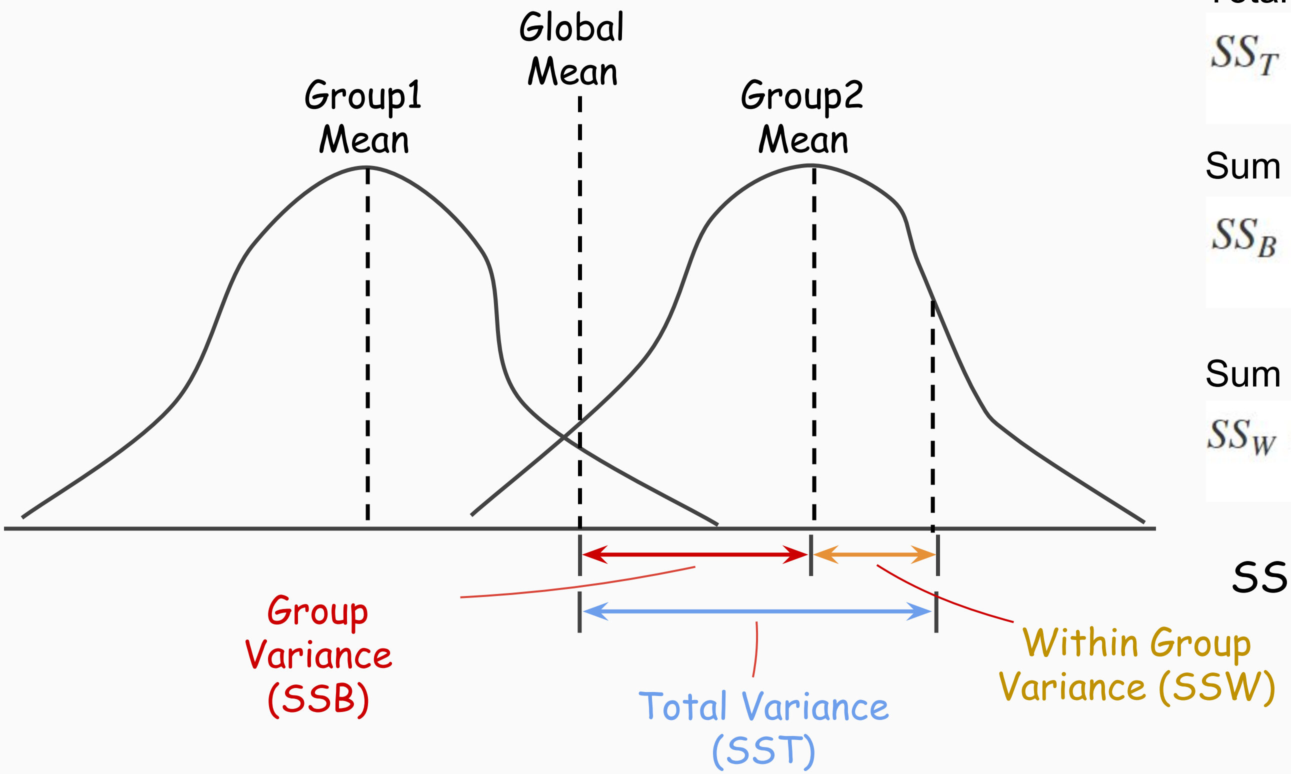
Prerequisite

- ➔ Each groups approximately follow **normal distribution (Guassian distribution)**
- ➔ Independent cases
- ➔ Equality (or "homogeneity") of variances



CloudMile

Nominal x Numerical (Non-Linear): ANOVA



Total sum of square

$$SS_T = \sum_i (X_i - \overline{X_{all}})^2$$

Sum of square between groups

$$SS_B = \sum_{\#group} (\overline{X_{group}} - \overline{X_{all}})^2$$

Sum of square within groups

$$SS_W = \sum_{\#group} \sum_{\#within\_group} (X_i - \overline{X_{group}})^2$$

SST = SSB + SSW

Nominal x Numerical (Non-Linear): ANOVA

$$MS_B = \frac{SS_B}{df_B}$$

$$MS_W = \frac{SS_W}{df_W}$$

$$\rightarrow \frac{MS_B}{MS_W}$$

	SS	DF	MS	F	P
B	SS_B	G - 1	MS_B	MS_B / MS_W	P-value
W	SS_W	(N - 1) - (G - 1)	MS_W		
T	SS_T	(N - 1)			

