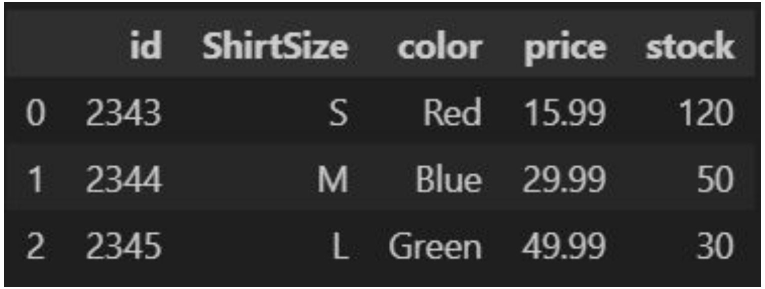
Kaggle Seasons #03

QUEEN MARY MACHINE
LEARNING SOCIETY

Encoding - Categorical

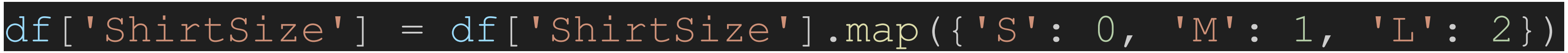
- What is Categorical Data?
  - Ordinal (e.g. Shirt sizes: Small < Medium < Large)
  - Nominal (e.g., Gender: Male/Female)
- How to Encode it?
  - Label Encoding (Mapping)
  - One-Hot-Encoding

Encode Categorical (Ordinal)

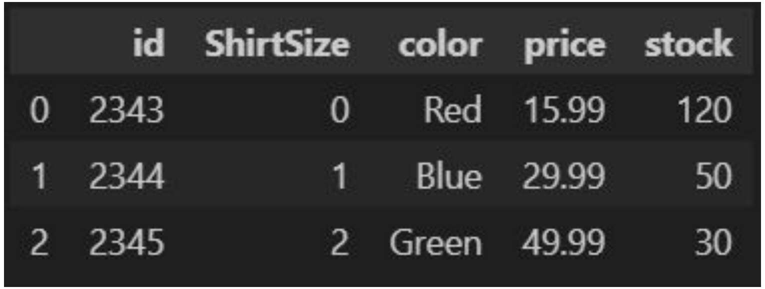


	id	ShirtSize	color	price	stock
0	2343	S	Red	15.99	120
1	2344	M	Blue	29.99	50
2	2345	L	Green	49.99	30

Encode ordinal column ‘ShirtSize’ using *Label Encoding*:

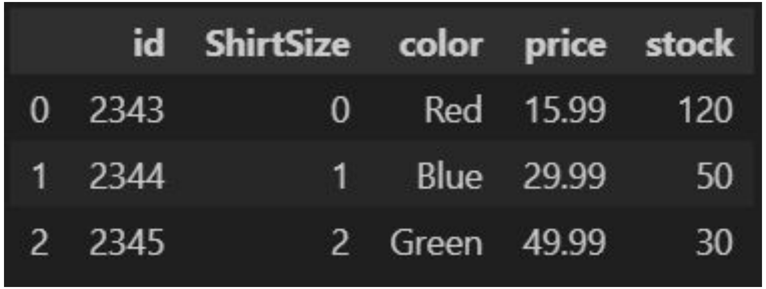


```
df['ShirtSize'] = df['ShirtSize'].map({'S': 0, 'M': 1, 'L': 2})
```



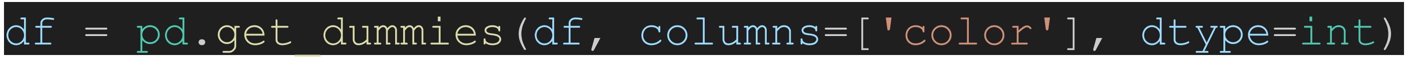
	id	ShirtSize	color	price	stock
0	2343	0	Red	15.99	120
1	2344	1	Blue	29.99	50
2	2345	2	Green	49.99	30

Encode Categorical (Nominal)

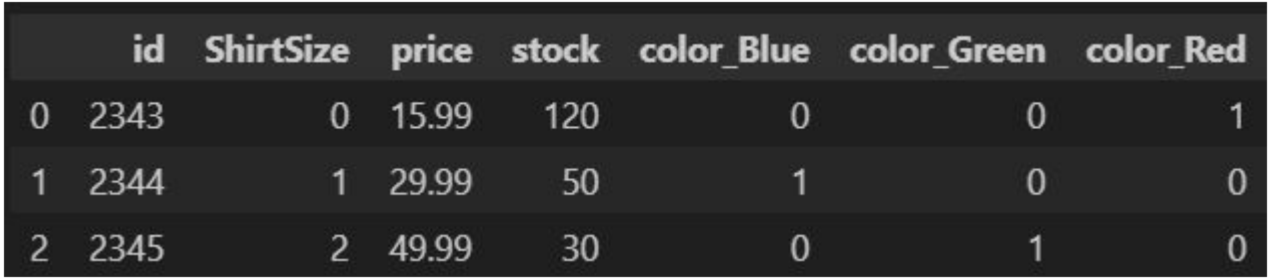


	id	ShirtSize	color	price	stock
0	2343	0	Red	15.99	120
1	2344	1	Blue	29.99	50
2	2345	2	Green	49.99	30

Encode nominal column ‘color’ using *One-Hot-Encoding*:



```
df = pd.get_dummies(df, columns=['color'], dtype=int)
```

