# Machine learning
# Data science

RITIK BILALA
ritik.bilala@gmail.com

TODAY……. .

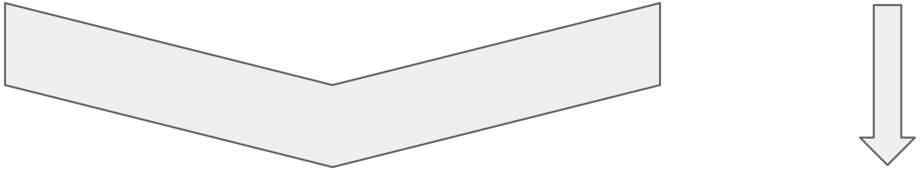
# CATEGORICAL FEATURES

**By: Ritik Bilala 2020**

# FEATURE ENCODING
# FEATURE SELECTION

# Consider this table………….

| Item_no | Item_Catg. | Gender | Age | Salary | Purchased |
|---|---|---|---|---|---|
| 1 | *Fitness* | *M* | 20 | 30000 | Yes |
| 2 | *Fitness* | *F* | 50 | 70000 | No |
| 3 | *Food* | *M* | 35 | 50000 | Yes |
| 4 | *Kitchen* | *M* | 22 | 40000 | No |
| 5 | *Kitchen* | *F* | 30 | 35000 | Yes |



**FEATURES**

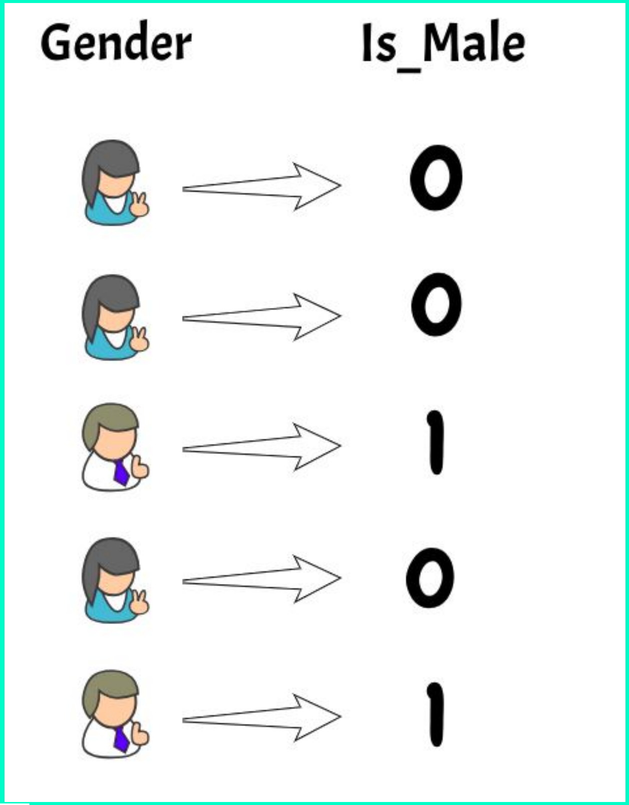
**TARGET**

# Let us try to fit logistic Regression model

BUT HOW ?
THERE IS LOT OF
NON-NUMERICAL DATA

# We need to encode categories into numbers

# Label ENcoding for dichotomous feautre



| Item_no | Item_Catg. | Gender | Age | Salary | Purchased |
|---|---|---|---|---|---|
| 1 | Fitness | 1 | 20 | 30000 | 1 |
| 2 | Fitness | 0 | 50 | 70000 | 0 |
| 3 | Food | 1 | 35 | 50000 | 1 |
| 4 | Kitchen | 1 | 22 | 40000 | 0 |
| 5 | Kitchen | 0 | 30 | 35000 | 1 |

Let's Apply encoding to Dichotomous : **Gender and Purchased**

BUT WAIT... ... . .

# What About….

# Features with more than 2 categories

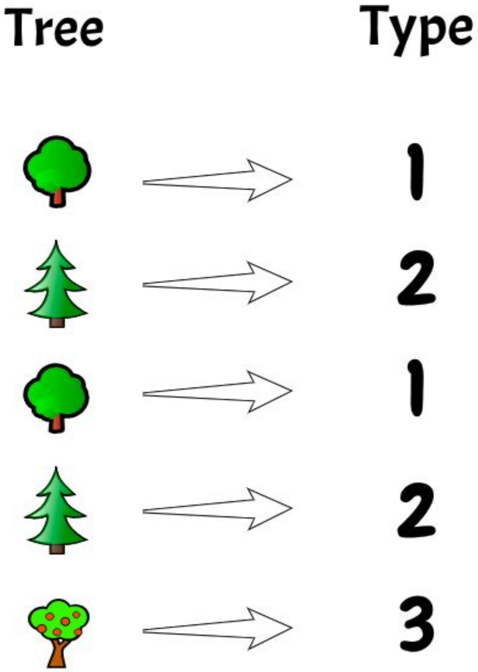
LET'S DISCUSS TYPES OF ENCODING

AND DECIDE...