We hope the F Larger

Total sum of square

$$SS_T = \sum_i (X_i - \overline{X_{all}})^2$$

Sum of square between groups

$$SS_B = \sum_{\#group} (\overline{X_{group}} - \overline{X_{all}})^2$$

Sum of square within groups

$$SS_W = \sum_{\#group} \sum_{\#within\_group} (X_i - \overline{X_{group}})^2$$

Area => P < α: Reject H_0
P >= α: Not reject H_0

Nominal x Numerical (Non-Linear): ANOVA

statsmodels package example

```
def anova(formula, data):
    table  = sm.stats.anova_lm(
        ols(formula, data=data).fit(),
        typ=2
    )
    return table

anova("Age ~ C(Exited)", data=data)
```

“Numerical ~ C(Nominal)”

	sum_sq	df	F	PR(>F)
C(Exited)	63991.391600	1.0	629.02925	2.266897e-133
Residual	762774.471867	7498.0	NaN	NaN

P-value

CloudMile

Recap

- ❏ Single Variable
  - ❏ Numerical variable ⇒ Boxplot, Violinplot, Distplot, Lineplot
  - ❏ Nominal variable ⇒ Countplot

- ❏ Multi-Variable
  - ❏ Numerical x Numerical ⇒ Regression plot
  - ❏ Nominal x Nominal ⇒ Heatmap
  - ❏ Nominal x Numerical ⇒ Conditional Boxplot, Conditional Violinplot ...

- ❏ Relationship
  - ❏ Numerical x Numerical ⇒ Pearson, Spearman Correlation Coefficient
  - ❏ Nominal x Nominal ⇒ Chi-Square
  - ❏ Nominal x Numerical ⇒ ANOVA

CloudMile

Lab: Exploratory Data Analysis

Topic : Exploratory Data Analysis

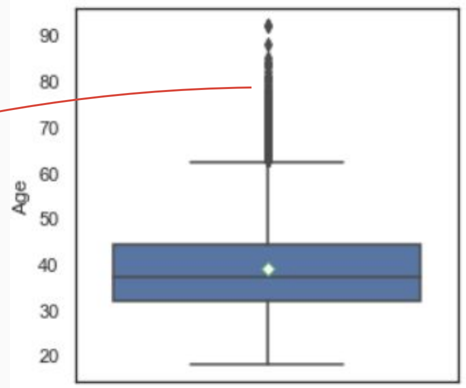
Filename	lab_eda_finance_customer_churn.ipynb
Data	Financial Customer Churn Prediction
Target	➔ Understand the data ➔ Features distribution ➔ Relationship between features or between features and label
Duration	About 20 min

Agenda

- Overall Workflow
- Exploratory Data Analysis
- Feature Engineering + Training

Basic - Data Clean

- ➔ Missing value in numerical variable
  - ◆ Fill **mean** or **median**
- ➔ Outlier in numerical variable
  - ◆ Utilize the quartile to find **outlier,** and fill mean or median

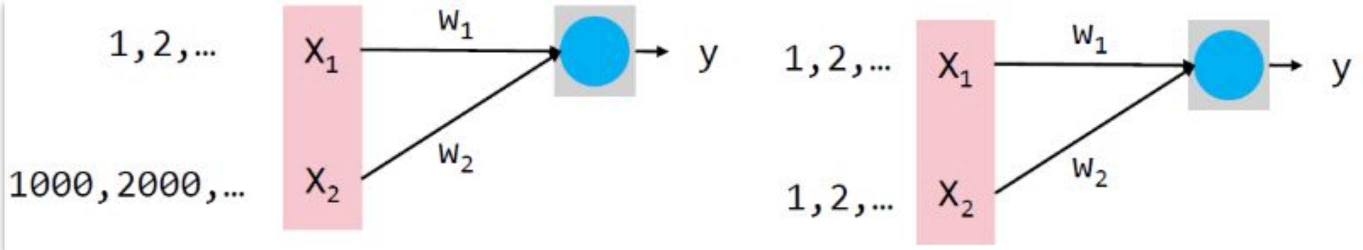


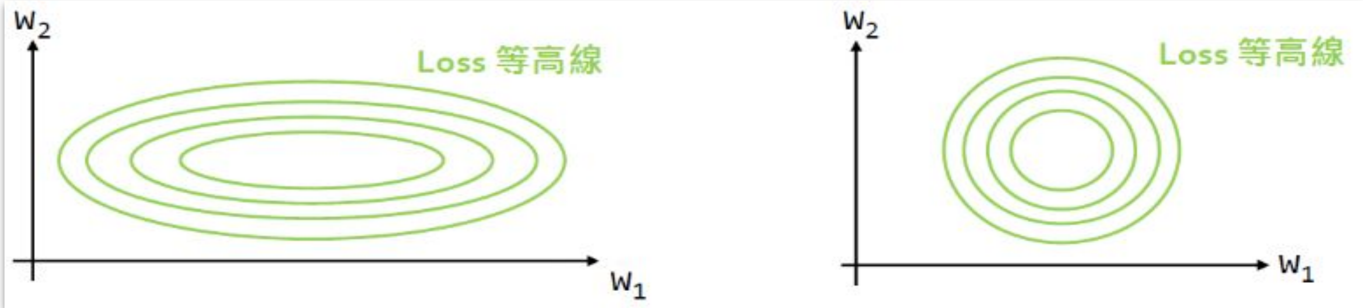
- ➔ Missing value in nominal variable
  - ◆ See missing value as an special class
  - ◆ Add a feature to describe missing value

