

### Why do we Encode?

- Most of the models only accept numeric values.
- We cannot afford to lose important features because of their data types.
- It is required to ensure correct and good performance of the model.

### Types of Encoding

- Ordinal Encoding
- One Hot Encoding
- Label Encoding

### Ordinal Encoding

- Used for encoding Ordinal Variables.
- Numbers are assigned to each category based on their order hierarchy of the variable.
- Assigned numbers can be any numbers as long as original order is unchanged.

Code:

```
!pip install category_encoders
import category_encoders as ce
encoder = ce.OrdinalEncoder(mapping=[-
{col: 'feedback', 'mapping': {'bad': 1, 'okay':
2, 'good':3}}])
encoder.fit(X)
X = encoder.transform(X)
X['feedback']
Output:
feedback
1
2
3
2
3
.
```

Documentation: [https://contrib.scikit-learn.org/category\\_encoders/ordinal.html](https://contrib.scikit-learn.org/category_encoders/ordinal.html)

### One-Hot Encoding

- Used when number of categories in the variable are low, max 3 or 4. Any more will seriously increase the size of your dataset and decrease performance of your model.
  - Assigns 0 and 1 to the categories based on their presence in the columns.
  - Creates extra columns based on the number of categorical elements in the main column.
- i.e if there are 3 categories in the column Shipping - Standard, One Day, Two Day, 3 extra columns are created in place of the original column, 1 for each category and 1 will be assigned for each unique value.

Usage:

```
import category_encoders as ce
encoder = ce.OneHotEncoder(cols=['-
Column Name'])
encoder.fit(df)
df = encoder.transform(df)
df['Shipping']
```

Documentation: [https://contrib.scikit-learn.org/category\\_encoders/onehot.html](https://contrib.scikit-learn.org/category_encoders/onehot.html)

### Output

	Shipping_1	Shipping_2	Shipping_3
	0	0	1
	0	1	0
	1	0	1
	0	1	0
	1	0	0
	.	.	.
	.	.	.

### Label Encoding

- Converts each category in a column to a number directly.
- Can also be used for non-numerical values as long as they are relevant and usable to the target variable.
- Different Methods can be applied according to your requirements.

```
from sklearn.preprocessing import LabelEncoder
le = LabelEncoder()
df['Column Name_Cat'] = le.fit_transform(df-
['Column Name'])
df
```

Documentation: <https://scikit-learn.org/stable/modules/generated/sklearn.preprocessing.LabelEncoder.html>

### Output

Column_1	Column_Name(Original)	Column_Name_Cat	...
.	A	1	.
.	B	2	.
.	C	4	.
.	D	3	.
.	.	.	.
.	.	.	.



By [deleted]  
[cheatography.com/deleted-141090/](https://cheatography.com/deleted-141090/)

Not published yet.  
 Last updated 10th December, 2021.  
 Page 1 of 1.

Sponsored by [ApolloPad.com](https://apollopod.com)  
 Everyone has a novel in them. Finish Yours!  
<https://apollopod.com>