One Hot Encoding

One Hot encoding is a process by which categorical variables are converted into a form that could be provided to ML algorithms to do a better job in prediction.

Before getting into One hot encoding, let's try to understand one more encoding called "**Label Encoding/Integer Encoding**".

Label Encoding refers to converting the labels into numeric form.

Consider the below example:

You are working with a dataset, having a categorical column containing the below values:

red
green
blue

When you go with Label Encoding, it assigns a numerical value to each categorical data. The above will be encoded as

Label	Label Encoding
red	0
green	1
blue	2

Disadvantages of Label Encoding:

For Machine Learning, this encoding can be problematic - in this example, we're essentially saying "green" is the *average* of "red" and "blue", which can lead to weird unexpected outcomes. Also the algorithm will more weightage to 'blue'.

To solve this, it's recommended to go with One Hot Encoding.

Label	Label Encoding	One Hot Encoding
red	0	[1, 0, 0]
green	1	[0, 1, 0]
blue	2	[0, 0, 1]

One Hot Encoding in Python

There are several ways to implement one-hot encoding in Python.

1. Using scikit-learn's OneHotEncoder

```
from sklearn.preprocessing import OneHotEncoder
import pandas as pd

# ignore tells the encoder to ignore new categories by encoding them with 0's
encoder = OneHotEncoder(handle_unknown = 'ignore')

one = encoder.fit_transform([['red'], ['green'], ['blue']])

one = pd.DataFrame(s, columns = encoder.get_feature_names())

# display the values
one
```

Output:

	x0_blue	x0_green	x0_red
0	0.0	0.0	1.0
1	0.0	1.0	0.0
2	1.0	0.0	0.0

The One-Hot Encoded value will then be merged with other numerical data columns

```
model_data = pd.concat([model_data[['Actual Value', 'Gross', 'Net']], one[:]], axis=1)
```

For the new data, it converts the value as below:

```
new = pd.DataFrame(encoder.transform([['green']]).toarray(), columns =
encoder.get_feature_names())

new              # display the values
```

Output:

	x0_blue	x0_green	x0_red
0	0.0	1.0	0.0

It's recommended to use "One-Hot Encoder", when you are planning to productionize your ML model.

2. Using Kera's to_categorical:

```
from keras.utils import to_categorical

print(to_categorical([0, 1, 2]))
```

Output:

```
[[1. 0. 0.]
 [0. 1. 0.]
 [0. 0. 1.]]
```

3. Using Numpy:

```
import numpy as np

arr = [2, 1, 0]
max = np.max(arr) + 1
print(np.eye(max)[arr])
```

Output:

```
[[0. 0. 1.]
 [0. 1. 0.]
 [1. 0. 0.]]
```

4. Using get_dummies:

```
model_data = pd.get_dummies(['red', 'green', 'blue'])
model_data
```

Output:

	blue	green	red
0	0	0	1
1	0	1	0
2	1	0	0