nominal column	added
A	0
NaN	1
B	0

Basic

Numeric variable ⇒ Normalization





Basic

Numeric variable ⇒ Normalization

- ➔ Min max scaler to [0, 1] **(Beware outlier)**

$$\frac{x_i - min}{max - min}$$

- ➔ Scale to standard normal distribution (Z-score standardize)

$$\mu = 0,\ \sigma = 1$$

Basic

Nominal variable ⇒ One Hot Encoding

column
class A
class B
class C

1
2
3

X

1	0	0
0	1	0
0	0	1

O

all the pairwise distance
are the same ⇒ sqrt(2)

Before Advanced Feature Engineering

Before Advanced Feature Engineering

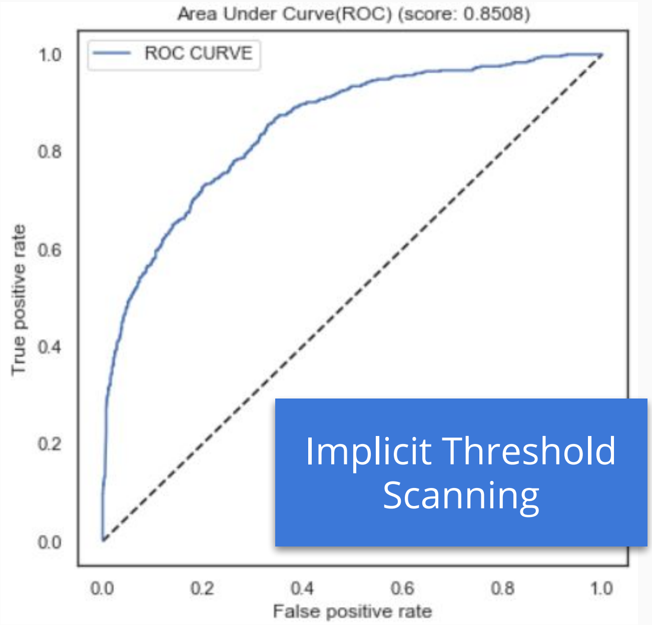
Topic : Knowing Programing Structure

Filename	lab_model_finance_customer_churn.ipynb
Data	Financial Customer Churn Prediction
Target	
Duration	

Before Advanced Feature Engineering

ROC: Receiver Operating Characteristic

TP Rate cross FP Rate
(0 < AUC < 1)



AUC = 0.5 (no discrimination)
0.7 ≦ AUC ≦ 0.8 (acceptable discrimination)
0.8 ≦ AUC ≦ 0.9 (excellent discrimination)
0.9 ≦ AUC ≦ 1.0 (outstanding discrimination)

		Prediction 0	Prediction 1
Ground Truth	0	TN	FP
	1	FN	TP

FP Rate = FP / (TN + FP)

TP Rate = TP / (FN + TP)

Before Advanced Feature Engineering

AUROC (Code)

```
from sklearn.metrics import roc_curve, auc

fpr, tpr, thres = roc_curve(y, pred, pos_label=1)
auc_scr = auc(fpr, tpr)
```

	fpr	tpr	threshold
0	0.000000	0.002045	0.999386
1	0.000000	0.104294	0.883024
2	0.000497	0.104294	0.880933
3	0.000497	0.110429	0.870801
4	0.000995	0.110429	0.870346
5	0.000995	0.118609	0.852421

Before Advanced Feature Engineering

Find Best Threshold ⇒ F Beta Score

$$F_\beta = (1+\beta^2)\cdot\frac{\text{precision}\cdot\text{recall}}{(\beta^2\cdot\text{precision})+\text{recall}}$$

Small β => prefer precision
Large β => prefer recall

Suggested adjust range
0.1 < β < 2

Prediction

Ground Truth	0	1
0	TN	FP
1	FN	TP

precision

recall

	precision	recall	f1-score	support
0	0.88	0.96	0.92	2011
1	0.73	0.45	0.56	489
avg / total	0.85	0.86	0.85	2500

CloudMile

Before Advanced Feature Engineering

Function **f_beta_scann**

```python
def f_beta_scann(y_true, y_pred, beta=0.5):
    """F beta score掃描找出最佳threshold"""

    y_pred = pd.Series(y_pred.ravel())
    # 切割100等分, 尋找最佳 f beta score
    bins = np.linspace(y_pred.min(), y_pred.max(), 100)
    # 找出F beta score最高的點
    result =
        np.array([
            precision_recall_fscore_support(
                y_true=y_true,
                y_pred=y_pred > thres, beta=beta)[2][1]
          for thres in bins])

    best_idx = result.argmax()
    return bins[best_idx], result[best_idx]
```

	0	1
precision		
Recall		
F score		
???		

