LOGISTIC REGRESSION II

MEASURES OF FIT, CLASSIFICATION, PERFORMANCE METRICS, R IMPLEMENTATION

DEVIANCE, AND MODEL FIT

- **Deviance** plays a role similar to "sum of squared errors" in linear regression
  - A numerical measure of how *badly* the model fits the data
  - Smaller deviance → better fit, just like smaller sum of squared errors (SSE) in linear regression
  - Linear regression: Uses SSE = $\Sigma(\text{observed} - \text{predicted})^2$ as a “badness” measure
  - Logistic regression: Uses **deviance** as a “badness” measure
    - Based on *how unlikely* the observed 0/1 outcomes are under the model’s predicted probabilities
- From a probabilistic perspective
  - The model assigns a probability to each outcome (e.g., P(default = 1 I predictors))
  - If the model gives **high probability** to what actually happened (0 or 1) → lower deviance
  - If the model often gives **low probability** to what actually happened → higher deviance
  - EXAMPLE: Predicts the **same probability** of default for everyone (e.g., “30% chance of default” for all loans).

MCFADDEN'S R^2 AND OTHER PSEUDO R^2 MEASURES

- No single R^2 ! (such as in linear regression)
  - In linear models, we have a clean R^2 = % of variance explained
  - In logistic regression, Y is 0/1 and the model is probabilistic, so that exact notion doesn't carry over
  - Instead, we use "pseudo R^2" measures that try to play a similar *summary* role
- McFadden's R^2 (most common pseudo R^2)
  - $$R^2{}_{\text{McF}} = 1 - \frac{\text{log-likelihood\_full}}{\text{log-likelihood\_null}}$$
    - $$\ell = \sum_{i=1}^{n} \left[y_i \log\left(p_i\right) + \left(1 - y_i\right) \log\left(1 - p_i\right) \right]$$
  - log-likelihood_null: model with only an intercept (everyone gets the same default probability)
  - log-likelihood_full: model with predictors (e.g., credit_score, loan_amount, prior_default…)

MCFADDEN'S R^2: INTERPRETATION

- $$R^2_{\text{McF}} = 1 - \frac{\text{log-likelihood\_full}}{\text{log-likelihood\_null}}$$
- Interpretation
  - If the full model is much more likely (higher log-likelihood) than the null model:
    - The ratio is small
    - McFadden's R^2 is closer to 1 **(**better fit)
- If the full model barely improves on the null model, McFadden's R^2 stays close to 0

HYPOTHESIS TESTS FOR COEFFICIENTS

- As with Linear Regression ….
- For each coefficient β_j, we will test
  - Null hypothesis H_0: $\beta_j = 0$
  - Alternate hypothesis H_A: $\beta_j \neq 0$
- R reports z-statistics and p-values based on large-sample theory.
- Small p-values suggest the predictor is associated with the outcome,after adjusting for other predictors in the model.

FROM PROBABILITIES TO **CLASSIFICATIONS**

- Logistic regression outputs estimated probabilities $\hat{p}(x)$
- To classify, we choose a **threshold** c
  - Often, c = 0.5
- If $\hat{p}(x) \geq c$, predict Y = 1; otherwise predict Y = 0
- Changing the threshold trades off false positives vs. false negatives
  - We will see how a few slides later

CONFUSION MATRIX

- A confusion matrix compares predicted classes with actual outcomes
- