13-09-2025

Agenda:

FE - V

→ Backward Elimination → p-value → remove the irrelevant one (highest)
- remove highest p-value column
- Train the model again (ML)

→ Forward selection / Elimination → R^2 value → add the relevant ones
→ R^2_absolute → filter the irrelevant ones
→ p-value

p-value: measure the prob of observing the data that you have

< 0.0

H_0: states that coefficient for that feature is 0

$y = m_0 + m_1x_1 + m_nx_n$

< 0.05 → reject H_0

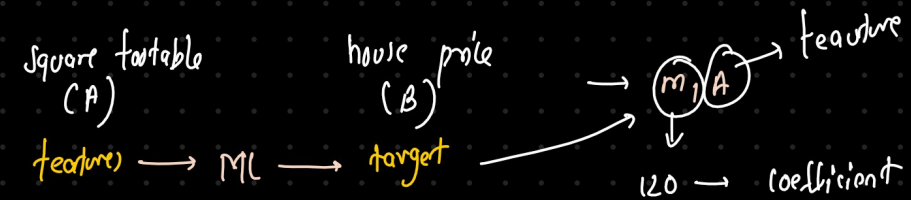
> 0.05 → fail to reject H_0

Example:

regression model → Linear Regression

↓ Type

↓ Name of Algo



SE = 40

$$T\text{-statistic} = \frac{\text{coefficient}}{SE} = \frac{120}{40} = 3$$ → T-table → 3.182 (P(value))

↓ not correct value

p-value < 0.05

→ fail to reject H_0

→ Forward Selection → R^2 (0-1)

Actual_Target_Column (A)	Model predicted (B)	Mean (C)
2	2.5	3.5
5	4.0	3.5
3	4.0	3.5
4	3.5	3.5

$$R^2 = 1 - \frac{SS\_res}{SS\_total}$$

SS_res → Sum of squares of Residual

SS_total : $\sum (A - C)^2$

: $(2-3.5)^2 + (5-3.5)^2 + (3-3.5)^2 + (4-3.5)^2$

: 5.0

SS_res : $\sum (A - B)^2$

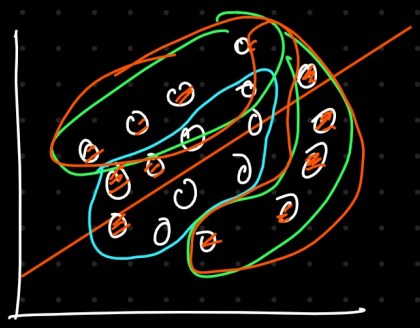
: $(2-2.5)^2 + (5-4.0)^2 + (3-4.0)^2 + (4-3.5)^2$

: 2.5

$$R^2 = 1 - \frac{2.5}{5.0}$$

$$= 1 - \frac{1}{2} = 1 - 0.5 = 0.5$$

mean our model explain 50 % of the variance of the data.



50%.