Advanced Feature Engineering

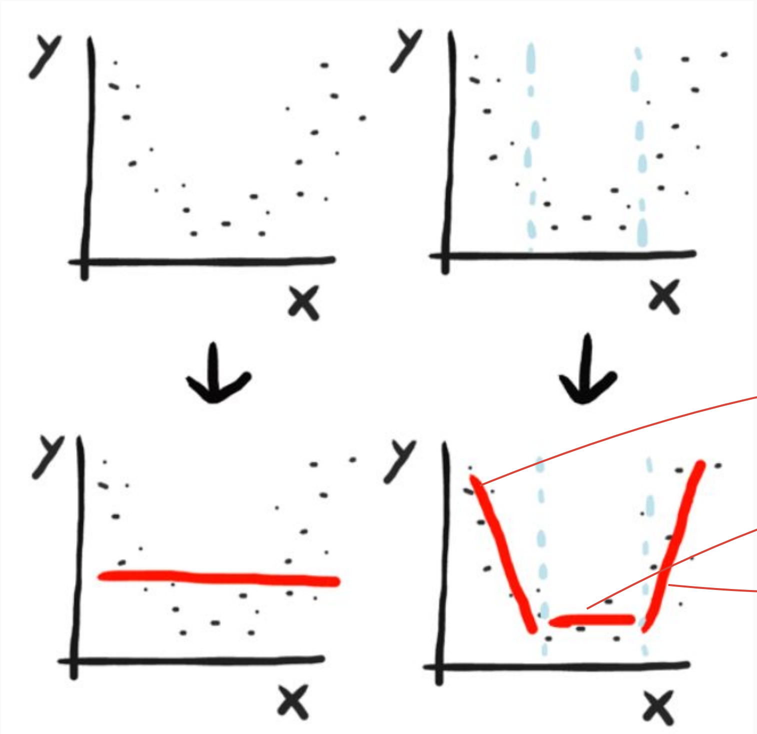
Advanced Feature Engineering

Coming up with features is difficult, time consuming, requires expert knowledge.
"**Applied machine learning**" is basically feature engineering.

