DUT · SEMESTER 3 · COMPUTER SCIENCE

Machine Learning

Session 03 · Linear regression: predicting a number

Dr. Soufiane Hourri, PhD, HDR

Associate Professor & AI Consultant

École Supérieure de Technologie de Safi · Cadi Ayyad University

2026-2027

Today's session

01 The regression line

02 Simple regression on the penguins

03 Measuring errors

04 Several features

05 Checking the model

06 Lab: fuel consumption of cars

TODAY'S FOCUS

In Session 01, r = 0.87 told us that flipper length and body mass move together. Today we turn that relationship into predictions, and we measure how wrong they are.

Learning objectives

By the end of this session, you will be able to

1. Explain slope, intercept and residual
2. Compute a least-squares line by hand
3. Evaluate LinearRegression with MAE, RMSE and R^2
4. Interpret a coefficient with its unit, "all else equal"
5. Read a residual plot

PREREQUISITES

- Session 01: mean, standard deviation, correlation r
- Session 02: split, one-hot encoding, pipelines
- A calculator

Warm-up from Session 02

QUESTION

Before a model can use the column island (Biscoe, Dream, Torgersen), what should we do?

A

Replace the islands by 1, 2 and 3

B

One-hot encode it, with categories learned on the training set

C

One-hot encode it, with categories learned on the whole dataset

D

Drop it: models cannot use text

01

CHAPTER

The regression line

Of all the lines we could draw through a cloud of points, which one is the best?

The equation of a line

$$\hat{y} = b_0 + b_1 \cdot x$$

SYMBOLS

x : the feature (input)

$\hat{y}$: the predicted value, read "y hat"

b_1 : slope, change in ŷ when x increases by 1

b_0 : intercept, the value of ŷ when x = 0

INTUITION

Two numbers describe the whole model.

The slope says how strongly y follows x, in the units of y per unit of x.

A residual is the error on one observation

DEFINITION

The residual of observation i is the difference between its true value and the prediction: $e_i = y_i - \hat{y}_i$. It is the vertical distance between the point and the line.

EXAMPLE

A student studied 4 hours and got 4/10. The line predicts 5.4. Residual: 4 − 5.4 = −1.4. The model predicted too high.

SIGN

e > 0: point above the line, prediction too low
e < 0: point below the line, prediction too high
e = 0: the line passes through the point

Least squares: the line with the smallest squared errors

$$b_1 = \Sigma(x_i - \bar{x})(y_i - \bar{y}) \,/\, \Sigma(x_i - \bar{x})^2 \qquad b_0 = \bar{y} - b_1 \cdot \bar{x}$$

THE CRITERION

Choose b_0 and b_1 that minimise the sum of squared residuals $\Sigma\, e_i^2$

This is ordinary least squares (OLS)

INTUITION

Squares remove signs and punish large errors, as in the variance.

The line always passes through the point $(\bar{x}, \bar{y})$.

Worked example: computing the slope by hand

x_i (hours)	y_i (grade)	$x_i - \bar{x}$	$y_i - \bar{y}$	$(x_i - \bar{x})(y_i - \bar{y})$	$(x_i - \bar{x})^2$
1	2	−2	−2.4	4.8	4
2	4	−1	−0.4	0.4	1
3	5	0	0.6	0	0
4	4	1	−0.4	−0.4	1
5	7	2	2.6	5.2	4
$\bar{x} = 3$	**$\bar{y} = 4.4$**			**Sum: 10**	**Sum: 10**

Example data. $b_1 = 10 / 10 = 1.0$ point per hour. $b_0 = 4.4 - 1.0 \times 3 = 1.4$. Line: $\hat{y} = 1.4 + 1.0\,x$.

