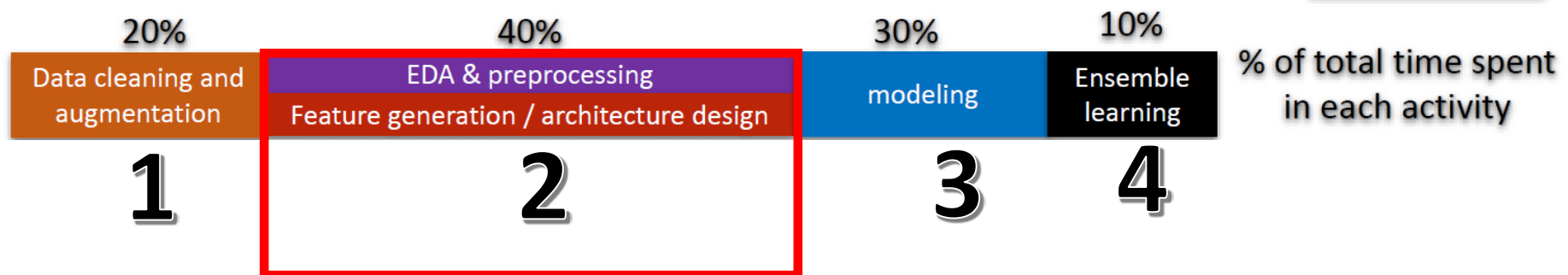
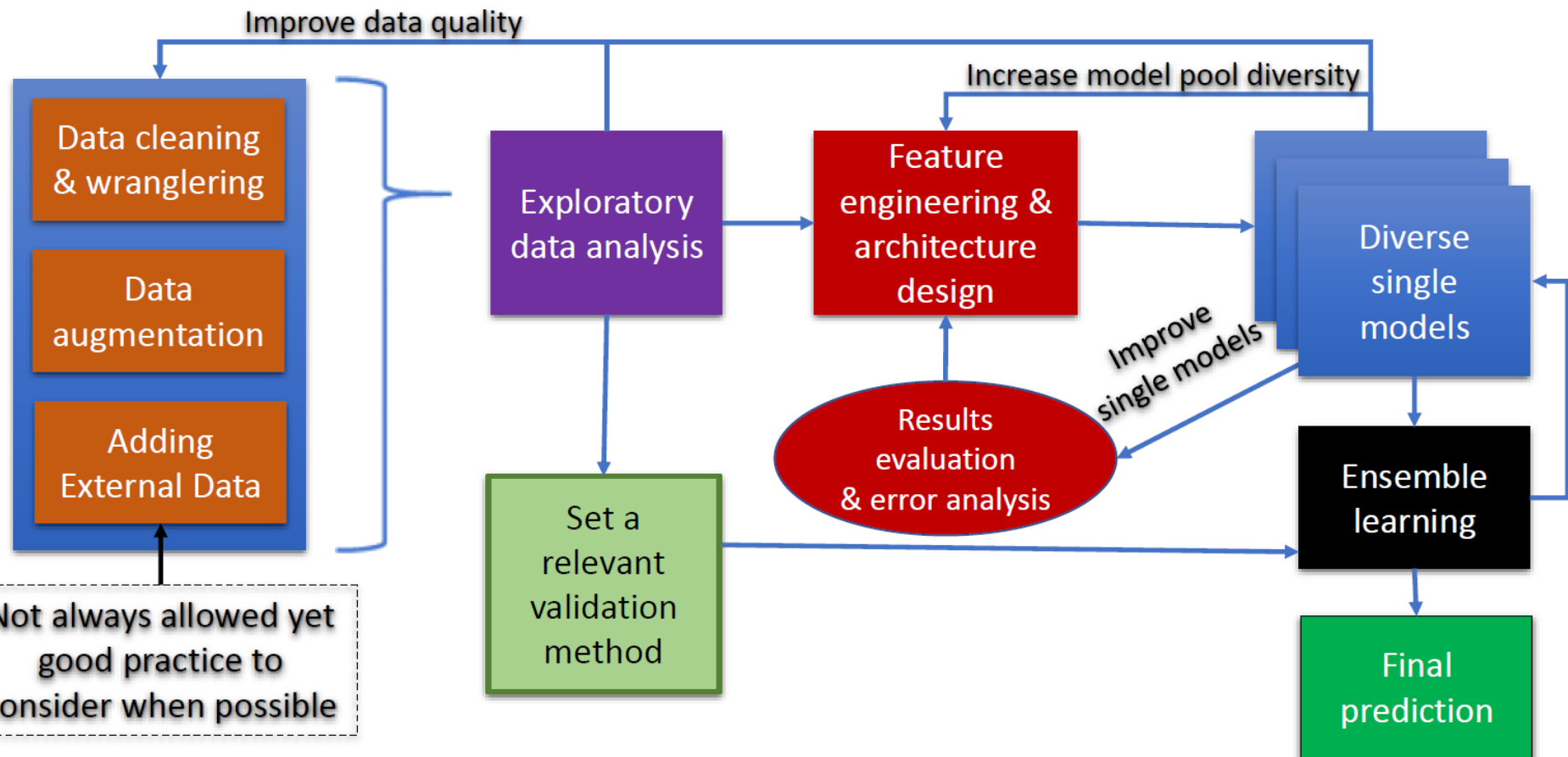