-Andrew Ng

Advanced - Binning Numerical Variable

Binning ⇒ Find The Non-Linear Relationship



$$Y = WX + b$$

$$Y = W_1X_1 + W_2X_2 + W_3X_3 + b$$

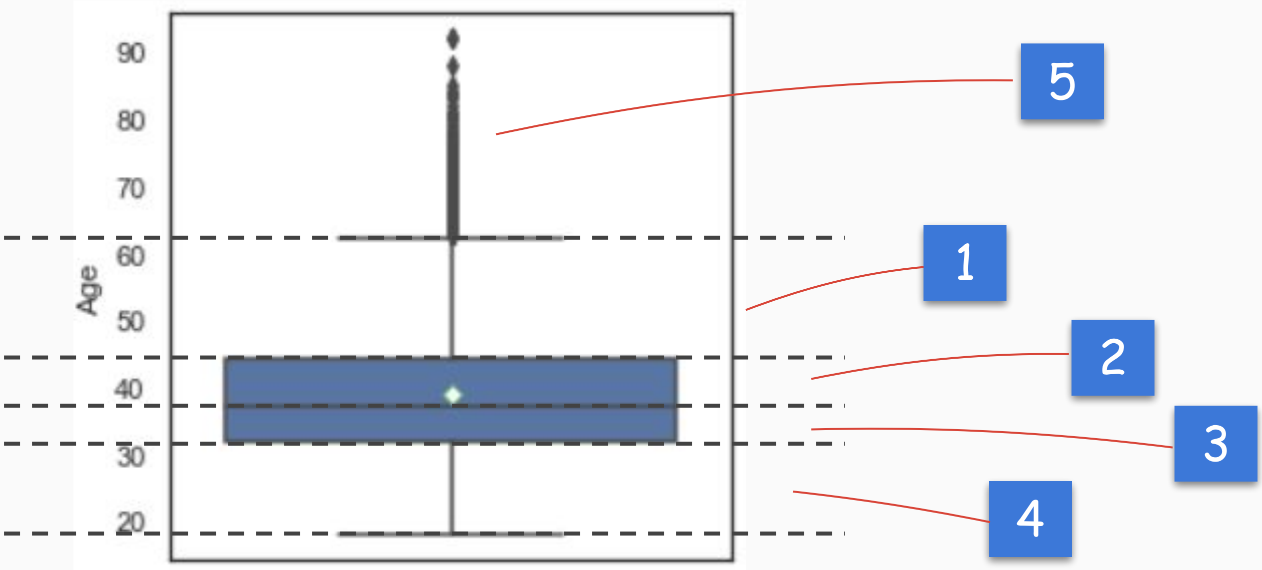
$$Y = W_1X_1 + \qquad\qquad + b$$

$$Y = \qquad W_2X_2 \qquad + b$$

$$Y = \qquad\qquad W_3X_3 + b$$

Advanced - Binning Numerical Variable

Binning Example ⇒ Quartile Cut



Advanced - Binning Numerical Variable

Binning Example ⇒ Quartile Cut

```
def quartile_binning(x):
    # Quartile cut
    bins = np.percentile(x, range(0, 100, 25))[1:].tolist()
    # IQR
    iqr_x_150 = (bins[-1] - bins[0]) * 1.5

    bins = [bins[0] - iqr_x_150] + bins + [bins[-1] + iqr_x_150]
                Lower bound                      Upper bound

    result = pd.Series(np.digitize(x, bins)) \
                .map(pd.Series([0, 1, 2, 3, 4, 0])).values
    return result, bins
```

Lab

Topic : Try Binning

Filename	lab_eda_finance_customer_churn.ipynb
Data	Financial Customer Churn Prediction
Target	Add binning features: ➔ Age, HasBalance, CreditScore, Tenure, EstimatedSalary
Duration	About 10 min

Advanced - WOE Encoding (Only for Binary Classification)

Weight of Evidence

Inspired from **Logistic Regression** ⇒ Binary Classification ⇒ 0 or 1

$$\underline{\underline{\sigma(wx+b)}} \qquad z = wx + b \qquad \boxed{\sigma = \frac{1}{1+e^{-z}}}$$

LR function

Odds Ratio ⇒ $\dfrac{P}{1-P}$

$$\frac{P}{1-P} = \frac{\frac{1}{1+e^{-z}}}{1-\frac{1}{1+e^{-z}}} = \ldots = e^{z} = e^{wx+b} \Rightarrow log(\frac{P}{1-P}) = wx + b$$

For the interpretable

CloudMile

Advanced - WOE Encoding (Only for Binary Classification)

Cancer Prediction Example ➡ $wx + b = 0.095 \cdot age + 2.645$

Log Odds Ratio $log(\frac{P}{1-P}) = 0.095 \cdot age + 2.645$

Odds Ratio $e^{log(\frac{P}{1-P})} = e^{0.095age+2.645} = e^{0.095age}e^{2.645}$

If age increase 1

$$e^{0.095(age+1)+2.645} = e^{0.095age}e^{0.095}e^{2.645}$$

(after increase) / (original)

$$\frac{e^{0.095age}e^{2.645}e^{0.095}}{e^{0.095age}e^{2.645}} = e^{0.095} = 1.099$$

When age add 1, the odds ration of have cancer increase 1.099

Advanced - WOE Encoding (Only for Binary Classification)

Weight of Evidence

$$WoE = \ln\left(\frac{\%\, non - events}{\%\, events}\right)$$

To avoid division by zero

$$WoE_{adj} = \ln\left(\frac{\text{Number of non-events in a group} + 0.5 \,/\, \text{Number of non-events}}{\text{Number of events in a group} + 0.5 \,/\, \text{Number of events}}\right)$$

Advanced - WOE Encoding (Only for Binary Classification)

Weight of Evidence

Feature	Outcome	WoE
A	1	0.4
A	0	0.4
A	1	0.4
A	1	0.4
B	1	0.74
B	1	0.74
B	0	0.74
C	1	-0.35
C	1	-0.35

	Non-events	Events	% of non-events	% of events	WoE
A	1	3	50	42	$\ln\left(\frac{(1+0.5)/2}{(3+0.5)/7}\right) = 0.4$
B	1	2	50	29	$\ln\left(\frac{(1+0.5)/2}{(2+0.5)/7}\right) = 0.74$
C	0	2	0	29	$\ln\left(\frac{(0+0.5)/2}{(2+0.5)/7}\right) = -0.35$

Advanced - WOE Encoding (Only for Binary Classification)

