Why generate features?

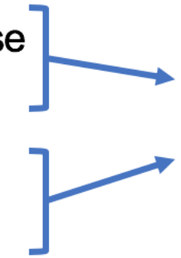
FEATURE ENGINEERING FOR MACHINE LEARNING IN PYTHON

Robert O'Callaghan
Director of Data Science, Ordergroove

Feature Engineering

House A is a **two** bedroomed house **2000** sq. ft brownstone.

House B is **1500** sq. ft with **one** bedroom.



House	Bedrooms	sq. ft
A	2	2000
B	1	1500
...	...	...

Different types of data

- Continuous: either integers (or whole numbers) or floats (decimals)
- Categorical: one of a limited set of values, e.g. gender, country of birth
- Ordinal: ranked values, often with no detail of distance between them
- Boolean: True/False values
- Datetime: dates and times

Course structure

- Chapter 1: Feature creation and extraction
- Chapter 2: Engineering messy data
- Chapter 3: Feature normalization
- Chapter 4: Working with text features

Pandas

```
import pandas as pd
df = pd.read_csv(path_to_csv_file)
print(df.head())
```

Dataset

```
              SurveyDate  \
0    2018-02-28 20:20:00
1    2018-06-28 13:26:00
2    2018-06-06 03:37:00
3    2018-05-09 01:06:00
4    2018-04-12 22:41:00

                              FormalEducation
0    Bachelor's degree (BA. BS. B.Eng.. etc.)
1    Bachelor's degree (BA. BS. B.Eng.. etc.)
2    Bachelor's degree (BA. BS. B.Eng.. etc.)
3    Some college/university study  ...
4    Bachelor's degree (BA. BS. B.Eng.. etc.)
```

Column names

```
print(df.columns)
```

```
Index(['SurveyDate', 'FormalEducation',
       'ConvertedSalary', 'Hobby', 'Country',
       'StackOverflowJobsRecommend', 'VersionControl',
       'Age', 'Years Experience', 'Gender',
       'RawSalary'], dtype='object')
```

Column types

```
print(df.dtypes)
```

```
SurveyDate                object
FormalEducation           object
ConvertedSalary          float64
...
Years Experience           int64
Gender                    object
RawSalary                 object
dtype: object
```

Selecting specific data types

```
only_ints = df.select_dtypes(include=['int'])
print(only_ints.columns)
```

```
Index(['Age', 'Years Experience'], dtype='object')
```

Lets get going!

FEATURE ENGINEERING FOR MACHINE LEARNING IN PYTHON

Dealing with Categorical Variables

FEATURE ENGINEERING FOR MACHINE LEARNING IN PYTHON

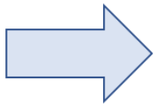
Robert O'Callaghan
Director of Data Science, Ordergroove

Encoding categorical features

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

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

```
pd.get_dummies(df, columns=['Country'],
               prefix='C')
```

```
   C_France  C_India  C_UK  C_USA
0         0        1     0      0
1         0        0     0      1
2         0        0     1      0
3         0        0     1      0
4         1        0     0      0
```

Dummy encoding

```
pd.get_dummies(df, columns=['Country'],
               drop_first=True, prefix='C')
```

```
   C_India  C_UK  C_USA
0        1     0      0
1        0     0      1
2        0     1      0
3        0     1      0
4        0     0      0
```

One-hot vs. dummies

- **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
```

Now you deal with categorical variables

FEATURE ENGINEERING FOR MACHINE LEARNING IN PYTHON

Numeric variables

FEATURE ENGINEERING FOR MACHINE LEARNING IN PYTHON

Robert O'Callaghan
Director of Data Science, Ordergroove

Types of numeric features

- Age
- Price
- Counts
- Geospatial data

Does size matter?

	Resturant_ID	Number_of_Violations
0	RS_1	0
1	RS_2	0
2	RS_3	2
3	RS_4	1
4	RS_5	0
5	RS_6	0
6	RS_7	4
7	RS_8	4
8	RS_9	1
9	RS_10	0

Binarizing numeric variables

```
df['Binary_Violation'] = 0

df.loc[df['Number_of_Violations'] > 0,
       'Binary_Violation'] = 1
```

Binarizing numeric variables

	Resturant_ID	Number_of_Violations	Binary_Violation
0	RS_1	0	0
1	RS_2	0	0
2	RS_3	2	1
3	RS_4	1	1
4	RS_5	0	0
5	RS_6	0	0
6	RS_7	4	1
7	RS_8	4	1
8	RS_9	1	1
9	RS_10	0	0

Binning numeric variables

```
import numpy as np
df['Binned_Group'] = pd.cut(
    df['Number_of_Violations'],
    bins=[-np.inf, 0, 2, np.inf],
    labels=[1, 2, 3]
)
```

Binning numeric variables

	Resturant_ID	Number_of_Violations	Binned_Group
0	RS_1	0	1
1	RS_2	0	1
2	RS_3	2	2
3	RS_4	1	2
4	RS_5	0	1
5	RS_6	0	1
6	RS_7	4	3
7	RS_8	4	3
8	RS_9	1	2
9	RS_10	0	1

Lets start practicing!

FEATURE ENGINEERING FOR MACHINE LEARNING IN PYTHON