Feature Engineering

Dr Alexander A S Gunawan

aagung@binus.edu
<http://sigmetris.com>
08175001010

Data Science Project Flow



2B. Feature Generation

A. Feature Engineering

- Rescaling/ standardization of existing features
- Performing data transformations: Tf-Idf, log1p, min-max scaling, binning of numeric features
- Turn categorical features to numeric (label encoding / one hot encoding)
- Create count features
- Parsing textual features to get more generalizable features
- Time series: Extracting date/time features i.e month, year, DayOfWeek, dayOfMonth, isHoliday?, isExtreme? etc.

B. Feature Selection

- Remove near-zero-variance features
- Use feature importance and eliminate least important features
- Remove 1-2 most significant features to increase model diversity
- Recursive Feature Elimination

Feature Generation



Squared area: 55 m^2

Price: 107000 \$

Price for 1 m^2 : $107000 \$ / 55 \text{ m}^2$

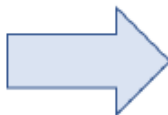
Feature generation is powered by:

- a. Prior knowledge
- b. Exploratory data analysis

Categorical Variables

Encoding categorical features


Index	Country
1	'India'
2	'USA'
3	'UK'
4	'UK'
5	'France'
...	...

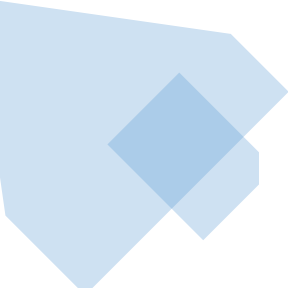


Index	C_India	C_USA	C_UK	C_France
1	1	0	0	0
2	0	1	0	0
3	0	0	1	0
4	0	0	1	0
5	0	0	0	1
...



Encoding categorical features


- One-hot encoding
 - Dummy encoding
-
- **One-hot encoding:** Explainable features
 - **Dummy encoding:** Necessary information without duplication
- 



Index	Sex
0	Male
1	Female
2	Male

Index	Male	Female
0	1	0
1	0	1
2	1	0

Index	Male
0	1
1	0
2	1



Limiting your columns

```
counts = df['Country'].value_counts()  
print(counts)
```

```
'USA'      8  
'UK'       6  
'India'    2  
'France'   1  
Name: Country, dtype: object
```

Limiting your columns

```
mask = df['Country'].isin(counts[counts < 5].index)
df['Country'][mask] = 'Other'
print(pd.value_counts(colors))
```

```
'USA'      8
'UK'       6
'Other'     3
Name: Country, dtype: object
```

