


Machine Learning : 06048203

**Feature Engineering**



# Feature Engineering and Data Preparation

A 3D rendering of a warehouse conveyor belt system. Several cardboard boxes are positioned on the belt, which is flanked by blue metal guides. Red laser lines project a grid pattern onto the floor and the boxes, suggesting a precision tracking or sorting system. The perspective is from a low angle, looking down the length of the conveyor.



# Feature Scaling

X1		X2		X3		X4	
\$	179.43		56.784		34.6181		3.55
\$	641.87		62.054		47.7306		1.692
\$	556.30		64.13		55.596		1.559
\$	578.47		63.377		52.7121		1.679
\$	591.16		61.553		46.1315		1.984
\$	242.03		58.29		39.2952		2.942
\$	364.66		59.93		42.4628		2.494
\$	190.68		57.271		36.2725		3.419
\$	547.23		63.763		54.1971		1.634
\$	359.69		59.375		41.5105		2.128
\$	438.08		60.484		43.493		2.47
\$	637.17		62.525		49.428		1.725

# Feature Scaling

**Min-Max scaling Normalization**

$$X' = \frac{X - X_{min}}{X_{max} - X_{min}}$$

$[0 ; 1]$

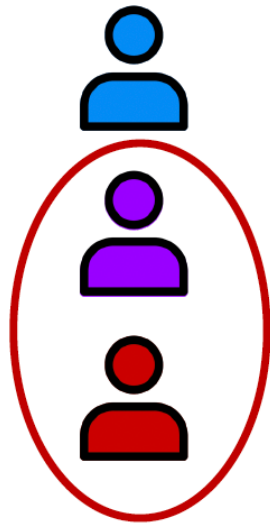
**Standard Scaler Normalization**

$$X' = \frac{X - \mu}{\sigma}$$





# Feature Scaling



70,000 \$

10,000

60,000 \$

8,000

52,000 \$

45 yrs

1

44 yrs

4

40 yrs

# Feature Scaling

Min-Max scaling Normalization

$$X' = \frac{X - X_{min}}{X_{max} - X_{min}}$$

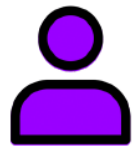
[0 ; 1]

# Feature Scaling



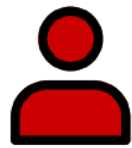
70,000 \$

45 yrs



60,000 \$

44 yrs

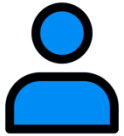


52,000 \$

40 yrs

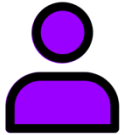


# Feature Scaling



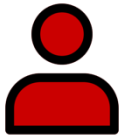
1

45 yrs



0.444

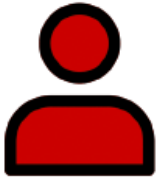
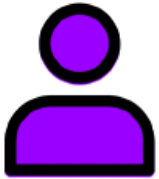
44 yrs



0

40 yrs

# Feature Scaling



1

0.444

0



1

0.75

0

# Min-Max scaling Normalization

```
from sklearn.preprocessing import MinMaxScaler
import pandas as pd
data = [[-1, 2], [-0.5, 6], [0, 10], [1, 18]]
scaler = MinMaxScaler()
```

```
xdata = pd.DataFrame(data, columns=['x1', 'x2'])
xdata
```

✓ 0.0s

	x1	x2
0	-1.0	2
1	-0.5	6
2	0.0	10
3	1.0	18

```
xscale = scaler.fit_transform(xdata)
xscale = pd.DataFrame(xscale, columns=['x1', 'x2'])
xscale
```

✓ 0.0s

	x1	x2
0	0.00	0.00
1	0.25	0.25
2	0.50	0.50
3	1.00	1.00

```
x = scaler.inverse_transform(xscale)
x = pd.DataFrame(x, columns=['x1', 'x2'])
x
```

