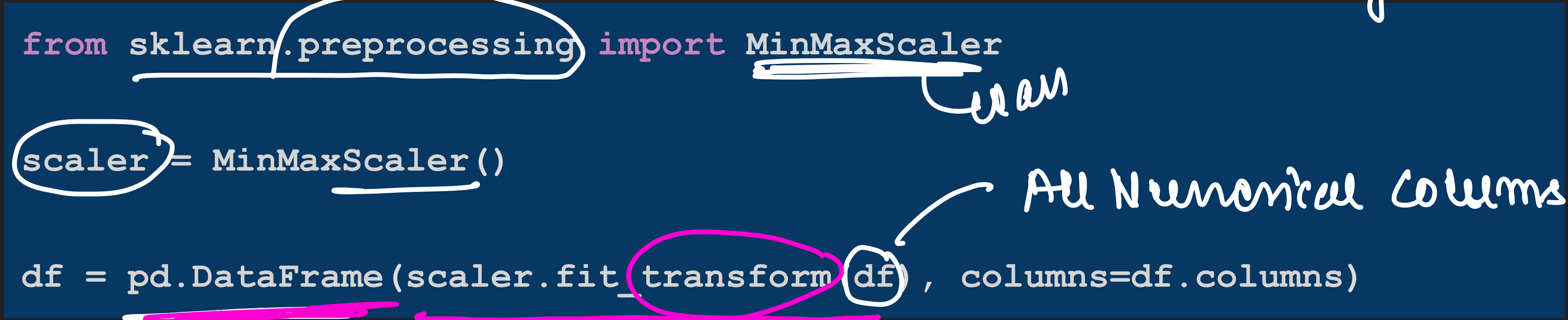
Cau24 EDA.

Line of Code

steps before model training



```
from sklearn.preprocessing import MinMaxScaler

scaler = MinMaxScaler()

df = pd.DataFrame(scaler.fit_transform(df) , columns=df.columns)
```

class

All Numerical columns.

① Cat + Num

② Feature Scaling

Data Notation

sample - m

dimension / # features - d

Predictors - (m, d) matrix X

ith sample - $x^{[i]}$ — one row

jth feature - x_j - one column

True output / target - y^i

Predicted output / target $\hat{y}^i$

Goal of generalization in ML

m samples - $\{x^i, y^i\}^m$

Historical Data. → Train an ML Model

$x^i \xrightarrow[\text{Model}]{\text{ML}} \hat{y}^i$ ← Predicted value

Ideally, $y^i \approx \hat{y}^i$ → Predicted Output

→ True output

⇒ For all m samples $\boxed{y^i \approx \hat{y}^i}$ → GOOD TRAINING

DO WE WANT TO PERFORM PREDICTION ON TRAINING

DATA??

For training data y is already present?

Why to even do prediction.

Goal of ML: GENERALISATION.

You model should perform well both on training and NEW data.

True output

$$\underset{\text{New}}{X^i} \xrightarrow[\text{Model}]{\text{ML}} \hat{y}^i \approx y^i$$

✓ Compare??

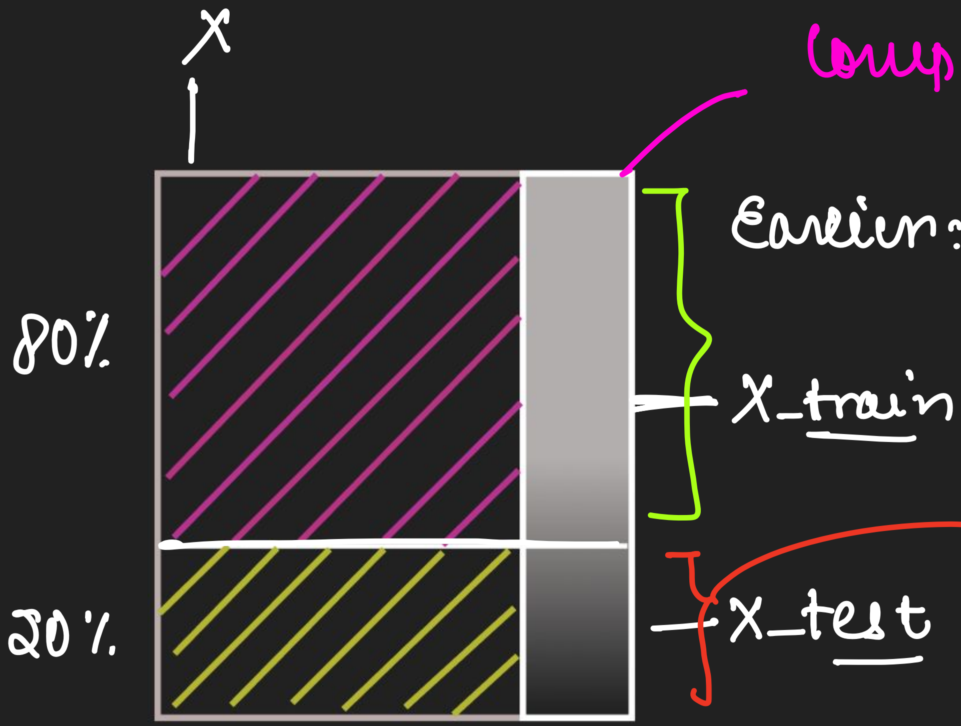
UNKOWN

Check if model is GENERALISING??

Problem: No labels for new data.

Solution:

Because we don’t know the ground truth (y) for New Sample



Complete Historical Data.

Earlier: Use X for training

X_train → Training Data

Proxy for new data.

Now: Use only X_train for training

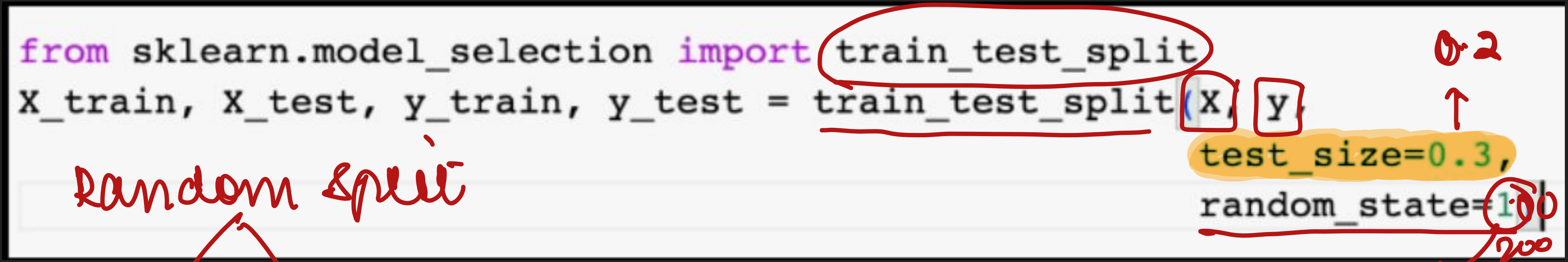
Phases for Model Development

① Training - use trainin split for model training

$y_{train} \approx \hat{y}_{train}$ — Model is learning

② Testing / Evaluation - test split

$y_{test} \approx \hat{y}_{test}$ - Model is generalising



```
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X, y,
                                                    test_size=0.3,
                                                    random_state=100)
```

0.2

Random split

80%   20%

200

200

seed.