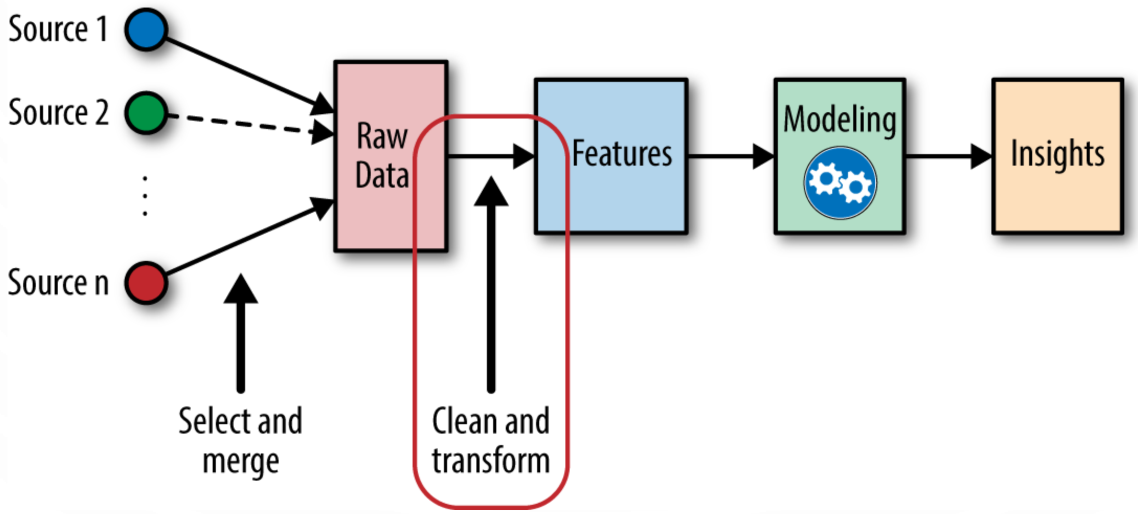
# 5.1 Feature Engineering

Dr. Sultan Alfarhood

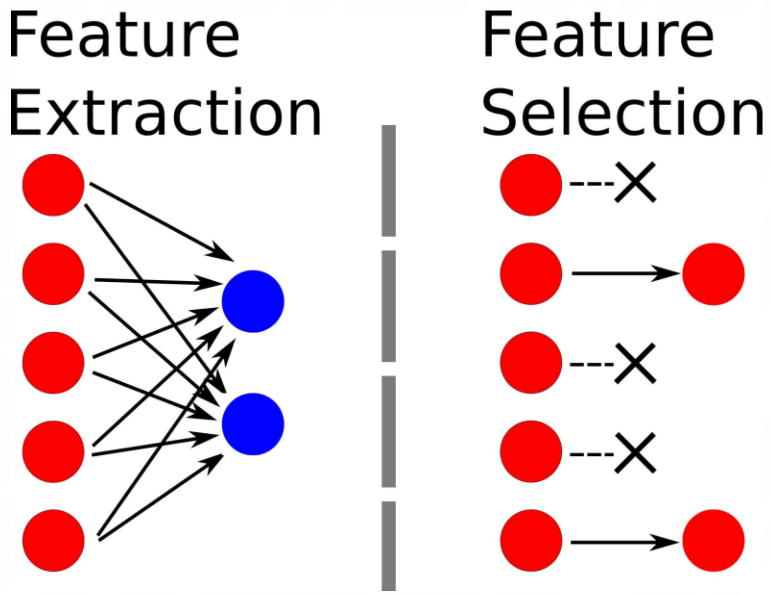
# Feature Engineering

- The problem of transforming raw data into a dataset is called feature engineering.
- **Informative features**: those would allow the learning algorithm to build a model that does a good job of predicting labels of the data used for training.
  - Highly informative features are also called features with high **predictive power**.



# Feature Selection vs Feature Extraction

- **Feature Selection**
  - Selecting subset of extracted features. This subset is relevant and contributes to minimizing the error rate of a trained model.
- **Feature Extraction**
  - Combining existing features to produce a more useful one.



# Label Encoding

- Encode attributes and target labels with value between 0 and NumberOfClasses-1
  - Using ordered numbers as values is likely to confuse the learning algorithm
- Label Encoding can be helpful when the ordering of values of some categorical variable matters

| ... | quality | ... |
|---|---|---|
| ... | bad | ... |
| ... | bad | ... |
| ... | good | ... |
| ... | excellent | ... |

| ... | quality | ... |
|---|---|---|
| ... | 0 | ... |
| ... | 0 | ... |
| ... | 1 | ... |
| ... | 2 | ... |

# One-Hot Encoding

Transforming categorical feature into several binary ones:

| id | color |
|---|---|
| 1 | red |
| 2 | blue |
| 3 | green |
| 4 | blue |

**One Hot Encoding**

| id | color_red | color_blue | color_green |
|---|---|---|---|
| 1 | 1 | 0 | 0 |
| 2 | 0 | 1 | 0 |
| 3 | 0 | 0 | 1 |
| 4 | 0 | 1 | 0 |

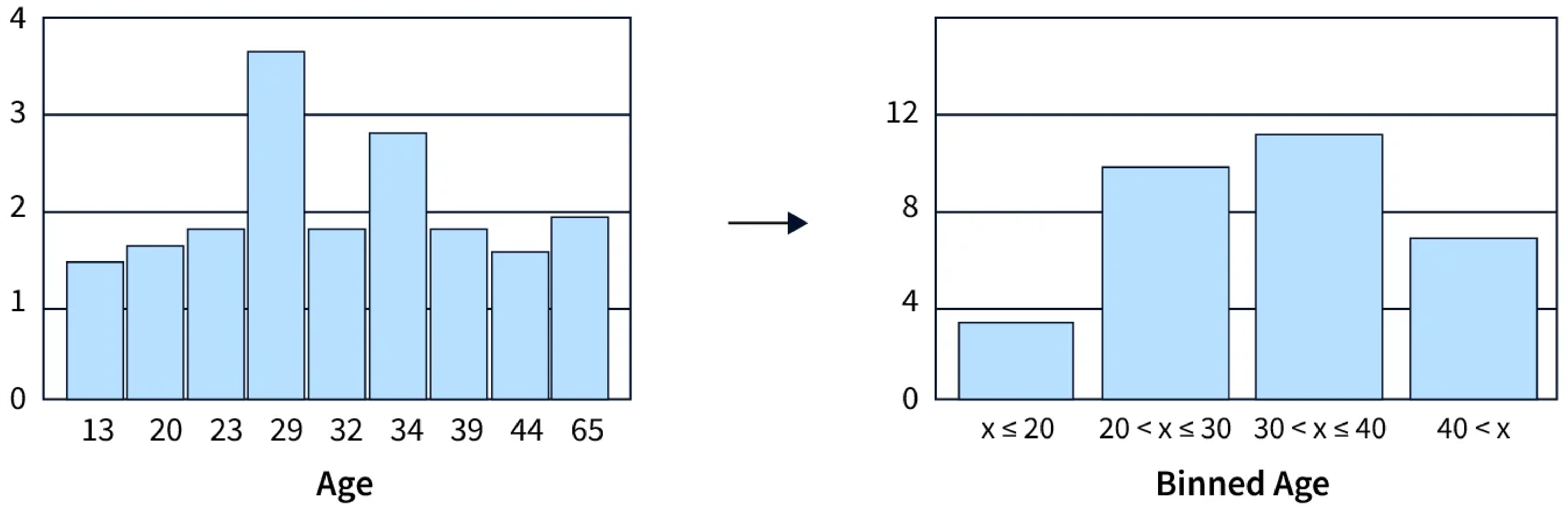
# Binning

- Binning is the conversion of continuous values into categorical ones.
- Prevent overfitting.

| Sex | Age |
|---|---|
| male | 22 |
| female | 38 |
| female | 26 |
| female | 35 |
| male | 35 |
| male | 80 |
| male | 54 |
| male | 2 |
| female | 27 |
| female | 14 |
| female | 4 |
| female | 58 |

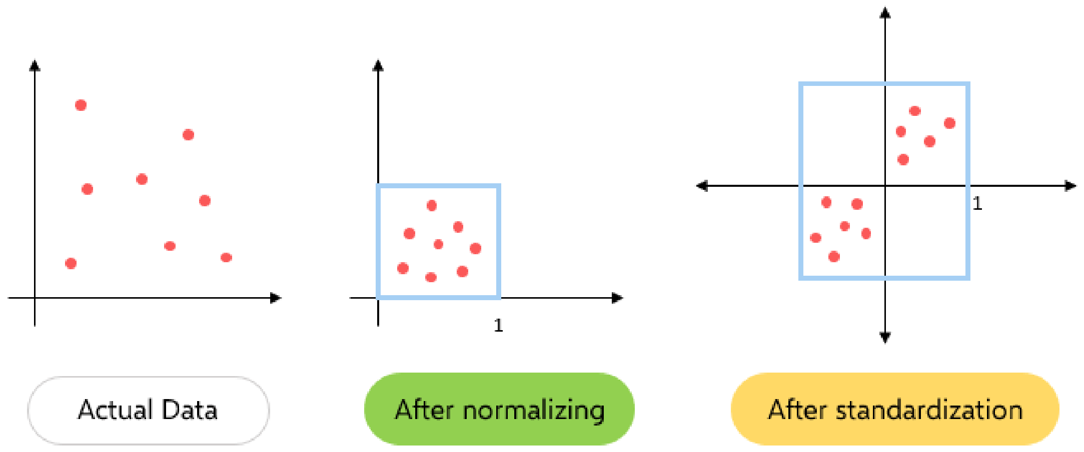
| Sex | Age |
|---|---|
| male | Adult |
| female | Adult |
| female | Adult |
| female | Adult |
| male | Adult |
| male | Elderly |
| male | Adult |
| male | Toddler/baby |
| female | Adult |
| female | Child |
| female | Toddler/baby |
| female | Adult |