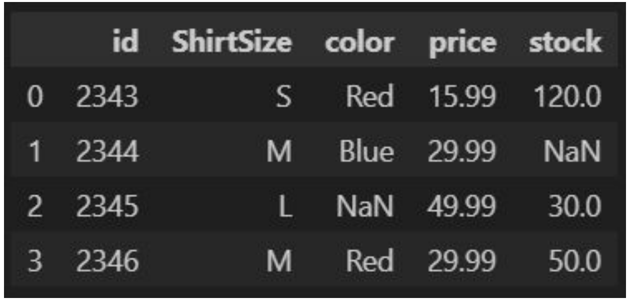


	id	ShirtSize	price	stock	color_Blue	color_Green	color_Red
0	2343	0	15.99	120	0	0	1
1	2344	1	29.99	50	1	0	0
2	2345	2	49.99	30	0	1	0

NaN Values

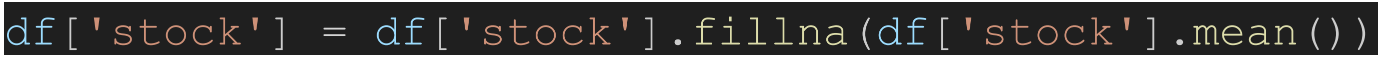
kaggle

Fill NaN Values (Numerical)

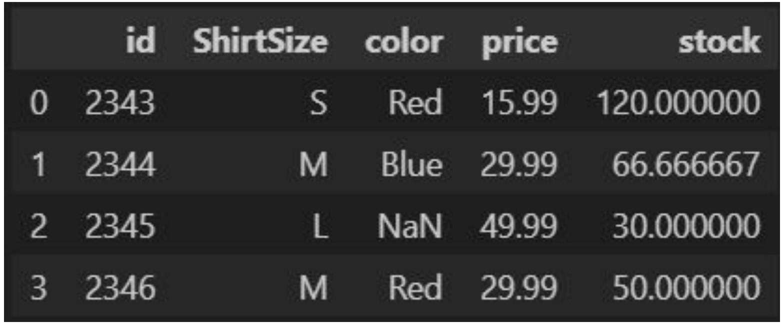


	id	ShirtSize	color	price	stock
0	2343	S	Red	15.99	120.0
1	2344	M	Blue	29.99	NaN
2	2345	L	NaN	49.99	30.0
3	2346	M	Red	29.99	50.0

Fill categorical NaN values using *Mean Imputation*:

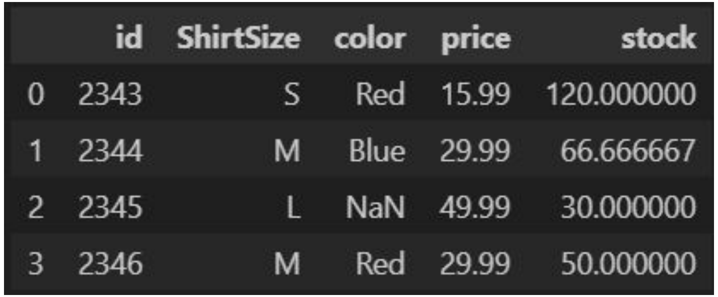


```
df['stock'] = df['stock'].fillna(df['stock'].mean())
```



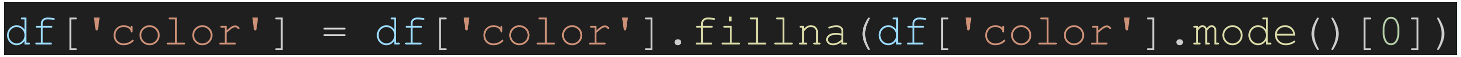
	id	ShirtSize	color	price	stock
0	2343	S	Red	15.99	120.000000
1	2344	M	Blue	29.99	66.666667
2	2345	L	NaN	49.99	30.000000
3	2346	M	Red	29.99	50.000000

Fill NaN Values (Categorical)

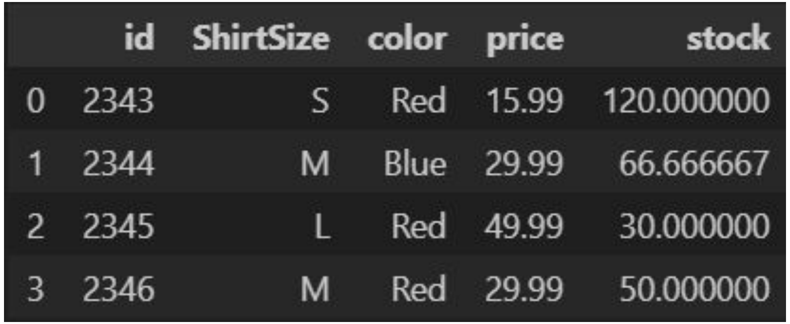


	id	ShirtSize	color	price	stock
0	2343	S	Red	15.99	120.000000
1	2344	M	Blue	29.99	66.666667
2	2345	L	NaN	49.99	30.000000
3	2346	M	Red	29.99	50.000000

Fill categorical NaN values using *Mode Imputation*:



```
df['color'] = df['color'].fillna(df['color'].mode()[0])
```



	id	ShirtSize	color	price	stock
0	2343	S	Red	15.99	120.000000
1	2344	M	Blue	29.99	66.666667
2	2345	L	Red	49.99	30.000000
3	2346	M	Red	29.99	50.000000

Suboptimal Scenarios?

- Mode Imputation:
  - Balanced Feature Values (Blue, Red, Yellow)
    - introduces class imbalance
- Mean Imputation:
  - Stock only contains values of 50 or 500
    - creates an unrealistic stock amount
- Mapping:
  - Might assume equal intervals
    - low, mid, very high
- One-Hot Encoding:
  - Increases Dimensionality

Let’s start the hacking!

QUEEN MARY MACHINE
LEARNING SOCIETY