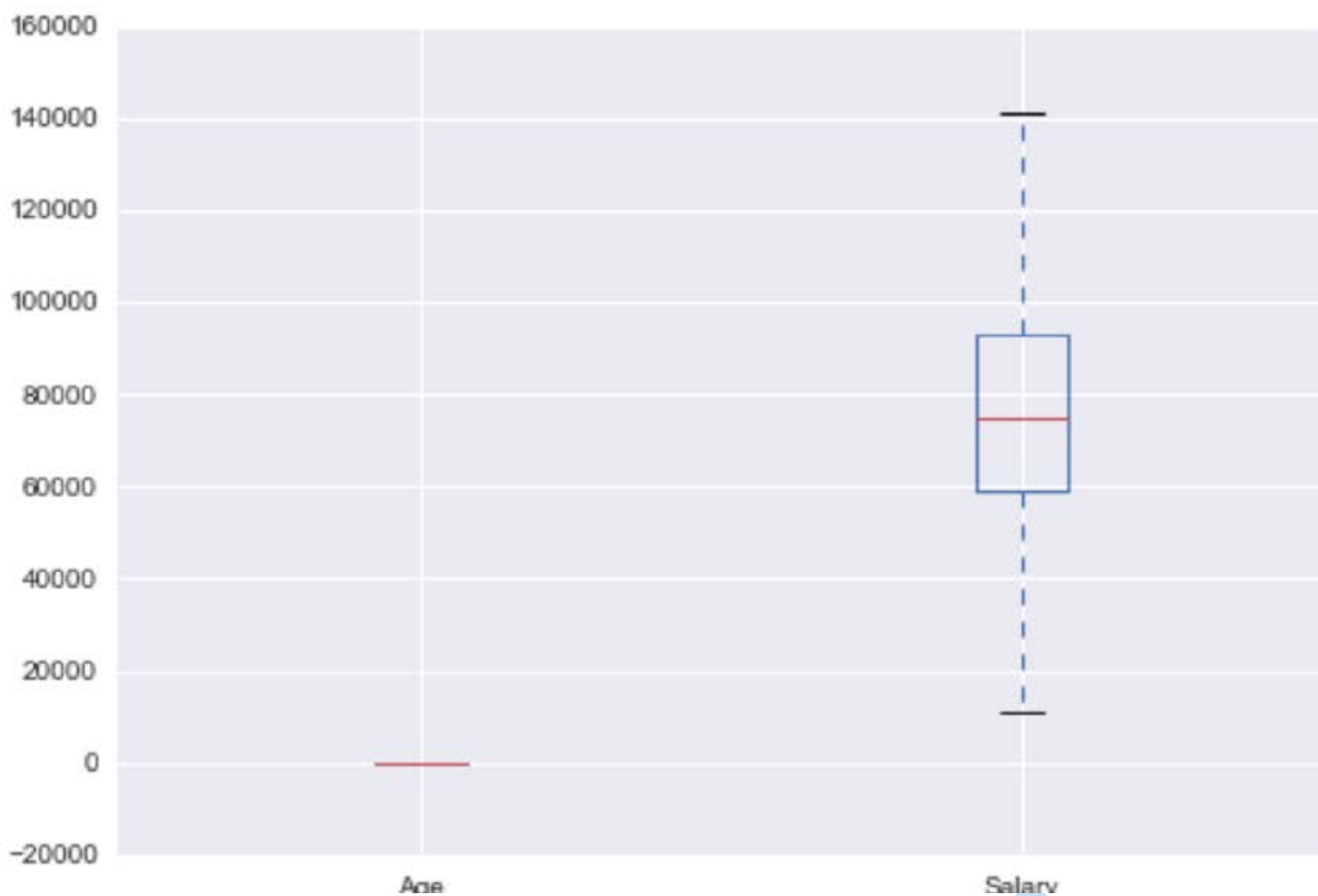
Numeric Variables



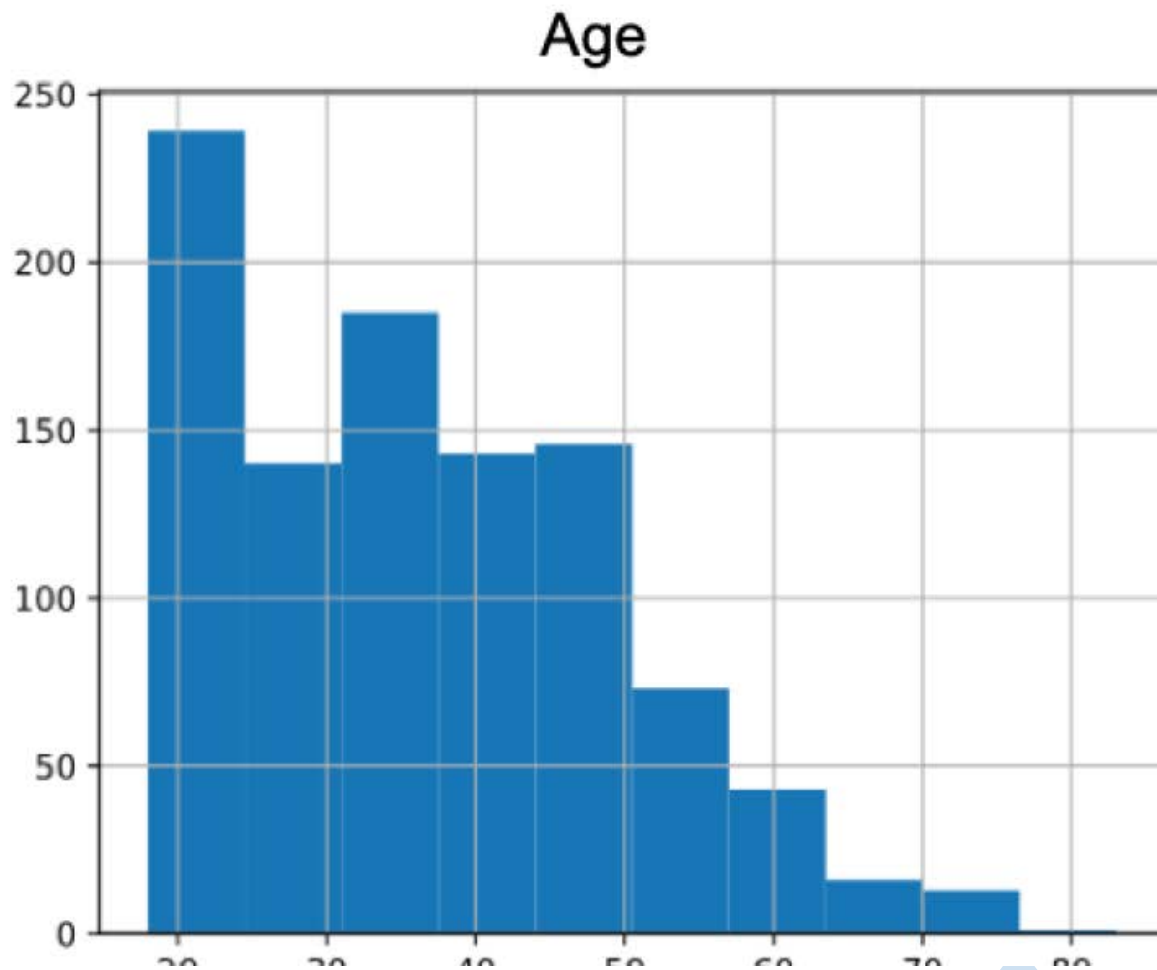
Types of numeric features

- Age
- Price
- Counts

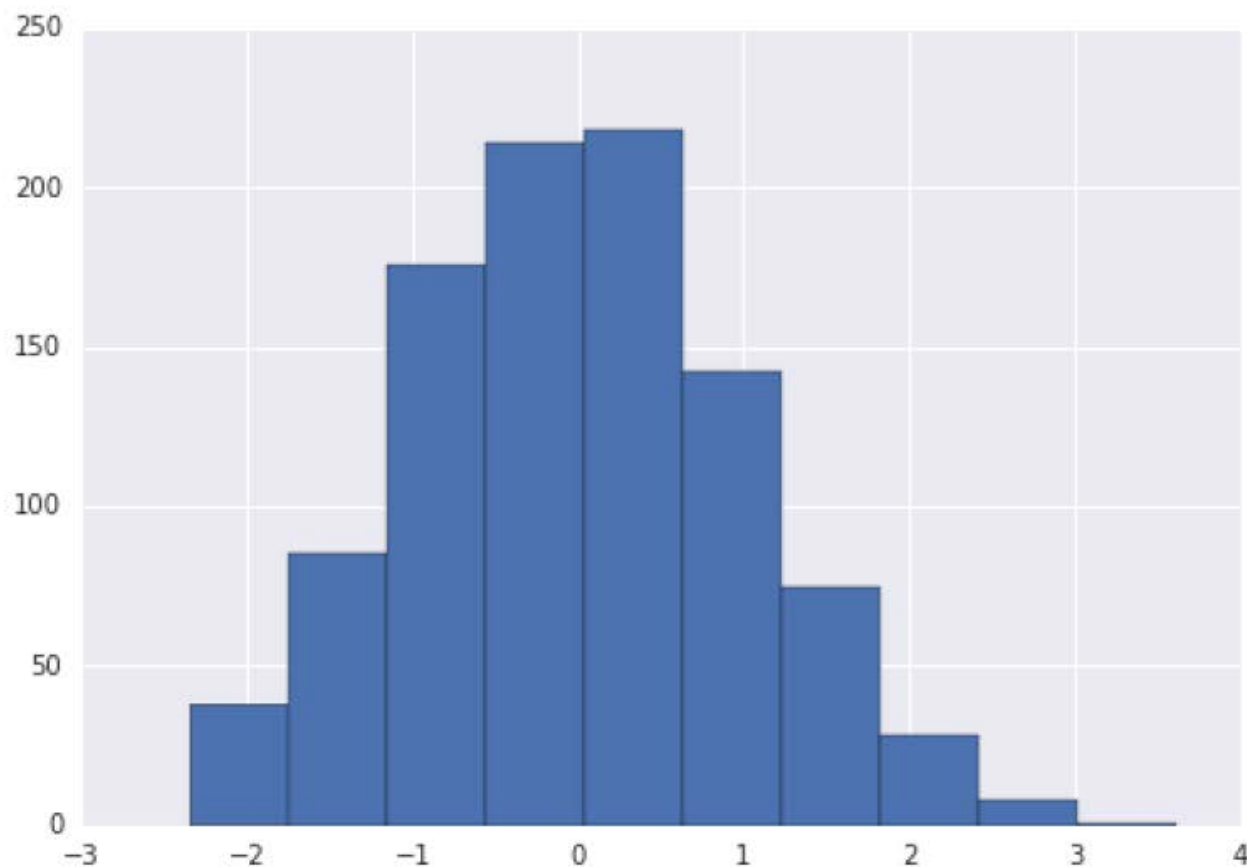
Scaling data



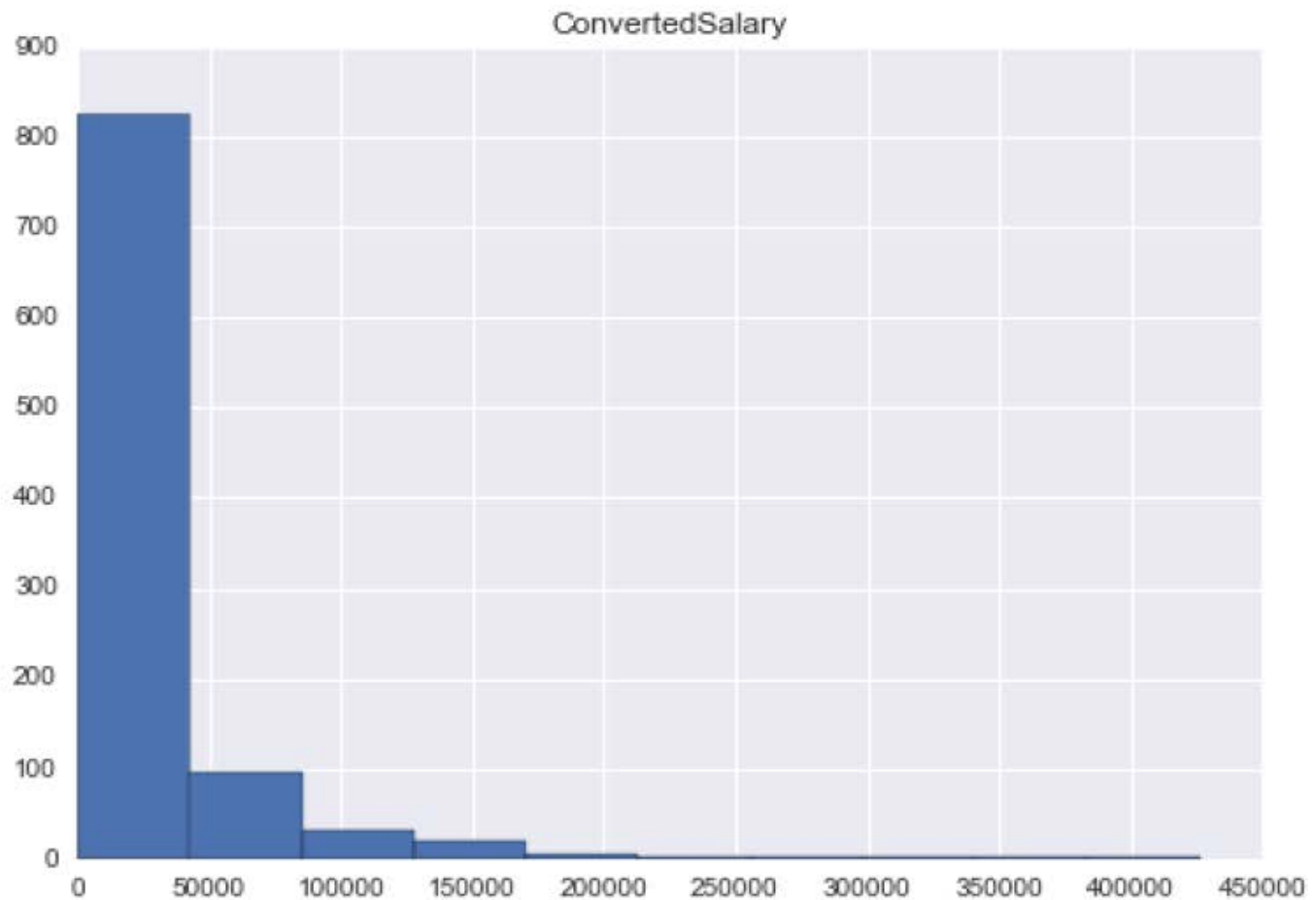
Min-Max scaling



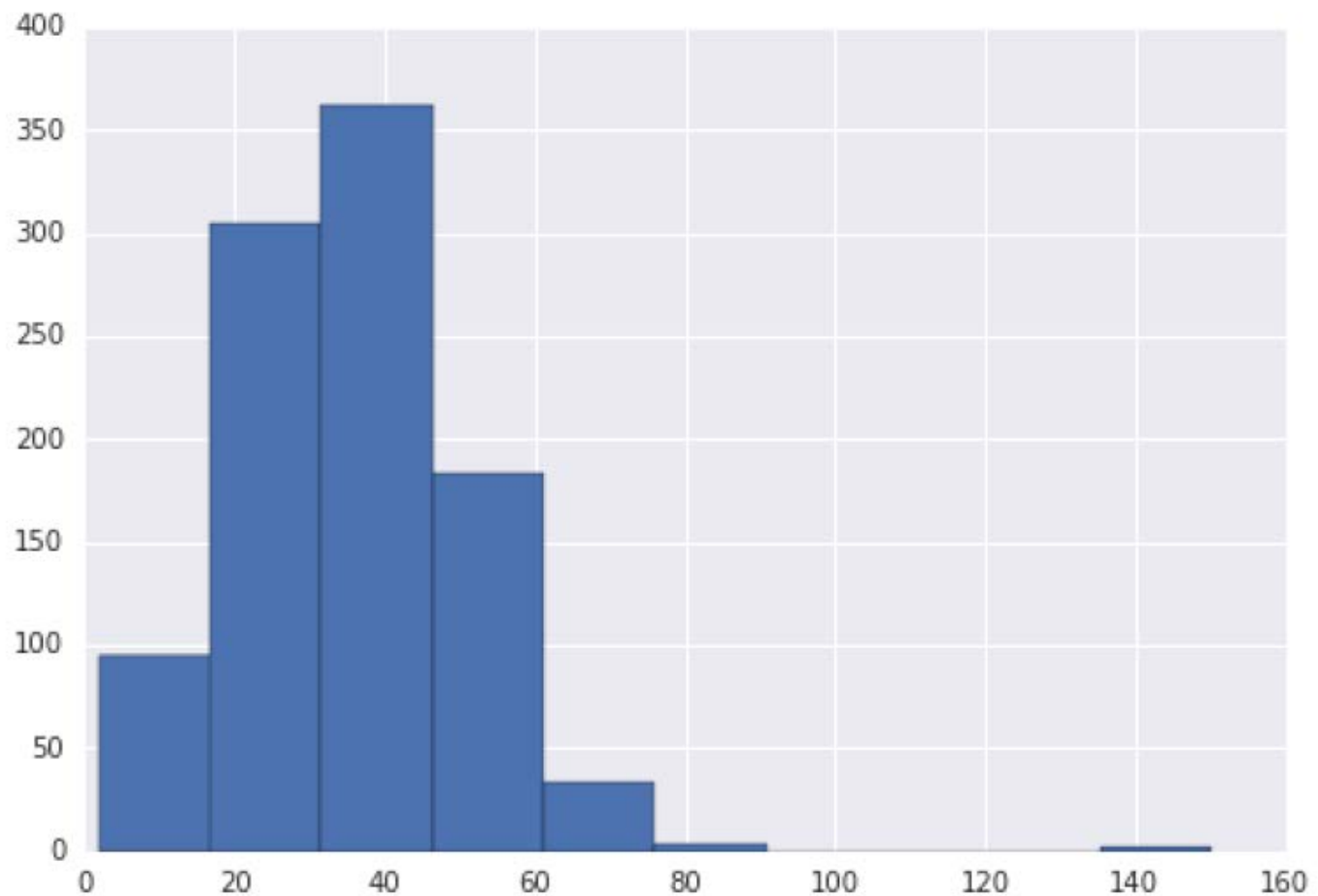