# Binning


4
3
2
1
0
13
20
23
29
32
34
39
44
65
Age
12
8
4
0
x ≤ 20
20 < x ≤ 30
30 < x ≤ 40
40 < x
Binned Age

# Feature Scaling

- There are two common ways to get all attributes to have the same scale:
  - Normalization
  - Standardization



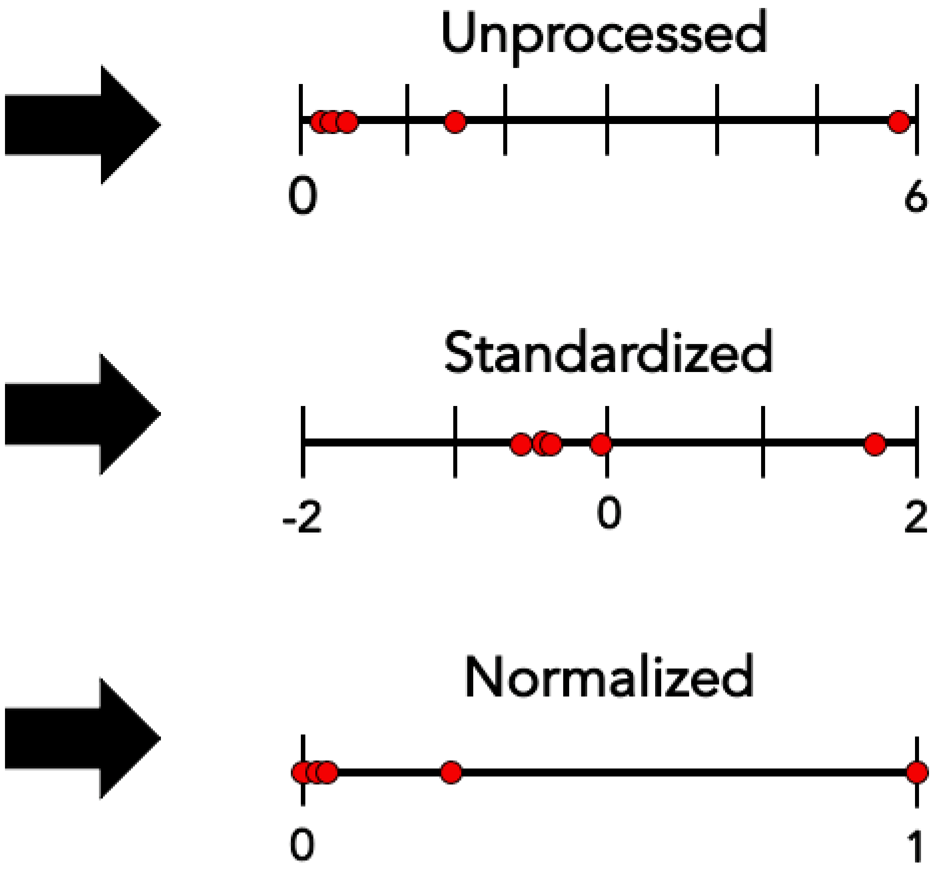
# No Scaling Problem

| *Person_name* | *Salary* | *Years_of_experience* | *Expected_Position_Level* |
|---|---|---|---|
| Ahmed | 100000 | 10 | **2** |
| Khaled | 78000 | 7 | **4** |
| Mohammed | 32000 | 5 | **8** |
| Ali | 55000 | 6 | **7** |
| Yousef | 92000 | 8 | **3** |
| Saleh | 120000 | 15 | **1** |
| Ayman | 65750 | 7 | **5** |

The attributes *Salary* and *Years_of_experience* are on different scale and hence attribute *Salary* can take high priority over attribute *Years_of_experience* in the model.