WHICH IS BEST FOR YOU

# Types of Encodings

# Ordinal Encoding or Label Encoding



Each category is given number from 1 to N = total categories

In our case:

| Item_Catg. | Enc. |
|---|---|
| *Fitness* | 1 |
| *Fitness* | 1 |
| *Food* | 2 |
| *Kitchen* | 3 |
| *Kitchen* | 3 |

But…

There is a issue

What ???

When feeding data into mathematical model

1 , 2 ,3..

Will be number in order

They won't have equal weightage

SOME NUMBERS ARE GREATER THAN OTHERS, THIS CAN SKEW THE MODELS LEADING TO INACCURATE RESULTS.

The solution to this problem is achieved by incorporating the concept of dummy columns

# ONE HOT ENCODING

Here, we map each category to a vector that contains 1 and 0 denoting the presence of the feature or not.

| Item_Catg. | Fitness_ | Food_ | Kitchen |
|---|---|---|---|
| *Fitness* | *1* | *0* | 0 |
| *Fitness* | *1* | *0* | 0 |
| *Food* | *0* | *1* | 0 |
| *Kitchen* | *0* | *0* | 1 |
| *Kitchen* | *0* | *0* | 1 |

For high cardinality features, this method produces a lot of columns that slows down the learning of the model significantly.