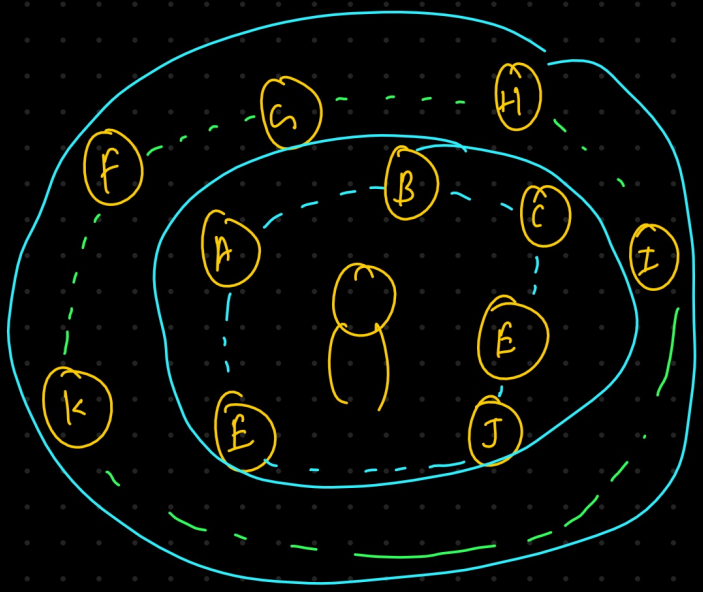
$R^2 = 0.5$

$R^2 = 0.1$

A B C D E F G H Target_variable

A → T →
B → T
C - D

→ R^2 ↑



$R^2 = 0.5$

10 day

50 %. ↑

50 %. X

A, B, C, D, E

A, B, C, D, E → 5 features

```
def forward_selection(X, y, threshold=0.05):
    selected_features = []
    remaining_features = list(X.columns)
    while remaining_features:
        pvals = {}
        # Keep adding all the features one by one and get the p_value of each selected features
        for feature in remaining_features:
            model = sm.OLS(y, sm.add_constant(X[selected_features + [feature]])).fit()
            pvals[feature] = model.pvalues[feature]
        # find minim from the value and get the key of the min values
        min_pval_feature = min(pvals, key=pvals.get)
        min_pval = pvals[min_pval_feature]
        if min_pval < threshold:
            selected_features.append(min_pval_feature) # add the feature we selected to select_features
            remaining_features.remove(min_pval_feature) # remove the feature we selected from X
            print(f"Selected {min_pval_feature} with p-value {min_pval:.4f}")
        else:
            break
    return selected_features
```

B, C, D, E

[A] + B

A

A → 0.01

[A, C]

model → A →

< A, C, B → 0.1
A, C, D → 0.2
A, C, E → 0.3

0.1 < 0.05

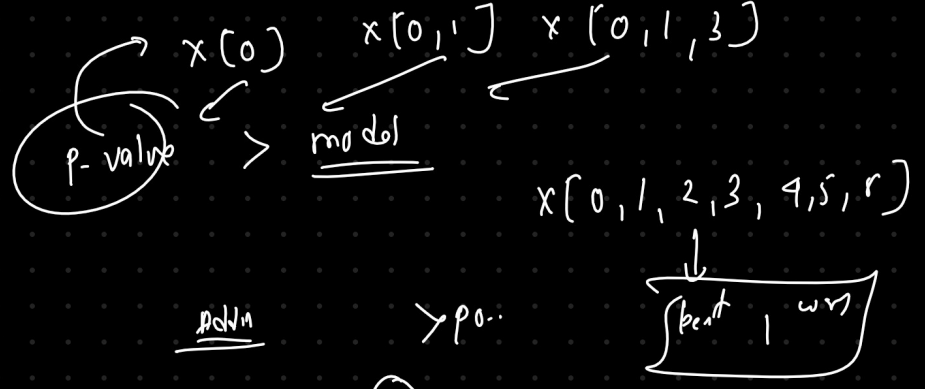
y

A

if 0.01 < 0.0

If → model
const + x[0]

p. value > 0.05



RFE (Recursive Feature Elimination)

↓

method to select features

↓

algo → linear - Regression

↘ Random forest

$$y = m_0 + m_1 x_1 + m_2 x_2 + m_3 x_3$$

Type_of_car	Engine_CC	Car_color	price	Type of car > Engine_CC > Car_color
Sedan	250	Red	Y_1	
Hatchback	200	white	Y_2	
Sedan	308	Blue	Y_3	
Sport car	600	white	Y_4	

$$y = m_0 + m_1 x_1 + m_2 x_2 + m_3 x_3$$

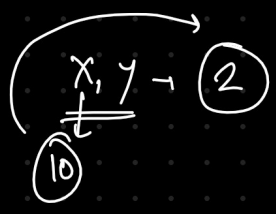
price: m_0 = 100, m_1 = 23.62, m_2 = 11.11, m_3 = 3.16

check the coefficient of features: $m_1 \rightarrow x_1$

$m_2 \rightarrow x_2$

$m_3 \rightarrow x_3$

RFE (algo, imp_features_to_select = 2)



RFE :—

Tree based method!

→ linear

→ non-linear

↓

linear-regression

→ linear data

Data → Data Analysis → Data cleaning → Data Handling

Data encoding

1 million rows, 120 column

↓

features

feature selection

BE → 40

FS/FE → 15 14 → 30 col

Tree based → 30

↓

120 → 40 column

30 column

model

78%.

↑ ↓ 15 ↑

15 column

84%.

result

90%. → 10%.