The fitted line and its residuals

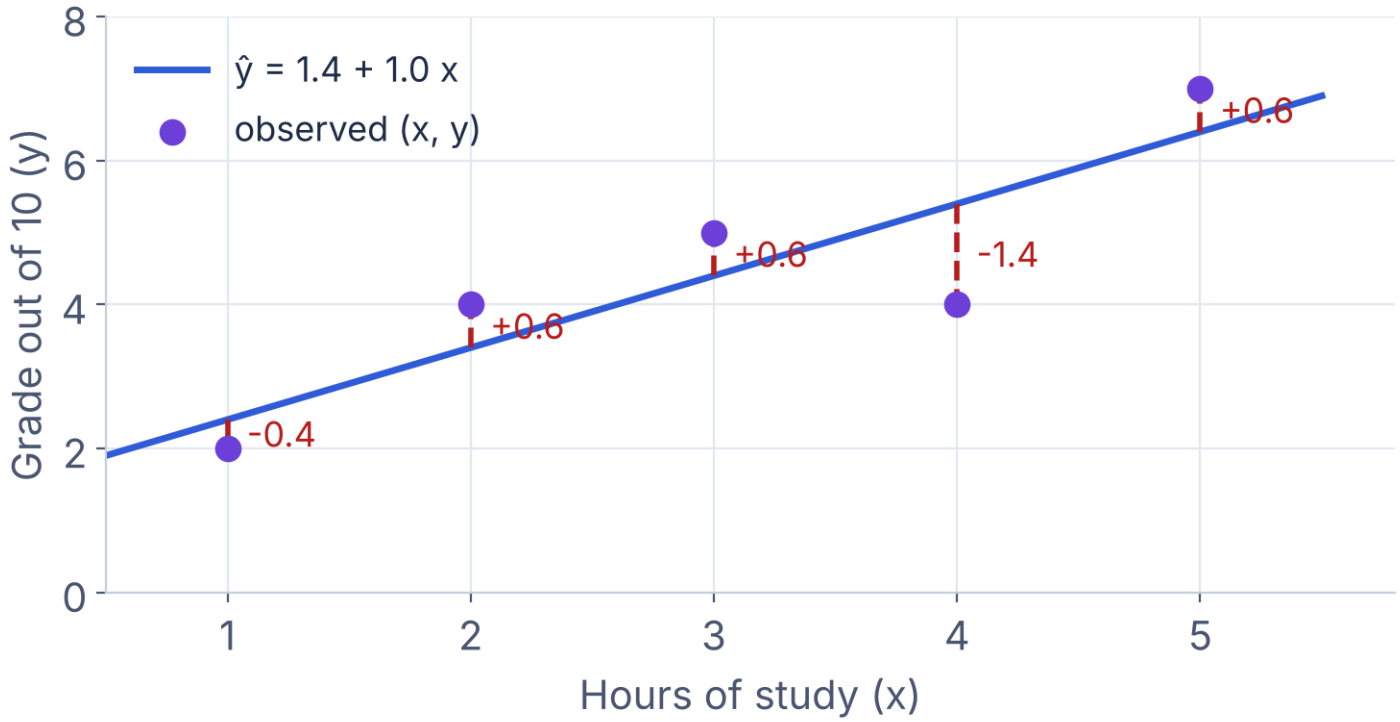


Figure 1. Example data, five students. Dashed segments are the residuals $e_i = y_i - \hat{y}_i$.

WHAT TO NOTICE

- Residuals: −0.4, +0.6, +0.6, −1.4, +0.6
- They add up to 0
- $\Sigma e_i^2 = 3.2$: no other line does better
- The biggest error dominates the sum

Knowledge check

QUESTION

With $\hat{y} = 1.4 + 1.0\,x$, what grade does the model predict for a student who studied 3.5 hours?

A

4.9

B

3.5

C

5.4

D

4.4

The slope is the correlation, rescaled

- **$b_1 = r \times s(y) / s(x)$**
- r gives the direction and strength; s(y) / s(x) converts to the units of y per unit of x
  - Penguins: 0.871 × 802 g / 14.06 mm ≈ 49.7 g per mm
- Same sign as r. If r = 0, the line is flat: $\hat{y} = \bar{y}$ for everyone

KEY TAKEAWAY

Simple regression turns a correlation into a prediction rule with units.

02

CHAPTER

Simple regression on the penguins

Predict body mass from flipper length with scikit-learn, then read the model.

LinearRegression in scikit-learn

CODE · PYTHON

```python
from sklearn.linear_model import (
    LinearRegression)

df = df.dropna(subset=["body_mass_g"])
X = df[["flipper_length_mm"]]
y = df["body_mass_g"]

lr = LinearRegression()
lr.fit(X, y)
print(lr.intercept_, lr.coef_)

new = pd.DataFrame({"flipper_length_mm": [200]})
print(lr.predict(new))
```

OUTPUT

```
-5780.8 [49.69]

[4156.]
```

Values rounded for display.

