https://colab.research.google.com/drive/1KxIeE7hhOlx-Zp-hdAdhN_6uCJ3UKwdc?usp=sharing

Linear Regression - 4

Agenda

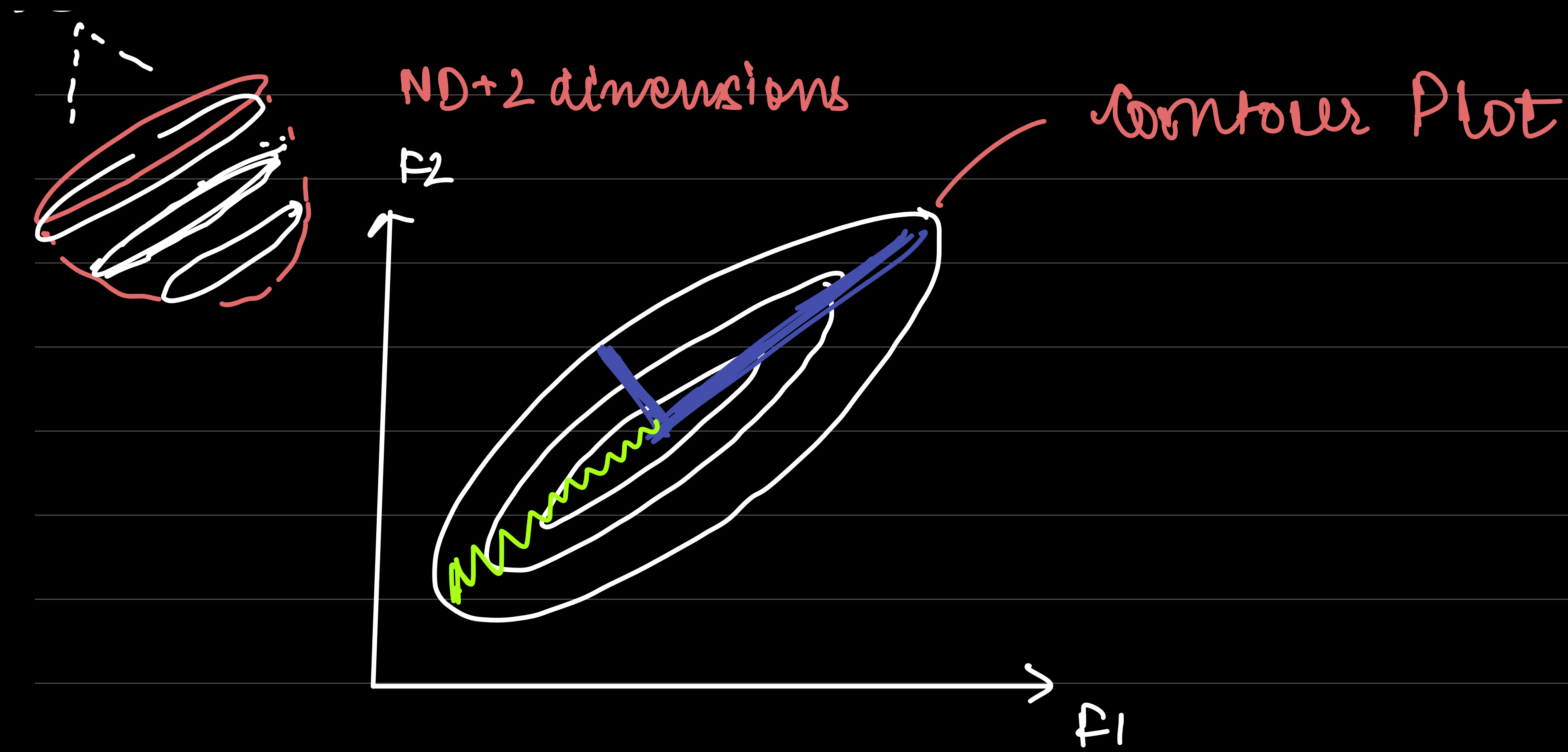
- Finish scratch code for Lin Reg ✓
- Adjusted R- square ✓
- Assumption of Linear Regression (by statisticians)

  5 assumptions (1, 0.5x2), 3, 4, 5 — pending

- Ment after break.