✓ 0.0s

	x1	x2
0	-1.0	2.0
1	-0.5	6.0
2	0.0	10.0
3	1.0	18.0

$$X' = \frac{X - X_{min}}{X_{max} - X_{min}}$$

$$X = X'(X_{max} - X_{min}) + X_{min}$$

# Standard Scaler Normalization

```
from sklearn.preprocessing import StandardScaler

import pandas as pd
data = [[-1, 2], [-0.5, 6], [0, 10], [1, 18]]
scaler = StandardScaler()
```

```
xdata = pd.DataFrame(data, columns=['x1', 'x2'])
xdata
```

✓ 0.0s

	x1	x2
0	-1.0	2
1	-0.5	6
2	0.0	10
3	1.0	18

```
xscale = scaler.fit_transform(xdata)
xscale = pd.DataFrame(xscale, columns=['x1', 'x2'])
xscale
```

✓ 0.0s

	x1	x2
0	-1.183216	-1.183216
1	-0.507093	-0.507093
2	0.169031	0.169031
3	1.521278	1.521278

```
x = scaler.inverse_transform(xscale)
x = pd.DataFrame(x, columns=['x1', 'x2'])
x
```

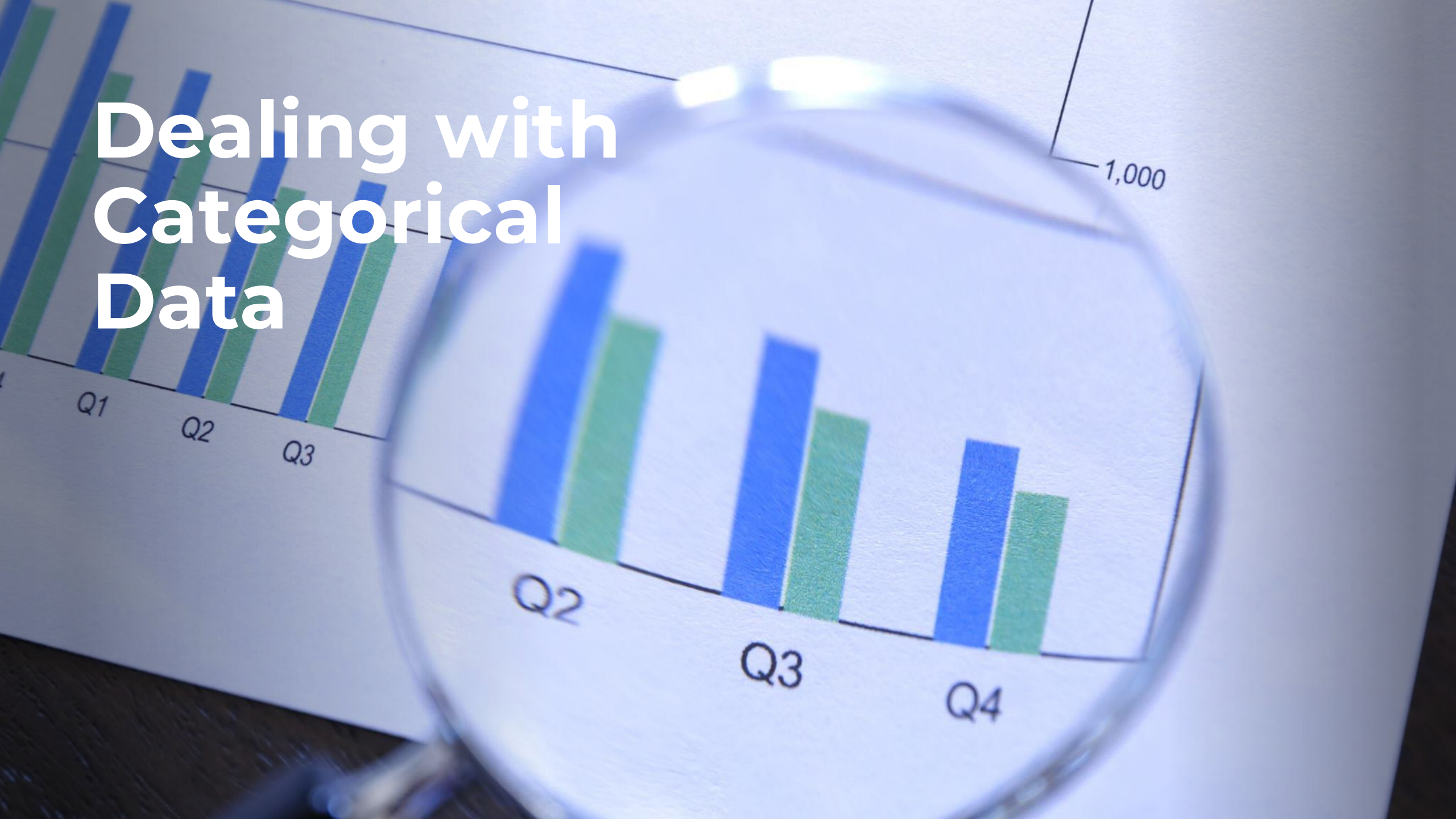
✓ 0.0s

	x1	x2
0	-1.0	2.0
1	-0.5	6.0
2	0.0	10.0
3	1.0	18.0

$$X' = \frac{X - \mu}{\sigma}$$

$$X = \sigma X' + \mu$$

# Dealing with Categorical Data



# Feature Engineering

- Integer Encoding
  - Directly convert categories into integers 1,2,3...N

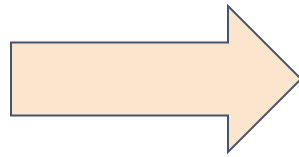
Country
USA
MEX
CAN
USA



# Feature Engineering

- Integer Encoding
  - Possible issue is implied ordering and relationship (ordinal variable)

Country
USA
MEX
CAN
USA



Country
1
2
3
1

# Feature Engineering

- Integer Encoding
  - Pros:
    - Very easy to do and understand.
    - Does not increase number of features.
  - Cons:
    - Implies ordered relationship between categories.

# Feature Engineering

- One Hot Encoding (Dummy Variables)
  - Convert categories into individual features that are either 0 or 1

Country
USA
MEX
CAN
USA

# Feature Engineering

- One Hot Encoding (Dummy Variables)
  - Convert categories into individual features that are either 0 or 1

Country
USA
MEX
CAN
USA



USA	MEX	CAN
1	0	0
0	1	0
0	0	1
1	0	0

# Feature Engineering

- One Hot Encoding (Dummy Variables)
  - No ordered relationship is implied between categories.

Country
USA
MEX
CAN
USA



USA	MEX	CAN
1	0	0
0	1	0
0	0	1
1	0	0

# Feature Engineering

- One Hot Encoding (Dummy Variables)
  - We can try to reduce this feature column expansion by creating higher level categories.
  - For example, regions or continents instead of countries.



# Feature Engineering

- One Hot Encoding (Dummy Variables)
  - Consider a binary category (only two options):