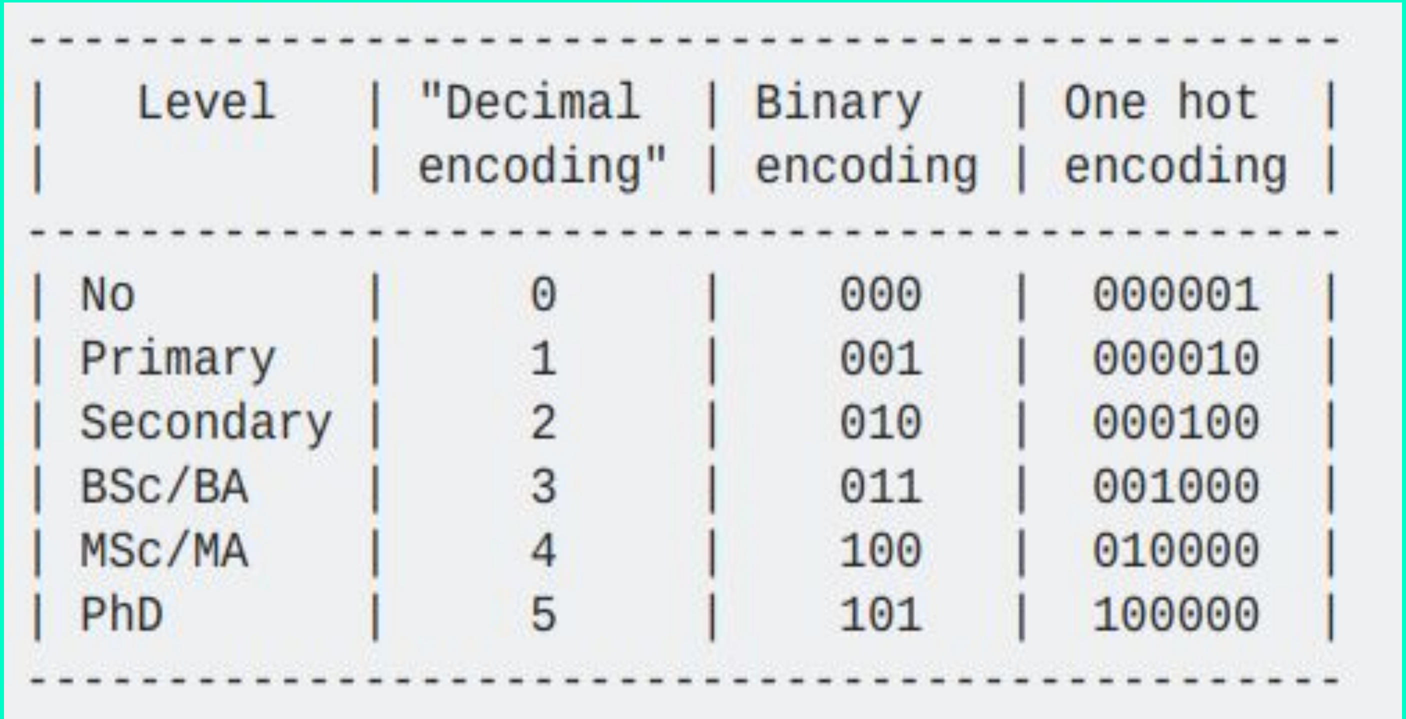
# BINARY ENCODING

First, the categories are encoded as ordinal, then those integers are converted into binary code, then the digits from that binary string are split into separate columns.

1. This is useful when you have a large number of categories

2. binary encoding is a good choice to encode the categorical variables with less number of dimensions.

# ALL IN ONE EXPLANATION



| Level | "Decimal encoding" | Binary encoding | One hot encoding |
|---|---|---|---|
| No | 0 | 000 | 000001 |
| Primary | 1 | 001 | 000010 |
| Secondary | 2 | 010 | 000100 |
| BSc/BA | 3 | 011 | 001000 |
| MSc/MA | 4 | 100 | 010000 |
| PhD | 5 | 101 | 100000 |

# Now, Something interesting

# TARGET/MEAN ENCODING

# TARGET/MEAN ENCODING

Here, features are replaced with a blend of the posterior probability of the target for the given particular categorical value and the prior probability of the target over all the training data.

| Item_Catg. | Purchased | Ordinal_enc | Mean_enc |