Function **do_woe_encoding**

```python
total_vc = data[label].value_counts().sort_index()
def woe(pipe, total_vc):
    # Count by label in this group
    group_vc = pipe[label].value_counts().sort_index()

    # Some class in the feature is missing, fill zero to missing class
    if len(group_vc) < len(total_vc):
        for key in total_vc.index:
            if key not in group_vc:
                group_vc[key] = 0.
        group_vc = group_vc.sort_index()

    # WOE formula
    r = ((group_vc + 0.5) / total_vc).values

    # Odd ratio => 1 to 0, you can define meaning of each class
    return np.log(r[1] / r[0])

return data.groupby(x).apply(lambda pipe: woe(pipe, total_vc))
```

	Group dist	Global dist
0	0	2
1	2	7

Advanced - Target Encoding

Nominal Variable Frequency Encoding

Feature	Encoded Feature
A	0.44
A	0.44
A	0.44
A	0.44
B	0.33
B	0.33
B	0.33
C	0.22
C	0.22

A	0.44 (4 out of 9)
B	0.33 (3 out of 9)
C	0.22 (2 out of 9)

Advanced - Target Encoding

Nominal Variable Mean Encoding

Feature	Outcome	MeanEncode
A	1	0.75
A	0	0.75
A	1	0.75
A	1	0.75
B	1	0.66
B	1	0.66
B	0	0.66
C	1	1.00
C	1	1.00

A	0.75 (3 out of 4)
B	0.66 (2 out of 3)
C	1.00 (2 out of 2)

Lab

Topic : WOE + Target Encoding

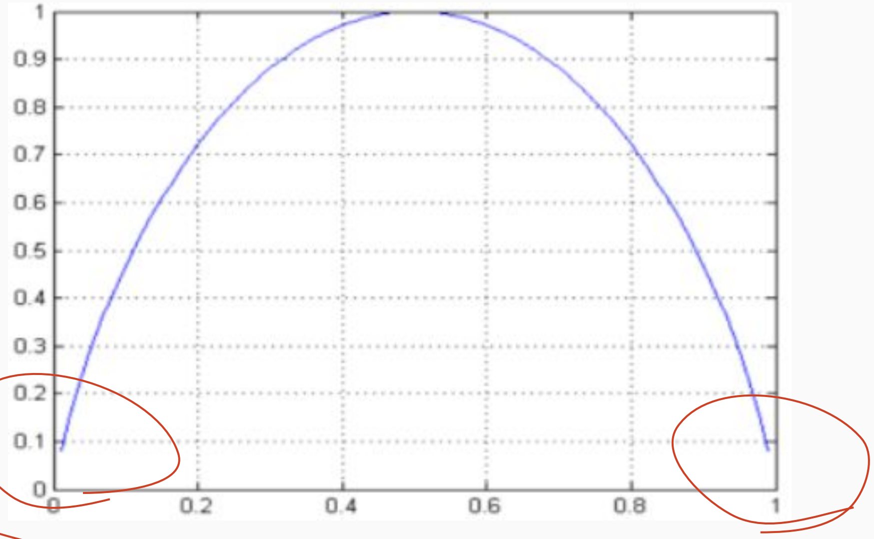
Filename	lab_eda_finance_customer_churn.ipynb
Data	Financial Customer Churn Prediction
Target	➔ Add WOE Encoding Feature ➔ Add Target Encoding Feature
Duration	About 15 min

Advanced - Entropy Encoding

Entropy ⇒ Define the events first!

$$-\sum_i p_i \times log(p_i)$$

Flip Coin Example



Lower entropy give more information !

CloudMile

Advanced - Entropy Encoding

Entropy

Feature	Outcome
A	1
A	0
A	1
A	1
B	1
B	1
B	0
C	1
C	1

Evnet: 0, 1

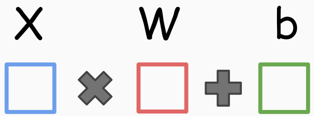
entropy(A) = - (1/4 x log(1/4) + 3/4 x log(3/4))
= 0.81
proba(A) = 4/9

