Categorical features: the Dummy Variable Trap for Linear Regression

In general, OHE of features is the best way to deal with Categorical features in Machine Learning.

However there is a mathematical issue for some models

- linear models (like Linear Regression and Logistic Regression).

This is called the **Dummy Variable Trap**

To avoid the trap, we need to perform OHE in a slightly different way for the affected models.

Special cases are unfortunate and we will only offer a quick explanation here.

For now, when using linear models there are several alternatives to avoid the trap

- if you have a categorical variable v with $||C||$ classes
- The vector $\mathbf{v}$ should consist of $||C|| - 1$ indicators rather than $||C||$
    - this solution is common enough that several toolkits provide functions to deal with it
        - `sklearn.preprocessing.OneHotEncoder` with argument `drop="first"`
        - Pandas: `pd.get_dummies` with argument `drop_first=True`
- Use a regularizer (e.g., Ridge regression)
- *Don't* include an intercept term
    - But this may cause problems
        - Having an intercept ensures that the errors are mean 0

Dummy variable trap: Multi-collinearity in Linear Regression

Consider the class $C = \{$ "Red", "Green", "Blue" $\}$ and a categorical variable v for this class.

Suppose we create $||C||$ indicator variables

- $\mathbf{v}_{Red}, \mathbf{v}_{Green}, \mathbf{v}_{Blue}$

By construction of the OHE of v, for each example i:

$$\sum_{c \in C} \mathbf{v}_c^{(\mathbf{i})} = 1$$

This means that the indicators in $\mathbf{v}$ are perfectly collinear with the "constant" attribute 1 in each example representing the intercept term, e.g, $\mathbf{x}_0$.

$$\mathbf{X}'' = \begin{array}{cccc} \textbf{const} & \textbf{IsRed} & \textbf{IsGreen} & \textbf{IsBlue} \\ \hline \end{array} \atop \begin{pmatrix} 1 & 1 & 0 & 0 \\ 1 & 0 & 0 & 1 \\ 1 & 0 & 1 & 0 \\ \vdots & & & \end{pmatrix}$$

When one feature (e.g., the constant) is equal to a linear combination of some other features, this is called Perfect Multi-collinearity.

Linear Regression has mathematical issues with Perfect Multi-colinearity (or even with Imperfect Multi-collinearity).

This manifests itself as

- some variables with huge positive parameter values (e.g., $\Theta_{Red}, \Theta_{Blue}$)
- and other variables with huge (offsetting) negative parameter values (e.g., Θ_{Green}).

Regularization skirts the issue by enforcing a constraint that restricts large values for parameters.

By turning the parameter value of one indicator in a class to 0, we effectively eliminate 1 indicator and avoid perfect collinearity.

So where did we get lucky in our two versions of Tittanic ?

- In the first version, a binary variable for `Sex` is same as $||C|| - 1$ indicators since $||C|| = 2$
- In the second version, with a full set of indicators for `Sex` (2) and `Pclass` (3)
  - `LogisticRegression` defaults to a regularized cost function

So by luck or design, we avoided any potential Dummay Variable Trap issues.

In [48]:

```
print("Done")
```

```
Done
```