|---|---|---|---|
| *Fitness* | *1* | *1* | 0.5 |
| *Fitness* | *0* | *2* | 0.5 |
| *Food* | *1* | *3* | 1 |
| *Kitchen* | *0* | *4* | 0.5 |
| *Kitchen* | *1* | *5* | 0.5 |

1. Fitness = [Average of Purchased under the label fitness]

2. Food = [Average of Purchased under the label food

Same for others too

IT SOLVES BOTH THE ENCODING TASK AND ALSO CREATES A FEATURE THAT IS MORE REPRESENTATIVE OF THE TARGET VARIABLE – ESSENTIALLY HITTING TWO TARGETS AT ONCE.

# Blending……

The blending parameter defines whether the target average should be weighted based on the count of the group.

It is often the case, that some groups may have a small number of records and the target average will be unreliable.

To prevent this, the blended average takes a weighted average of the group’s target value and the global target value.

# But why IS MEAN encodings Better ?

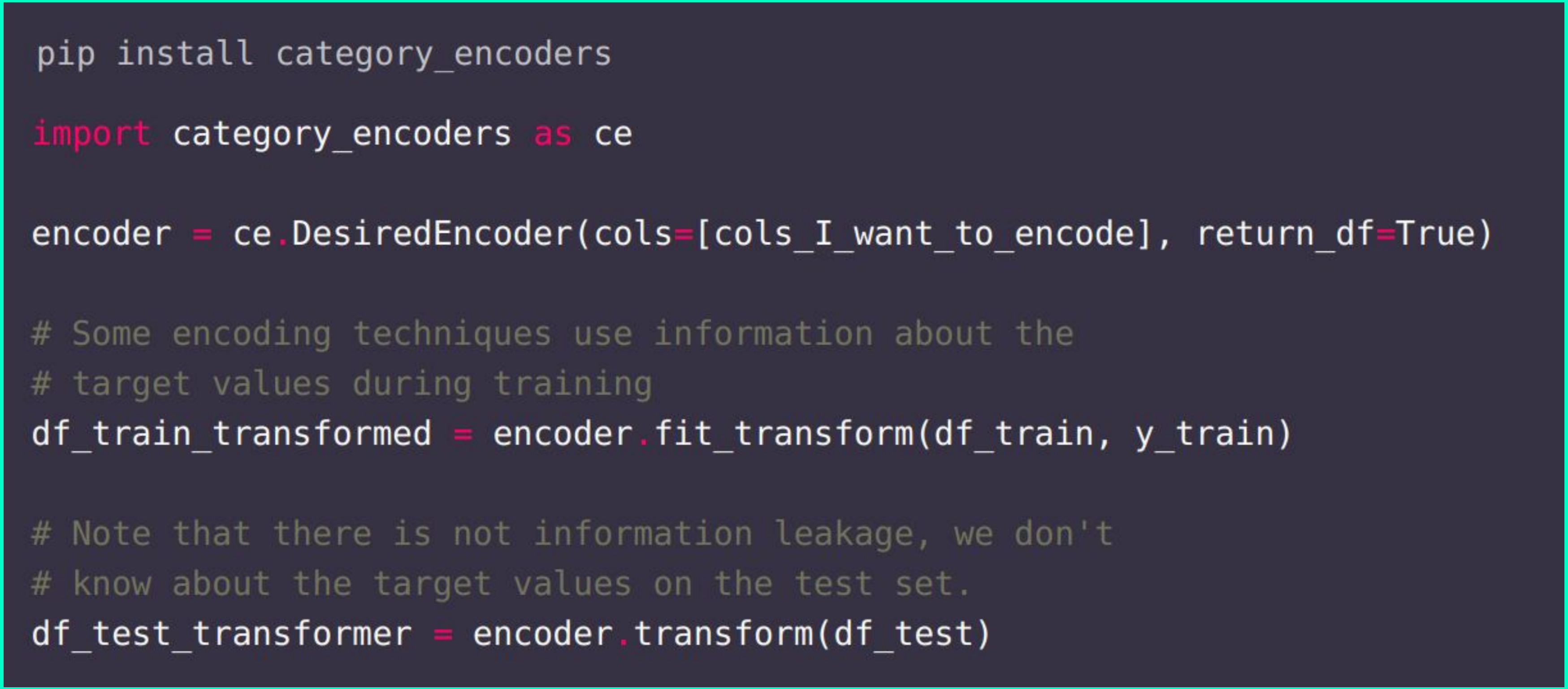
1. Mean encoding can embody the target in the label whereas label encoding has no correlation with the target