=> result = 0.81 * 4/9 = 0.36

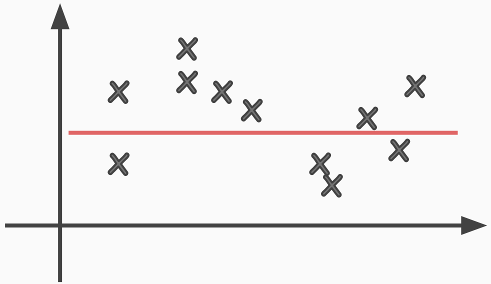
entropy(C) = -(1 x log(1)) = 0

CloudMile

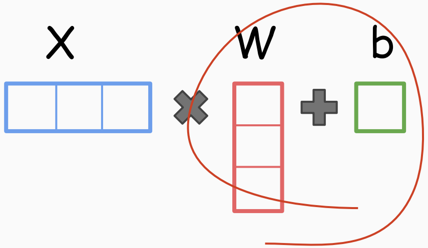
Advanced - Polynomial Encoding



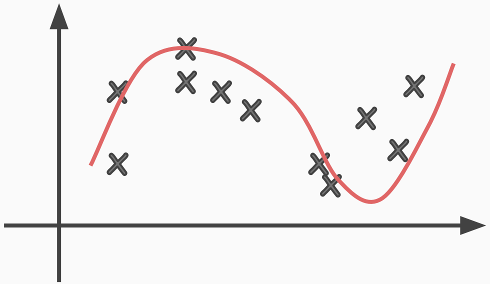
$$Y = W_1X + b$$



Normal



$$Y = W_1X + W_2X^2 + W_2X^3 + b$$



Better fitting capability

```
from keras.layers import Dense

nets = Dense(units=64, ...)
nets = Dense(units=32, ...)
logits = Dense(units=1, ...)
```

Tensorflow Playground

CloudMile

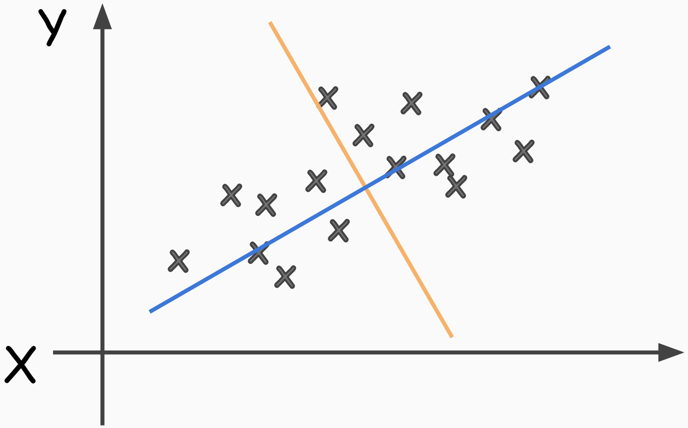
Lab

Topic : Entropy Encoding + Polynomial Encoding

Filename	lab_eda_finance_customer_churn.ipynb
Data	Financial Customer Churn Prediction
Target	➔ Add Entropy encoding ➔ Add Polynomial encoding
Duration	About 10 min

Advanced - PCA (Principal Component Analysis)

Sometimes we could face the curse of dimensionality



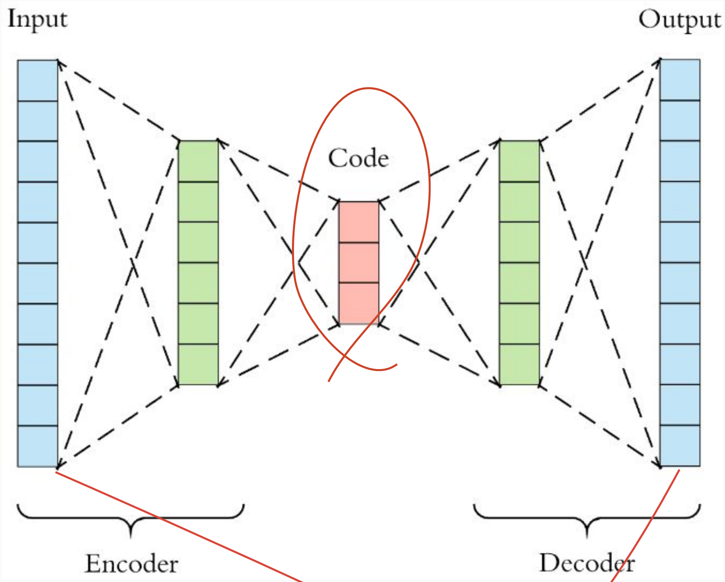
- Reduce the dimension of data
- Coordinate system transformation
- Mutually orthogonal axis
- Linear transformation

```
from sklearn.decomposition import PCA

pca = PCA(32)
data = pca.fit_transform(data)
```

So simple ! thank god we have scikit learn

Advanced - AutoEncoder



Minimize Loss(input, output)