Importance of Feature Scaling in Machine Learning

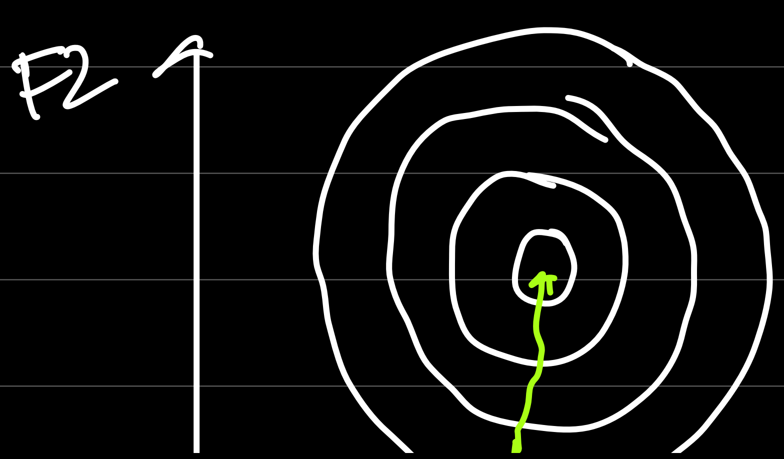
- Similar units → Interpret the weights / coefficients
- Helps GD converge faster ☆☆☆



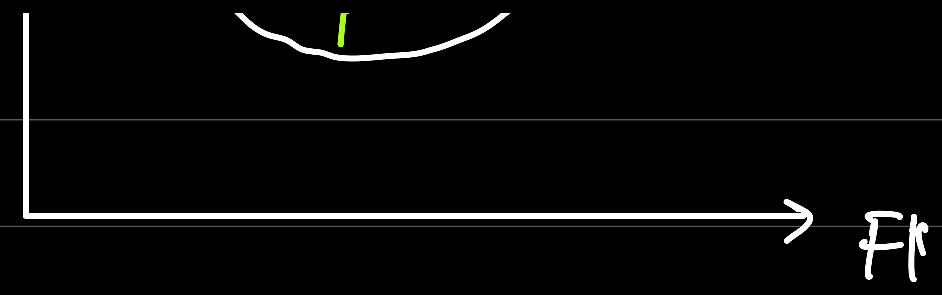
If non feature scaled

- F_1 F_2 may have diff scale, range, variance

If we scale the feature.



Training would be faster here



Problems with R-squared

Lin Regression - N features + 1 feature

R-squared (R1)

$$w_0 + w_1x_1 + w_2x_2 + \dots + w_nx_n + w_{n+1}x_{n+1}$$

Re-calculate R-square: (R2)

$R_1 \; > \; < \; = \; R_2$???

case1) $R1 > R2$ ✗

Model always have an option to make $w_{n+1} = 0$

case2) $R1 < R2$ ✓

New feature lead to better performance. (Trivial)

case3) $R1 = R2$ $d \uparrow \Rightarrow$ Performance —

No added value

Prefer this ✓

M1

d features

R

M2

d+2 features

R

Why prefer M1 over M2 ??

Occam's Razor

Preference for simplicity

Goal: Adjust the metric R2 in such a way that it penalises calculation if with ↑ in d, there is NO increase in performance

$$\text{Adj. R-squared} = 1 - \underbrace{\left[\frac{(1-R^2)(m-1)}{(m-d-1)}\right]}_{A}$$

If $d\uparrow \rightarrow$ denom $\downarrow$, R^2 - no change, $A\uparrow$, Adj-R^2 - $\Downarrow$

If df → denom ↓, (k − 1) ↑, A ↑↓ depends which is more extreme.

Performance increased

Intro to statsmodel

Colab

statistics. ML

statsmodel Sklearn

- p-value
- confidence interval
- coefficients / std error
- Residual plot
- QQ plot

- GD
- Feature Scaling
- Cross validation — NEXT CLASS
- Evaluation Metrics

Summary: Classification Models

Accuracy: Classification Metrics

Performance

$R2 = 0.75$

75% of variance in y can be explained by the model.

Assumptions of Linear Regression.