2. In case of large number of features, mean encoding could prove to be a much simpler alternative

3. A histogram of predictions using label & mean encoding show that mean encoding tend to group the classes together whereas the grouping is random in case of label encoding

4. But overfitting may be an issue, can be solved using regularisation

# ENCODING IN PYTHON 3



```
pip install category_encoders

import category_encoders as ce

encoder = ce.DesiredEncoder(cols=[cols_I_want_to_encode], return_df=True)

# Some encoding techniques use information about the
# target values during training
df_train_transformed = encoder.fit_transform(df_train, y_train)

# Note that there is not information leakage, we don't
# know about the target values on the test set.
df_test_transformer = encoder.transform(df_test)
```

## SOme Other encoding to explore yourself:

1. James-Stein encoders
2. Hash Encoder
3. Dracula

What is the next step ????
How to find which feature to use ?

Correlation ?

Pearson linear correlation?

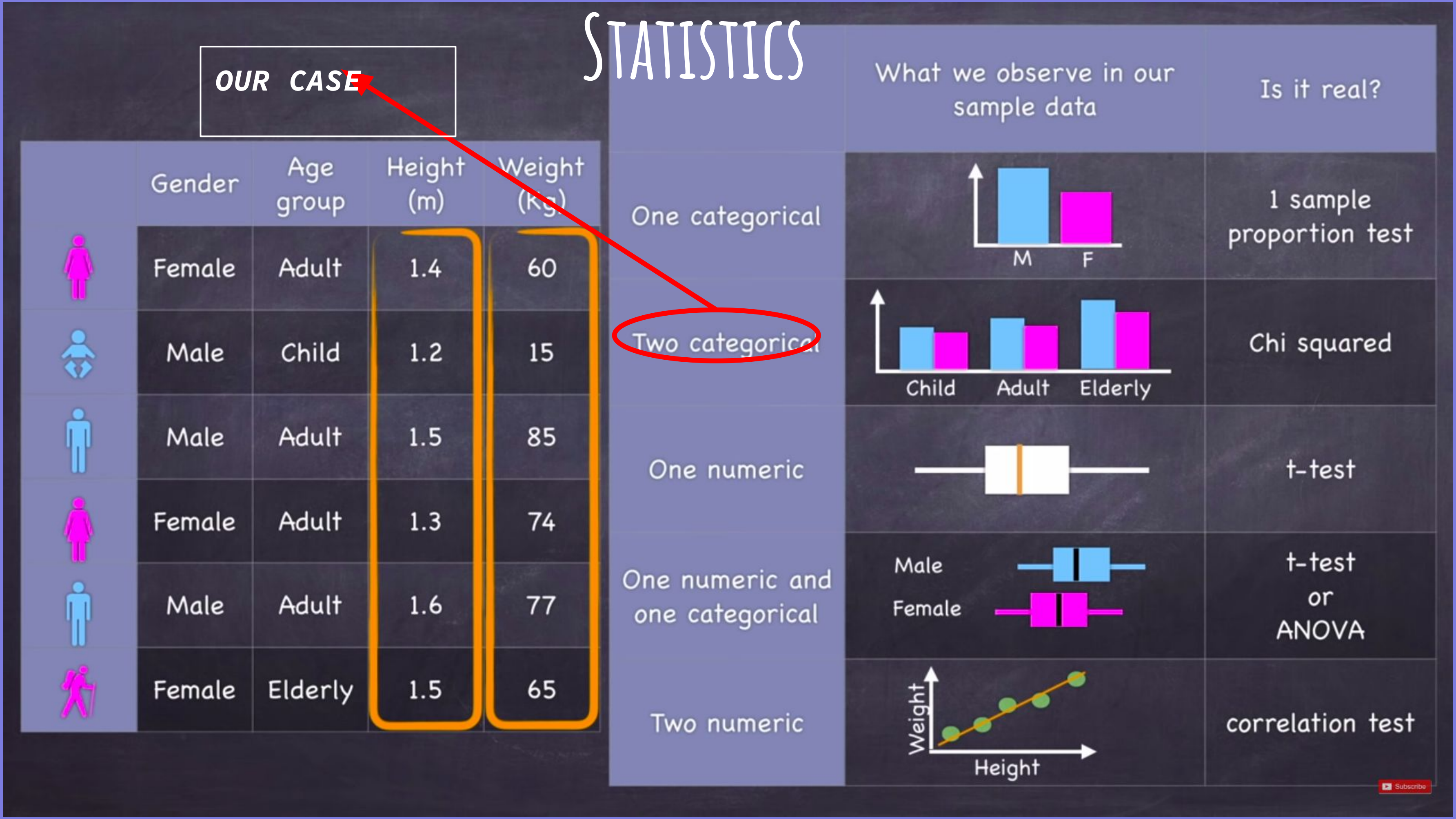
Nope !!

Categories have no numerical relation

so linear correlation won’t help

For a dichotomous categorical variable and a continuous variable you can calculate a Pearson correlation if the categorical variable has a 0/1-coding for the categories.

This correlation is then also known as a point-biserial correlation coefficient.



# STATISTICS

OUR CASE

|  | Gender | Age group | Height (m) | Weight (Kg) |
|---|---|---|---|---|
|  | Female | Adult | 1.4 | 60 |
|  | Male | Child | 1.2 | 15 |
|  | Male | Adult | 1.5 | 85 |
|  | Female | Adult | 1.3 | 74 |
|  | Male | Adult | 1.6 | 77 |
|  | Female | Elderly | 1.5 | 65 |

|  | What we observe in our sample data | Is it real? |
|---|---|---|
| One categorical |  | 1 sample proportion test |
| Two categorical |  | Chi squared |
| One numeric |  | t-test |
| One numeric and one categorical |  | t-test<br>or<br>ANOVA |
| Two numeric |  | correlation test |

# Pearson Chi-Square test:

The Chi–Squared test is a statistical hypothesis test that assumes (the null hypothesis) that the observed frequencies for a categorical variable match the expected frequencies for the categorical variable. The test calculates a statistic that has a chi–squared distribution, named for the Greek capital letter Chi (X) pronounced “ki” as in kite.

# CHI-SQUARED

## FOR FEATURE SELECTION

To use $\chi^2$ for feature selection, we calculate $\chi^2$ between each feature and the target, and select the desired number of features with the best $\chi^2$ scores.

The intuition is that if a feature is independent to the target it is uninformative for classifying observations.

# of observations in class i

$$\chi^2 = \sum_{i=1}^{n} \frac{(O_i - E_i)^2}{E_i}$$

# of expected observations in class i if there was no relationship between the feature and target.

# Select 'k' best Categorical features using chi2 test

-scikitlearn

# SCIKIT-LEARN

```
from sklearn.feature_selection import chi2
from sklearn.feature_selection import SelectKBest

# load the dataset
X, y = load_dataset('file.csv')

# split into train and test sets
X_train, X_test, y_train, y_test = train_test_split(X,
y,test_size=0.33, random_state=1)

fs = SelectKBest(score_func=chi2,k='5')
fs.fit(X_train, y_train)
```

# Now Apply it in your project

THANKS FOR WATCHING IT !!!!!