The least-squares line through 342 penguins

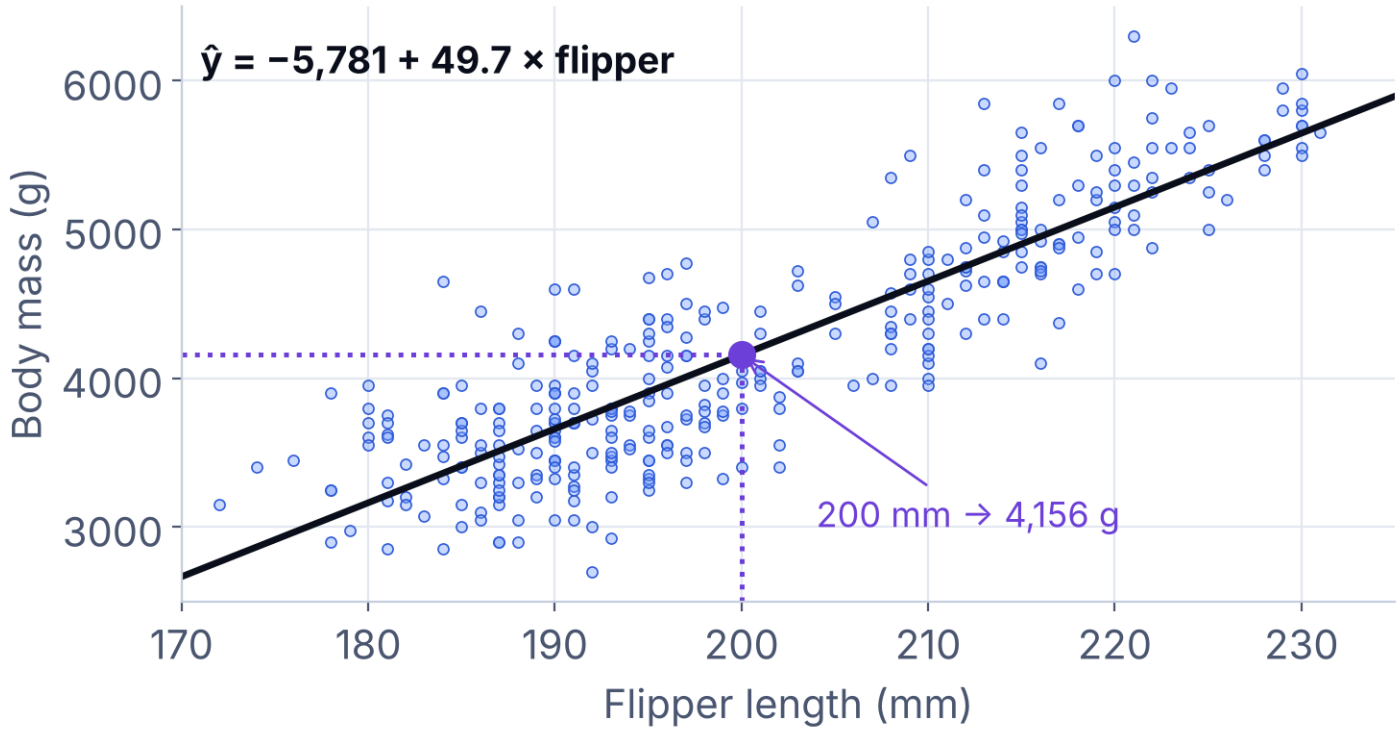


Figure 2. Palmer penguins (n = 342). Line fitted by LinearRegression (scikit-learn 1.9.1).

WHAT TO NOTICE

- Slope: +49.7 g per mm of flipper
- Prediction at 200 mm: 4,156 g
- The line follows the trend; individual penguins scatter around it

Reading the two coefficients

Slope b_1 = 49.7 g per mm

- One more millimetre of flipper is associated with about 50 g more predicted mass
- 10 mm more: about 497 g more
- Say "associated with", not "causes"

Intercept b_0 = –5,781 g

- The predicted mass for a flipper of 0 mm
- Impossible: no penguin has 0 mm flippers
- It positions the line; it has no meaning outside the data (172 to 231 mm)

Predicting far outside the data

✗ INCORRECT APPROACH

Use the penguin line to predict a 100 mm flipper: −5,781 + 49.7 × 100 = −811 g

✓ CORRECTION

Predict only inside the range seen in training (172 to 231 mm). Outside it, collect data or say the model does not apply.

WHY IT FAILS

A line is only an approximation where data exists. Outside that range, nothing tells us the relationship stays straight.

Knowledge check

QUESTION

Penguin A has flippers 10 mm longer than penguin B. With slope 49.7 g per mm, how much heavier is A predicted to be?

A

About 50 g

B

About 5,781 g

C

About 500 g

D

About 4,156 g

03

CHAPTER

Measuring errors

Three numbers that say how wrong the predictions are, measured on unseen data.