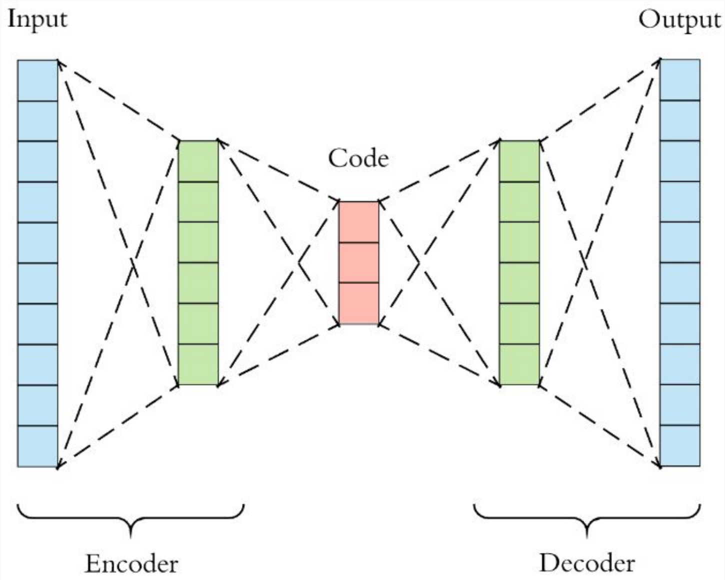
- Goal
  - Learn a specified data representation
  - Reduce the dimension of data

- Unsupervised learning
  - The input is the label

- Can be non-linear transformation
- The **"Code"** is what we want

CloudMile

Advanced - AutoEncoder



	AutoEncoder	PCA
Linear?	Non-linear	Linear
Dimension limitation	Non-limited	Less than input dimension

Advanced - AutoEncoder

```
inputs = Input(shape=(inputs_dim, ))
# Encoder
encoded = Dense(inputs_dim, activation='selu')(inputs)
encoded = Dense(128, activation='selu')(encoded)
encoded = Dense(64, activation='selu')(encoded)

encoded = Dense(64, activation='selu')(encoded)

# Decoder
decoded = Dense(64, activation='selu')(encoded)
decoded = Dense(128, activation='selu')(decoded)
decoded = Dense(inputs_dim, activation='linear')(decoded)

# this model maps an input to its reconstruction
autoencoder = Model(inputs, decoded)
# Adam Optimizer + Mean square error loss
autoencoder.compile(optimizer='adam', loss='mse')

# this model maps an input to its encoded representation
encoder = Model(inputs, encoded)
```

Coded

AutoEncoder model for training

Encoder model for prediction

Lab

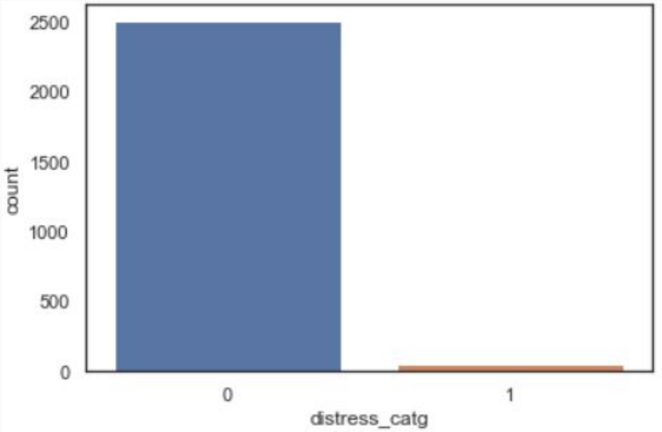
Topic : PCA Encoding + AutoEncoder Encoding

Filename	lab_eda_finance_customer_churn.ipynb
Data	Financial Customer Churn Prediction
Target	➔ Add PCA Encoding ➔ Add AutoEncoder Encoding
Duration	About 10 min

CloudMile

Advanced - Imbalanced Data For Classification

Majority Minority




```
class_weight: ...
    This can be useful to tell the model to
    "pay more attention" to samples from
    an under-represented class.
```

$$Loss = -\left(\underbrace{W * Y\log(\hat{Y})}_{\text{Minority}} + \underbrace{(1 - Y)\log(1-\hat{Y})}_{\text{Majority}} \right)$$

Minority : Majority = 1 : 40

class weights ⇒ {Minority : 40, Majority: 1}

CloudMile

Advanced - RFM (Recency, Frequency, Monetary)

How recently, how often and how much did they buy.

Train

Company	rfm_all_freq	distress_num	rfm_all_mean
36	1	0.026703	0.026703
36	2	0.020268	0.023485
36	3	-0.046938	0.000011
36	4	-0.290000	-0.072492
36	5	-0.447700	-0.147533
36	6	-0.333620	-0.178548

Test

Company	rfm_all_freq	distress_num	rfm_all_mean
36	6	?	-0.178548
36	6	?	-0.178548
36	6	?	-0.178548

Beware the "Data leakage", label not in test data, so we take the RFM value from the last moment of train data

CloudMile

Conclusion

- ➔ Domain knowlage is still the key of model performance
  ⇒ Why do you know RFM are good for transactional data?
- ➔ Deep learning can learning the feature transformation,
  but still got limitation
- ➔ Still need “a little” trial and error