Vertical Direction
UP
DOWN
UP
DOWN

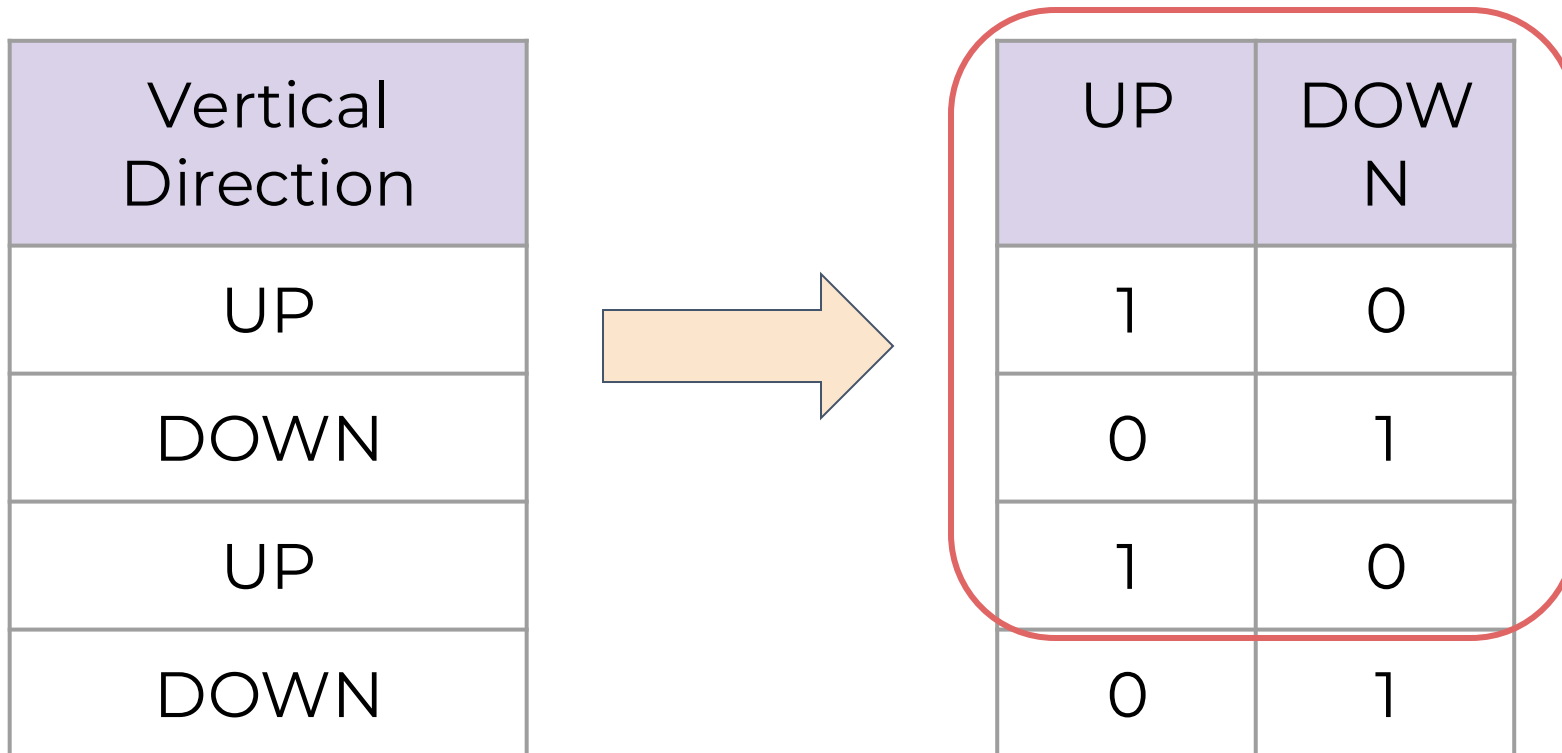
# Feature Engineering

- One Hot Encoding (Dummy Variables)
  - Consider a binary category (only two options):

Vertical Direction		UP	DOW N
UP	→	1	0
DOWN		0	1
UP		1	0
DOWN		0	1

# Feature Engineering

- One Hot Encoding (Dummy Variables)
  - The new columns are duplicate information with inverted encoding.



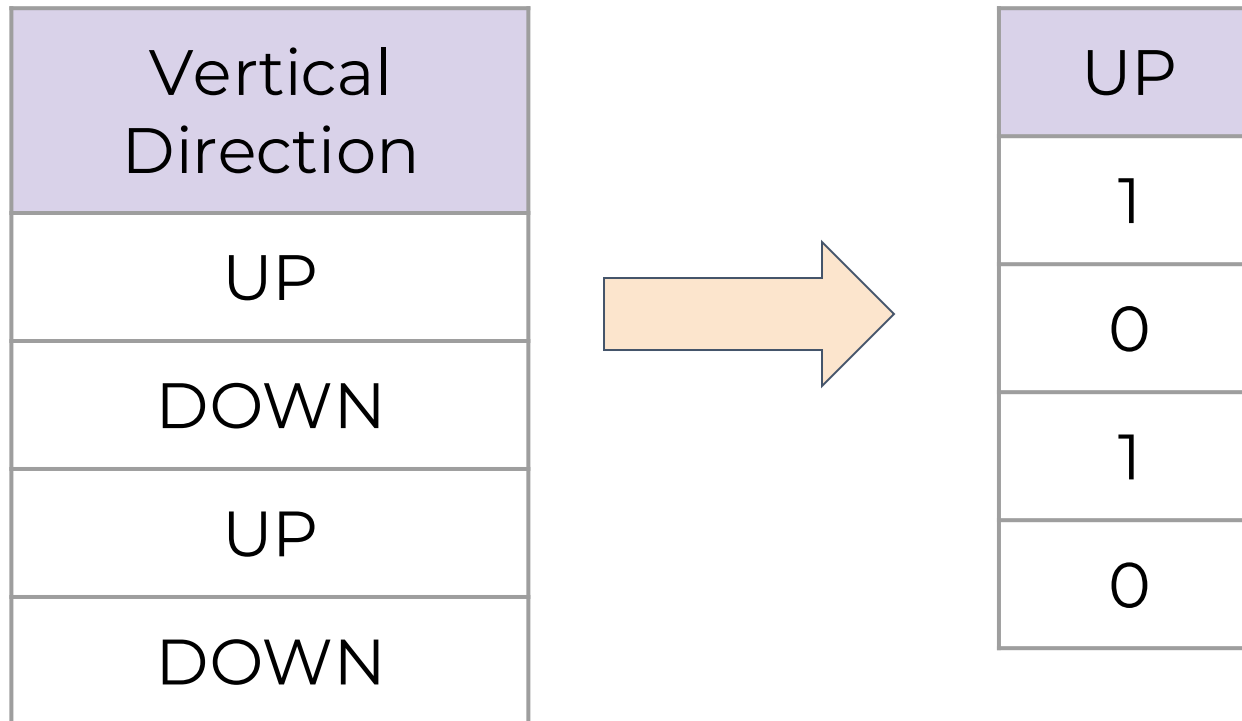
The diagram illustrates the process of one-hot encoding. On the left, a table with a single column 'Vertical Direction' contains four rows of categorical data: 'UP', 'DOWN', 'UP', and 'DOWN'. An orange arrow points to the right, where a new table is shown. This new table has two columns, 'UP' and 'DOWN', which are highlighted by a red rounded rectangle. Each row in the new table contains binary values (0 or 1) that represent the presence of the original category. For the first 'UP' row, 'UP' is 1 and 'DOWN' is 0. For the first 'DOWN' row, 'UP' is 0 and 'DOWN' is 1. This pattern repeats for the second 'UP' and 'DOWN' rows.

Vertical Direction
UP
DOWN
UP
DOWN

UP	DOWN
1	0
0	1
1	0
0	1

# Feature Engineering

- One Hot Encoding (Dummy Variables)
  - Easily fixed by simply dropping last column.



# Feature Engineering

- One Hot Encoding (Dummy Variables)
  - This can be extended to more than 2 categories:

Country
USA
MEX
CAN
USA



USA	MEX
1	0
0	1
0	0
1	0

# Feature Engineering

- One Hot Encoding (Dummy Variables)
  - Pros:
    - No ordering implied.
  - Cons:
    - Potential to create many more feature columns and coefficients.
    - Dummy variable trap consideration.
    - Not easy to add new categories.



The background features a series of flowing, wavy blue shapes that resemble liquid or smoke, creating a dynamic and modern aesthetic. The colors range from deep blue to lighter, almost white, tones, with soft gradients and highlights that give the waves a three-dimensional feel.

# Dealing with Outliers

# Outliers

- Often a data set will have a few points that are extreme outliers.
- It's often better to simply remove these few points from the data set in order to have a more generalized model.