MAE and RMSE: the typical size of an error

$$MAE = \Sigma\ |y_i - \hat{y}_i|\ /\ n \qquad RMSE = \sqrt{(\Sigma\ (y_i - \hat{y}_i)^2\ /\ n)}$$

MAE (LEFT)

Mean absolute error, in the unit of y

"On average, the prediction is off by …"

Five students: 3.6 / 5 = 0.72 point

RMSE (RIGHT)

Like a standard deviation of the errors

Large errors weigh more; RMSE ≥ MAE

Five students: √(3.2 / 5) = 0.80 point

R^2: the share of variation explained

$$R^2 = 1 - \Sigma (y_i - \hat{y}_i)^2 \,/\, \Sigma (y_i - \bar{y})^2$$

READING R^2

1: perfect predictions

0: no better than predicting the mean $\bar{y}$

< 0 on a test set: worse than the mean

LINK TO SESSION 01

Simple regression, training data: $R^2 = r^2$

Penguins: $0.871^2 = 0.759$. Flipper length explains about 76% of the variation in mass.

Evaluating on the test set

CODE · PYTHON

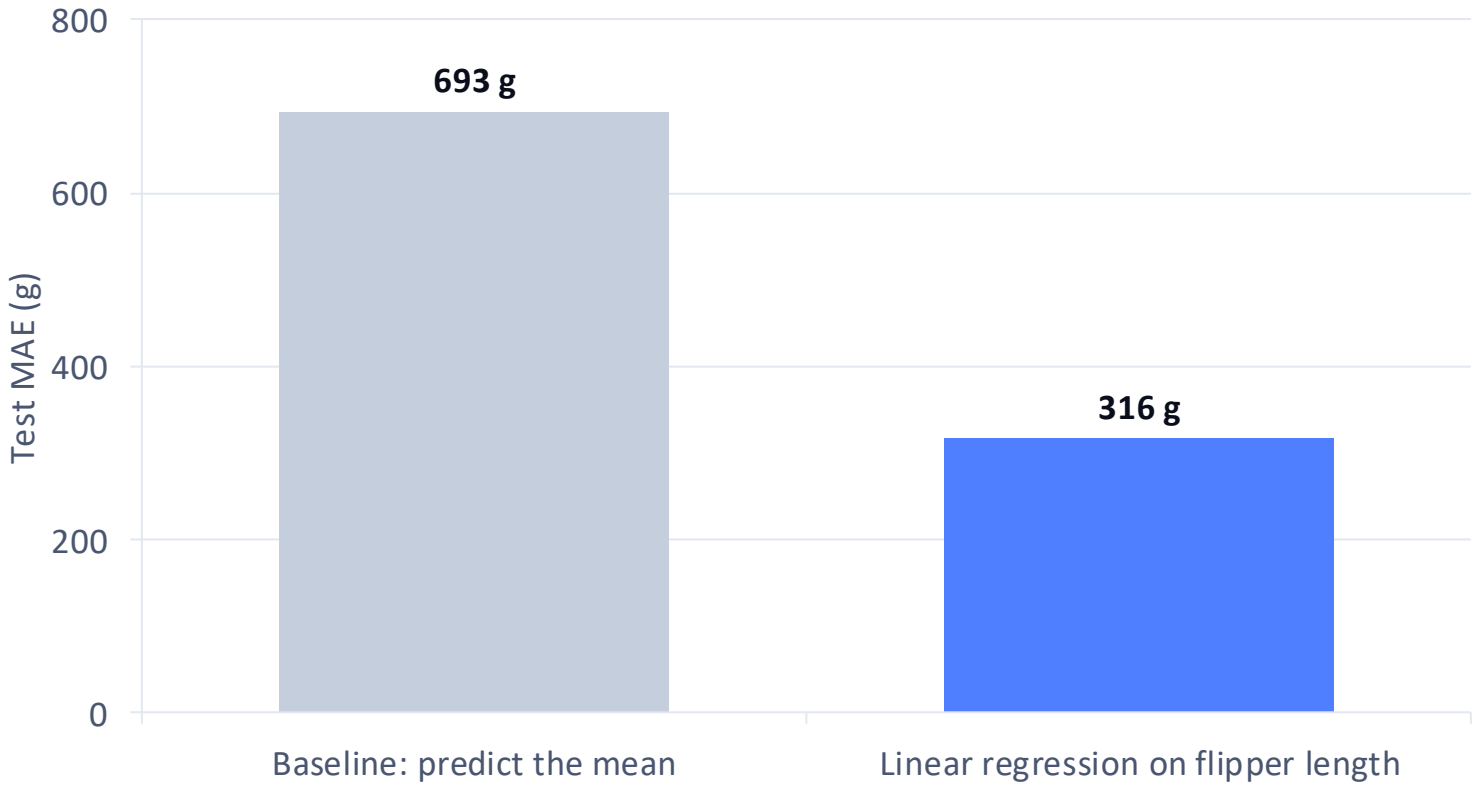
```python
from sklearn.metrics import (
    mean_absolute_error,
    root_mean_squared_error, r2_score)

train, test = train_test_split(
    df, test_size=0.2, random_state=42)
lr.fit(train[["flipper_length_mm"]],
       train["body_mass_g"])
pred = lr.predict(test[["flipper_length_mm"]])
y_test = test["body_mass_g"]
print(mean_absolute_error(y_test, pred))
print(root_mean_squared_error(y_test, pred))
print(r2_score(y_test, pred))
```