# Feature Scaling

| Acres |
| --- |
| 0.23 |
| 1.46 |
| 0.29 |
| 0.36 |
| 5.87 |



# Normalization

- Normalization (or min-max normalization) scale all values in a fixed range between **0 and 1**.

$$\bar{x}^{(j)} = \frac{x^{(j)} - min^{(j)}}{max^{(j)} - min^{(j)}}$$

- $min^{(j)}$: Minimum value of the feature $j$
- $max^{(j)}$: Maximum value of the feature $j$

| … | cost | … |
|---|---|---|
| … | 55000 | … |
| … | 70000 | … |
| … | 65000 | … |
| … | 43000 | … |

| … | cost | … |
|---|---|---|
| … | 0.4444 | … |
| … | 1 | … |
| … | 0.8148 | … |
| … | 0 | … |

# Standardization

- **Standardization** (or z-score normalization) is the procedure during which the feature values are rescaled so that they have the properties of a standard normal distribution with $\mu$ = 0 and $\sigma$ = 1.

$$\hat{x}^{(j)} = \frac{x^{(j)} - \mu^{(j)}}{\sigma^{(j)}}$$

$\mu^{(j)}$: Mean value of the feature $j$

$\sigma^{(j)}$: Standard deviation from the mean value of the feature $j$

| ... | cost | ... |
|---|---|---|
| ... | 55000 | ... |
| ... | 70000 | ... |
| ... | 65000 | ... |
| ... | 43000 | ... |

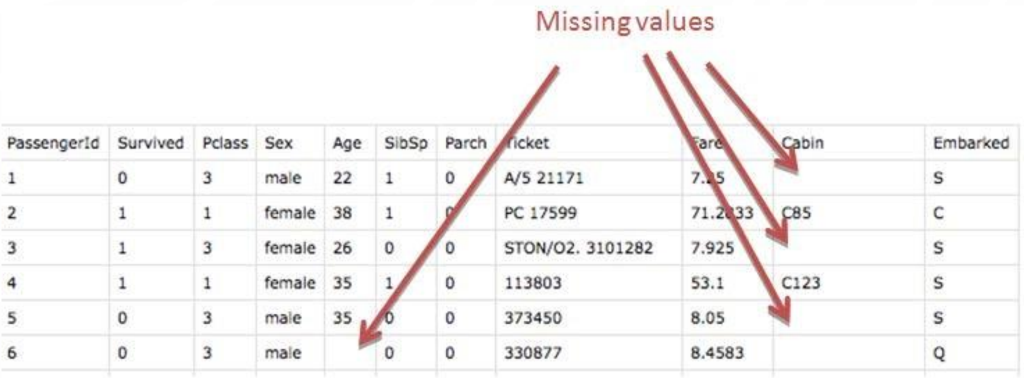
| ... | cost | ... |
|---|---|---|
| ... | -0.314 | ... |
| ... | 1.137 | ... |
| ... | 0.653 | ... |
| ... | -1.476 | ... |

- Standardization is much **less** affected by **outliers**.

# Dealing with Missing Features

- Missing data are values that are not recorded in the dataset, represented by NaN.

- Different ways of dealing with missing features:
    1. Removing the examples with missing data from the dataset.
    2. Using a learning algorithm that can deal with missing feature values.
    3. Using a data imputation technique.




| PassengerId | Survived | Pclass | Sex | Age | SibSp | Parch | Ticket | Fare | Cabin | Embarked |
|---|---|---|---|---|---|---|---|---|---|---|
| 1 | 0 | 3 | male | 22 | 1 | 0 | A/5 21171 | 7.25 | | S |
| 2 | 1 | 1 | female | 38 | 1 | 0 | PC 17599 | 71.2833 | C85 | C |
| 3 | 1 | 3 | female | 26 | 0 | 0 | STON/O2. 3101282 | 7.925 | | S |
| 4 | 1 | 1 | female | 35 | 1 | 0 | 113803 | 53.1 | C123 | S |
| 5 | 0 | 3 | male | 35 | 0 | 0 | 373450 | 8.05 | | S |
| 6 | 0 | 3 | male | | 0 | 0 | 330877 | 8.4583 | | Q |

# Data Imputation Techniques

- Data Imputation Techniques are ways to deal with missing features by filling them with values such as:
  - Mean/Median Values
  - Most Frequent or Zero/Constant Values
  - Predicted value using a regression model

| | col1 | col2 | col3 | col4 | col5 |
|---|---|---|---|---|---|
| **0** | 2 | 5.0 | 3.0 | 6 | NaN |
| **1** | 9 | NaN | 9.0 | 0 | 7.0 |
| **2** | 19 | 17.0 | NaN | 9 | NaN |

mean()

| | col1 | col2 | col3 | col4 | col5 |
|---|---|---|---|---|---|
| **0** | 2.0 | 5.0 | 3.0 | 6.0 | 7.0 |
| **1** | 9.0 | 11.0 | 9.0 | 0.0 | 7.0 |
| **2** | 19.0 | 17.0 | 6.0 | 9.0 | 7.0 |

# Python Example

- https://colab.research.google.com/drive/1YwvH-HLpmm4RDBrqOVX_UHQ66UskHwgS?usp=sharing