Standardization



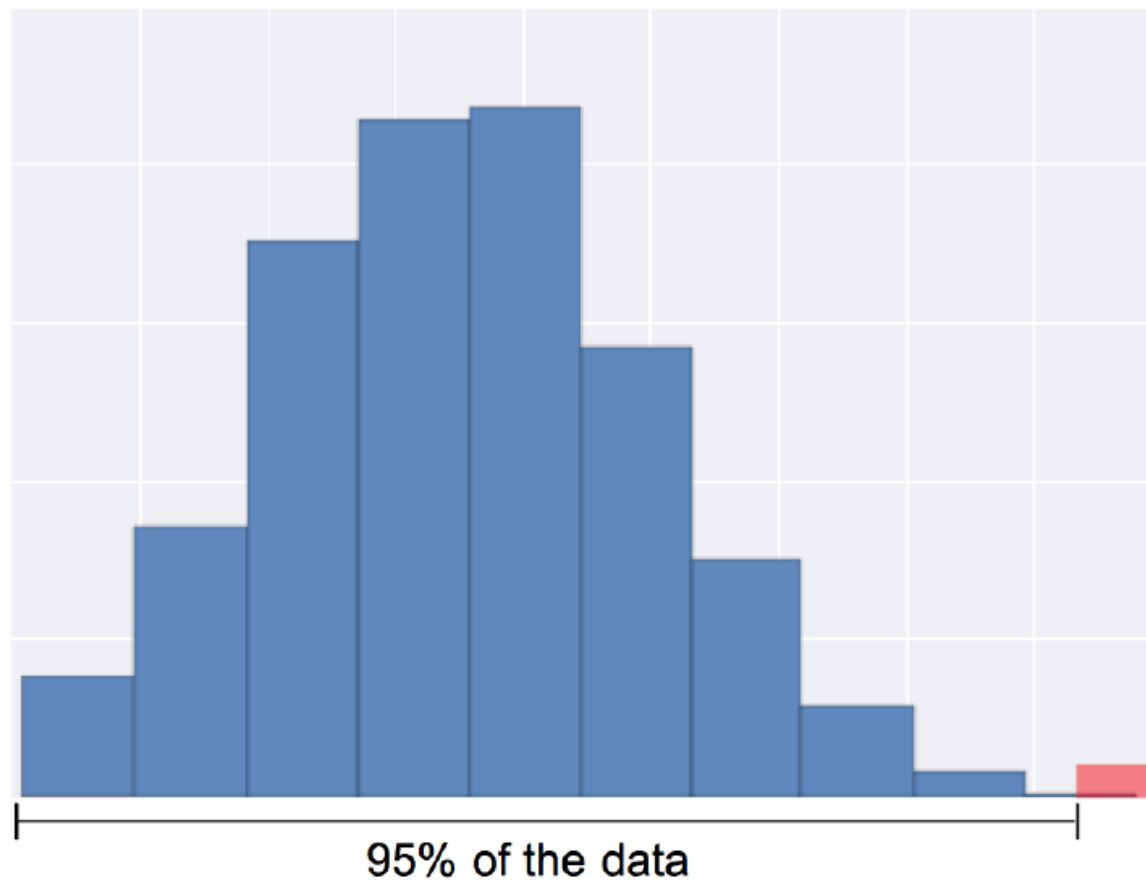
Log Transformation



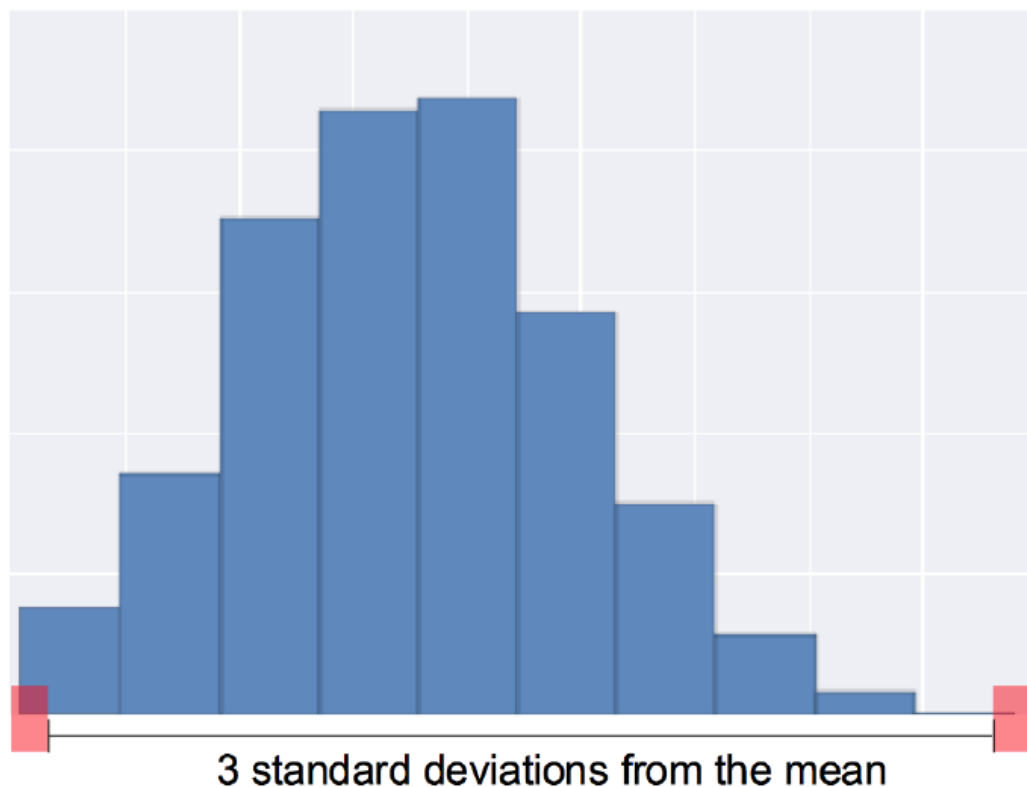
What are outliers?



Quantile based detection



Standard deviation based detection



Datetime and coordinates

Date and time

1. Periodicity