OUTPUT

```
316
381
0.782
```

Rounded. 273 training and 69 test penguins.

Always compare with a baseline



Test set of 69 penguins. Baseline: DummyRegressor(strategy="mean"). scikit-learn 1.9.1.

MAIN FINDING

The line cuts the typical error from 693 g to 316 g.

Baseline $R^2 \approx 0$; model $R^2 = 0.78$.

Break · 15 minutes

After the break: what if we use more than one feature?

04

CHAPTER

Several features

One coefficient per feature, each read "all else being equal".

Multiple linear regression

$$\hat{y} = b_0 + b_1 x_1 + b_2 x_2 + \ldots + b_p x_p$$

SYMBOLS

$x_1 \ldots x_p$: the p features

b_j : coefficient of feature j

All coefficients chosen together by least squares

INTUITION

b_j = change in $\hat{y}$ when x_j increases by 1, all else being equal.

Adding a feature can change the other coefficients.

A category in a linear model: dummy variables

species	species_Chinstrap	species_Gentoo	Meaning
Adelie	0	0	Reference category
Chinstrap	1	0	Difference Chinstrap vs Adelie
Gentoo	0	1	Difference Gentoo vs Adelie

Table 1. OneHotEncoder(drop="first"): Adelie becomes the reference; each coefficient is a difference from Adelie.

Flipper length and species together

CODE · PYTHON

```python
cols = ["flipper_length_mm", "species"]
prep = ColumnTransformer([
    ("num", "passthrough",
        ["flipper_length_mm"]),
    ("cat", OneHotEncoder(drop="first"),
        ["species"])])
model = make_pipeline(prep, LinearRegression())

model.fit(train[cols], train["body_mass_g"])
print(model[-1].intercept_)
print(model[-1].coef_)
print(r2_score(y_test,
    model.predict(test[cols])))
```

OUTPUT

```
-3623.3
[ 38.5 -173.9 319.1]
0.8
```

Order: flipper, Chinstrap, Gentoo. Rounded.

Reading the model with species

PROBLEM

Model: $\hat{y} = -3{,}623.3 + 38.5 \times \text{flipper} - 173.9 \times \text{Chinstrap} + 319.1 \times \text{Gentoo}$. Predict the mass of a penguin with 200 mm flippers for each species.

REASONING

1. Adelie (reference): $-3{,}623.3 + 38.5 \times 200 = 4{,}076.7$
2. Chinstrap: $4{,}076.7 - 173.9 = 3{,}902.8$
3. Gentoo: $4{,}076.7 + 319.1 = 4{,}395.8$
4. The flipper slope fell from 48.8 to 38.5 g/mm once species was added