0.7 to 1.0 or -0.7 to -1.0 → highly correlated

Feature → Target column

LotArea → SalePrice → keep the feature

Feature → Feature

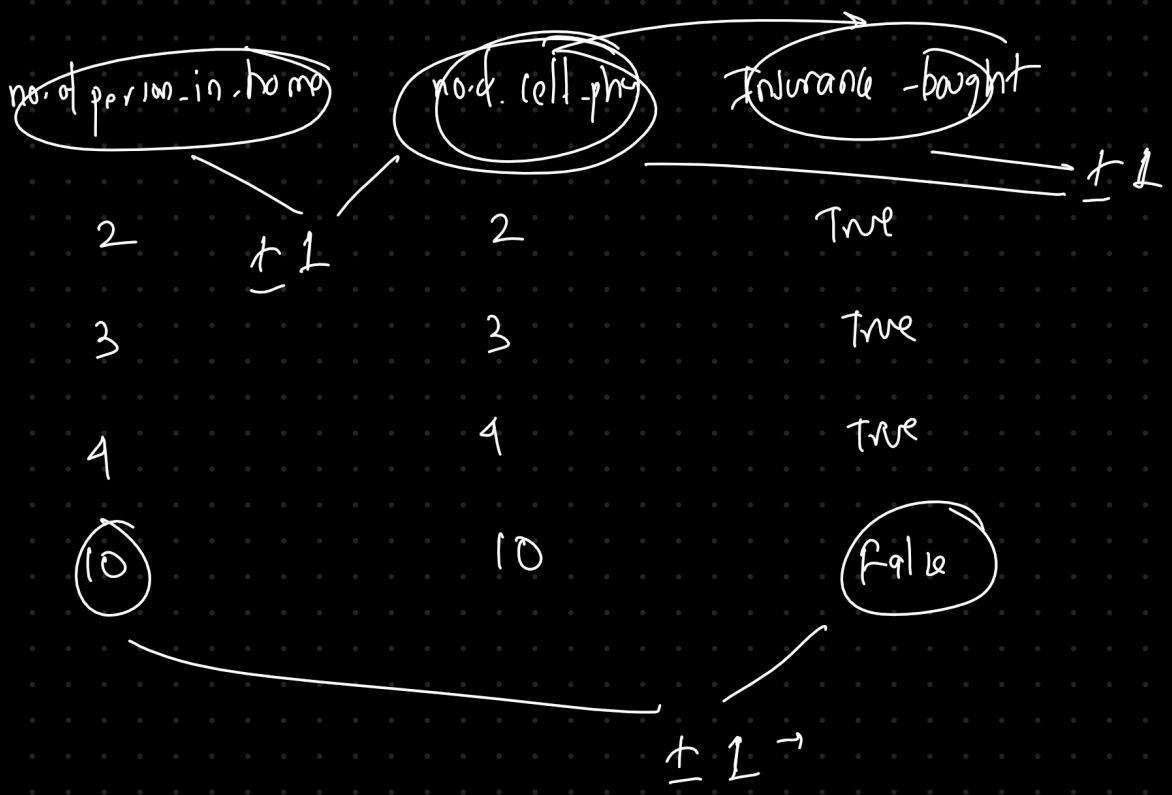
LotArea GrLivArea → keep only one

0.5 to 0.7 or -0.5 to -0.7

usually fine to keep

0.0 to 0.3 or -0.3 to 0.0

weak correlation (little linear relation)



VIF : Variance Inflation Factor

→ detect multicollinearity in regression feature

→ feature → correlated ← feature)

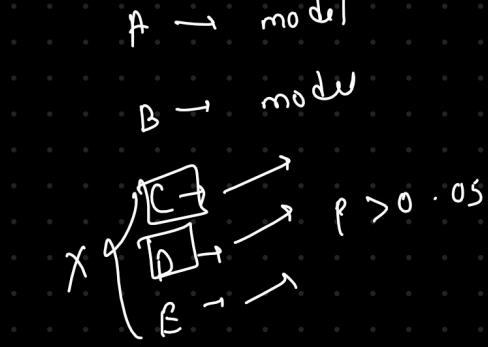
$$VIF = \frac{1}{1 - R^2}$$

VIF : 1 : No multicollinearity

: 1-5 : Moderate correlation

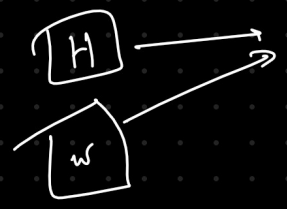
: >10 : Strong multicollinearity

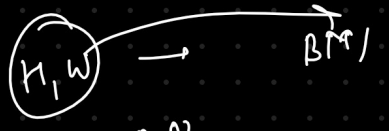
x_1 : A bedroom	x_2 : B no of floors	→ C
2	2	1200 $
3	3	1600 $
4	4	1700 $
5	5	1900 $



Height → $p > 0.06$ → BMI

weight → $p > 0.06$ → BBI





It → BE → 99%
→ VIF

→ BE ∪ FE 92%.
→ VIF

→ BE → 90%.
→ comet & VIF