Day number in week, month, season, year
second, minute, hour.

2. Time since

a. Row-independent moment

For example: since 00:00:00 UTC, 1 January 1970;

b. Row-dependent important moment

Number of days left until next holidays/ time passed
after last holiday.

3. Difference between dates

`datetime_feature_1 - datetime_feature_2`

Coordinates

Additional
data



Other train samples
and centers of
clusters



Aggregated
stats

Text

Length of text

```
speech_df['char_cnt'] = speech_df['text'].str.len()  
print(speech_df['char_cnt'].head())
```

```
0    1889
```

```
1     806
```

```
2    2408
```

```
3    1495
```

```
4    2465
```

```
Name: char_cnt, dtype: int64
```


Word counts

```
speech_df['word_cnt'] =  
    speech_df['text'].str.split()  
speech_df['word_cnt'].head(1)
```

```
['fellow', 'citizens', 'of', 'the', 'senate', 'and', ...]
```

Text to columns

“citizens of the senate and of the house of representatives”



Index	citizens	of	the	senate	and	house	representatives
1	1	3	2	1	1	1	1

TF-IDF

$$\text{TF-IDF} = \frac{\frac{\text{Count of word occurrences}}{\text{Total words in document}}}{\log\left(\frac{\text{Number of docs word is in}}{\text{Total number of docs}}\right)}$$