RESULT

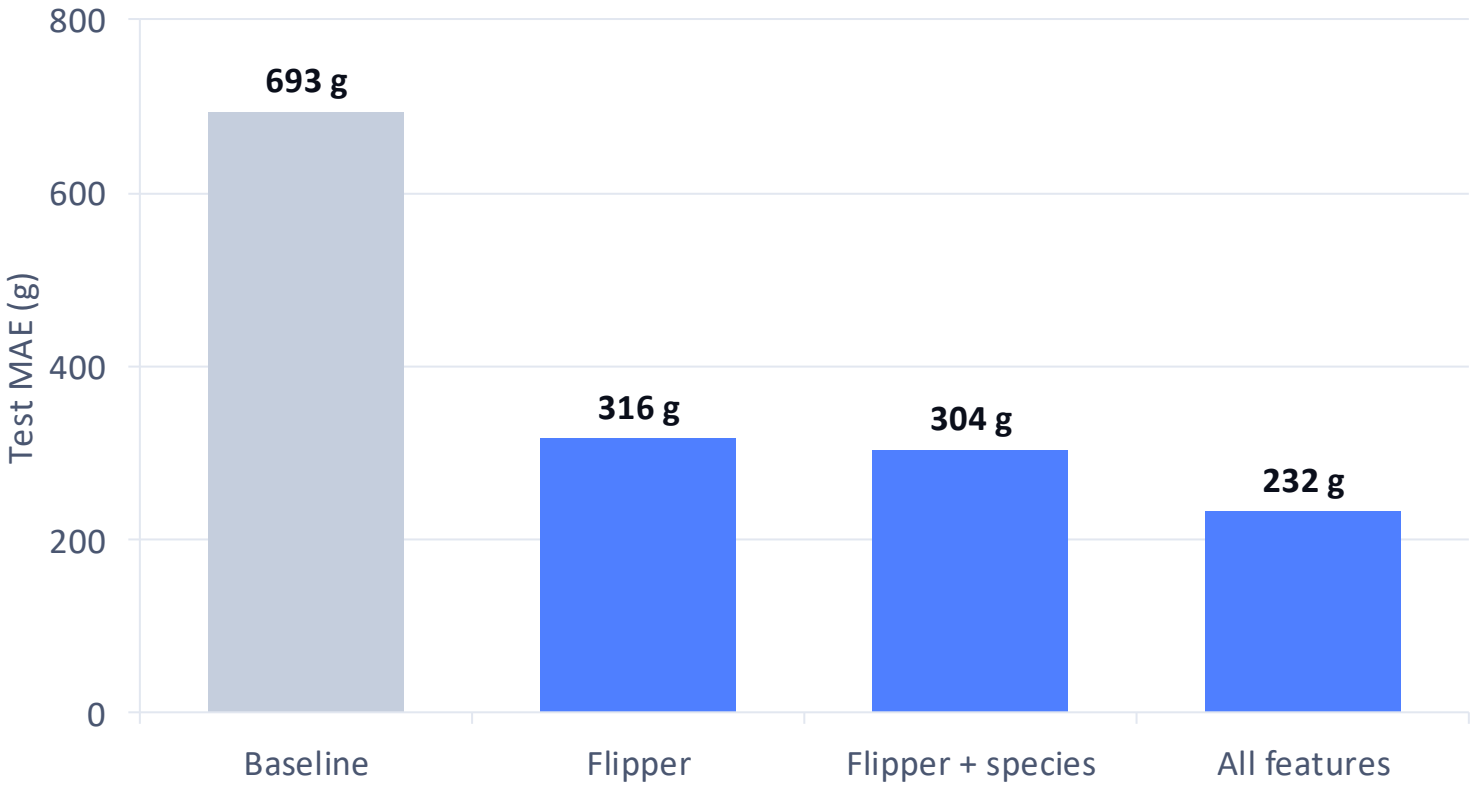
Adelie ≈ 4,077 g

Chinstrap ≈ 3,903 g

Gentoo ≈ 4,396 g

Same slope, shifted lines

More informative features, smaller errors



Same 69 test penguins. "All features": bill length, bill depth, flipper, species, island, sex, with the Session 02 pipeline.

MAIN FINDING

Each informative feature lowers the error: 316 → 304 → 232 g.

Test R^2: 0.78 → 0.80 → 0.88.

Reading a big coefficient as a big effect

✗ INCORRECT APPROACH

"Gentoo has a coefficient of 319, flipper only 38.5, so species matters 8 times more"

✓ CORRECTION

Compare coefficients only after standardising the numerical features, and read them as associations, not causes

WHY IT FAILS

A coefficient depends on the unit of its feature: in metres, the flipper coefficient would be 38,500 g per metre. Same data, different number.

05

CHAPTER

Checking the model

A model can produce numbers even when a straight line is the wrong shape.
The residuals tell us.

What linear regression assumes

- The relationship is roughly a straight line
- The errors have about the same size for small and large x
- The errors are independent of each other (careful with time series)
- No extreme outlier pulls the line on its own

KEY TAKEAWAY

Check these conditions with a residual plot: residuals against x, or against the prediction.