1. Linearity ✓
2. NO MULTI-COLLINEARITY 0.5
3. Normality of Residuals (Errors)
4. No Heteroskedasticity
5. No Autocorrelation.

Next class.

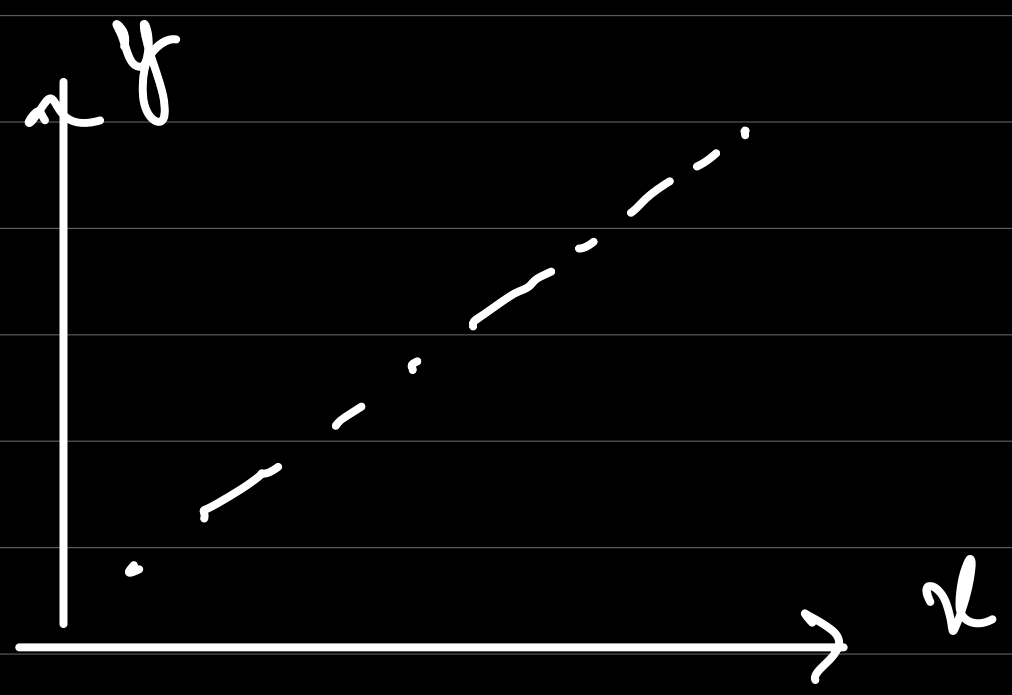
① Assumption of linearity

You model will perform well if

$$X \xrightarrow{f} y$$

Trivial.

f : linear



② No multi-collinearity

⇒ Collinearity

f_1 and f_2

$$f_1 = \alpha + \beta f_2$$

f_1 · linear Tranformation of f_2

f_1 and f_2 are collinear.

Collinear Variables

- Age of car, model Year
- Deg in celcius and F

⇒ Multi-collinearity

f_1, f_2, f_3, f_4

$f_1 = \alpha_1 + \alpha_2 f_2 + \alpha_3 f_3 + \alpha_4 f_4$

$f_1 \; f_2 \; f_3 \; f_4$ are multicollinear.

$\Downarrow$

They can be written as lin. comb. of each other

NO MULTI-COLLINEARITY. WHY?

$f_1, f_2, f_3.$

$M1 - [w_1, w_2, w_3] - \langle 1, 2, 3 \rangle$

f_2 and f_1 are collinear $\Rightarrow$ $f_2 = 1.5 f_1$

$$\hat{y} = w_0 + w_1 x_1 + w_2 x_2 + w_3 x_3$$

$$\hat{y} = w_0 + x_1 + 2x_2 + 3x_3$$

$$\hat{y} = w_0 + x_1 + 3x_1 + 3x_3$$

$$\hat{y} = w_0 + 4x_1 + 0 \cdot x_2 + 3x_3$$

$\langle 4, 0, 3 \rangle$ ✓

$\langle 1, 2, 3 \rangle$ ✓

Feature Imp. ??

Unstable Training

Goal : To remove redudant features

⇓

TO HAVE NO MULTI-COLLINEARITY

VIF - Variance Inflation Factor

← Next Class.