Reading a residual plot

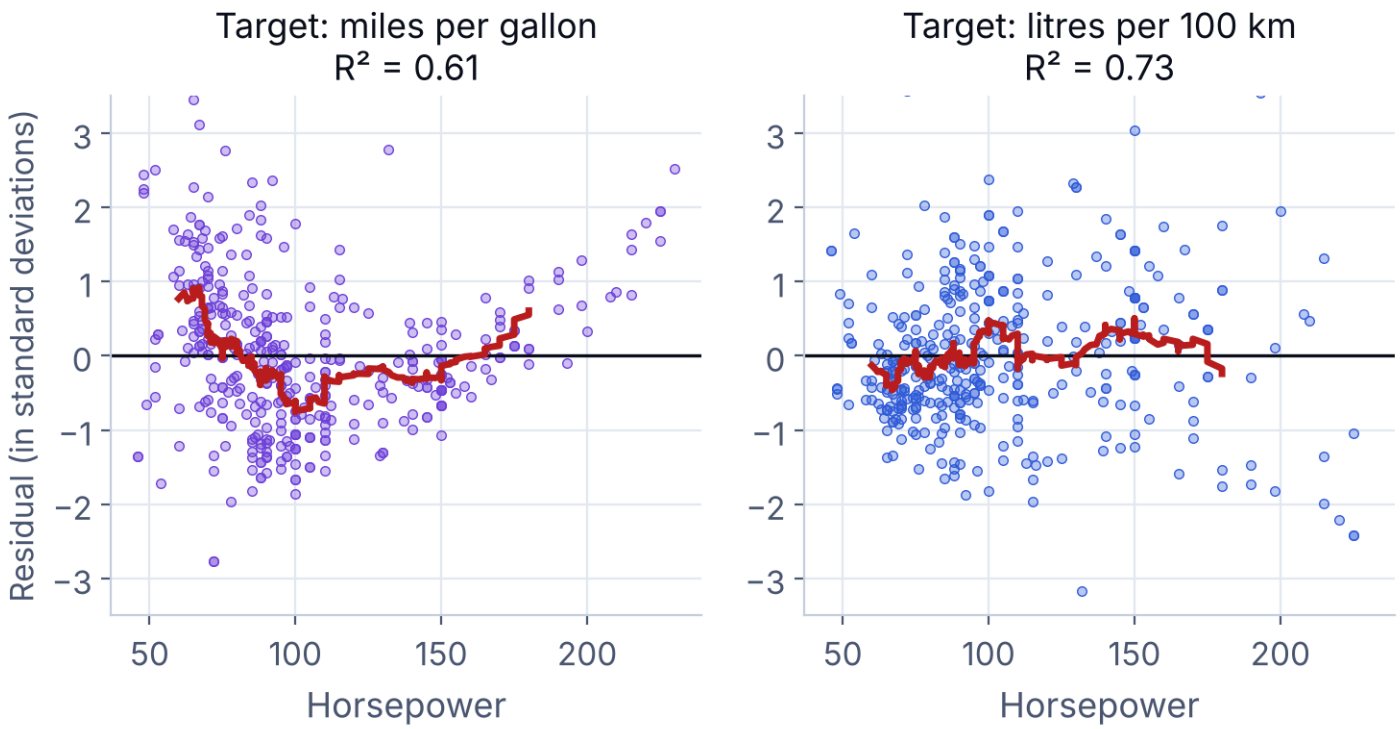


Figure 3. Auto MPG data (392 cars). Red: moving average of the residuals. Residuals divided by their standard deviation.

WHAT TO NOTICE

- Left: a U-shape. The straight line is systematically wrong
- Right: litres per 100 km. No pattern left
- Same feature: R^2 0.61 → 0.73
- Good residuals look like random noise around 0

Overfitting check: training score vs test score

Penguin models

- Flipper only: R^2 0.75 (train) vs 0.78 (test)
- All features: R^2 0.87 (train) vs 0.88 (test)
- Close scores: no sign of overfitting

When to worry

- Training score much higher than test score
- Many features compared with the number of rows
- Then use fewer features or regularisation

Looking ahead: regularised regression

LinearRegression

CRITERION

Least squares only

EFFECT

Keeps every coefficient

PENGUINS, TEST

R^2 0.876

Ridge

CRITERION

Least squares + $\Sigma\, b_j^2$

EFFECT

Shrinks all coefficients

PENGUINS, TEST

R^2 0.881 (alpha = 1)

Lasso

CRITERION

Least squares + $\Sigma\, |b_j|$

EFFECT

Sets some to exactly 0

PENGUINS, TEST

R^2 0.876; 4 of 11 at 0 (alpha = 10)

All three use the scaled all-features pipeline of Session 02. scikit-learn: Ridge(alpha=...), Lasso(alpha=...).

06

LAB

Lab: fuel consumption of cars

Predict the fuel consumption of a car from its weight, power, year and origin.

LAB · DATA

The Auto MPG dataset

Column	Meaning	Type	Notes
`mpg`	Miles per US gallon	Numerical	Convert: L/100 km = 235.215 / mpg (target)
`weight`	Weight in pounds	Numerical	Convert: kg = pounds × 0.4536
`horsepower`	Engine power	Numerical	6 missing values
`model_year`	Year of the model, 70 to 82	Numerical	1970 to 1982
`origin`	usa, japan, europe	Categorical	One-hot encode

Table 2. Auto MPG (398 cars, 1970 to 1982). Quinlan (1993), UCI Machine Learning Repository; available as seaborn's "mpg" dataset.

LAB · GUIDED CODE

Load, convert, split

CODE · PYTHON

```python
cars = sns.load_dataset("mpg")
cars["l_per_100km"] = 235.215 / cars["mpg"]
cars["weight_kg"] = cars["weight"] * 0.4536

train, test = train_test_split(
    cars, test_size=0.2, random_state=42)
print(len(train), len(test))
print(train["horsepower"].isna().sum())
```

OUTPUT

```
318 80
5
```

LAB · GUIDED CODE

Baseline and a first model

CODE · PYTHON

```python
from sklearn.dummy import DummyRegressor

y_tr = train["l_per_100km"]
y_te = test["l_per_100km"]
f = ["weight_kg"]
base = DummyRegressor().fit(train[f], y_tr)
lr = LinearRegression().fit(train[f], y_tr)
for m in (base, lr):
    p = m.predict(test[f])
    print(mean_absolute_error(y_te, p),
          r2_score(y_te, p))
print(lr.coef_)
```

OUTPUT

```
3.14 -0.001
1.25 0.805
[0.00902]
```

Rounded. Baseline first, then weight.

LAB · EXERCISE

Lab · Model fuel consumption

GOAL

Model l_per_100km with linear regression. One sentence of interpretation per question.

TIME

30 min

INSTRUCTIONS

1. Interpret the weight slope per 100 kg; predict for 1,200 kg
2. Add horsepower, model_year, origin with a pipeline; report MAE, RMSE, R^2
3. Interpret the model_year coefficient, all else equal
4. Interpret the origin coefficients (which is the reference?)
5. Plot residuals against predictions: any pattern?
6. Can you predict for a 3,000 kg vehicle? Why?

DELIVERABLE

Your notebook, with code and one sentence per question.

Bring it to the Week 4 practice session.

Exit question

QUESTION

A model with 120 features, trained on 150 houses, has $R^2 = 0.80$ on its training data. What can we conclude?

A

80% of its predictions are exactly right

B

The features cause the price

C

Its errors are below 20% of the price

D

Nothing yet: check the test score, overfitting is likely

Recap

KEY TAKEAWAYS

- Least squares picks the line with the smallest Σe^2
- Slope = change in $\hat{y}$ per unit of x; $b_1 = r \times s(y) / s(x)$
- Report MAE or RMSE (units) and R^2 on the test set, against a baseline
- Read coefficients "all else equal", with units; not as causes
- Check residuals; do not extrapolate

NEXT

- Session 04: logistic regression, predicting a class
- Week 4 practice: finish the lab notebook
- Read: ISLP, chapter 3

References

(1) James, G., Witten, D., Hastie, T., Tibshirani, R., & Taylor, J. (2023). An Introduction to Statistical Learning, with Applications in Python, ch. 3 (Linear Regression). Springer. statlearning.com

(2) Diez, D., Çetinkaya-Rundel, M., & Barr, C. D. (2019). OpenIntro Statistics (4th ed.), ch. 8 (Introduction to linear regression). openintro.org/book/os

(3) scikit-learn 1.9 User Guide: 1.1 Linear Models (ordinary least squares, Ridge, Lasso); 3.4 Metrics (regression metrics). scikit-learn.org/stable

(4) Quinlan, R. (1993). Auto MPG [Dataset]. UCI Machine Learning Repository. doi.org/10.24432/C5859H

(5) Horst, A. M., Hill, A. P., & Gorman, K. B. (2020). palmerpenguins: Palmer Archipelago (Antarctica) penguin data. doi.org/10.5281/zenodo.3960218

SH

Thank you · Questions?

Next session: logistic regression. Bring your lab notebook to the Week 4 practice.

Dr. Soufiane Hourri, PhD, HDR

drsoufiane.com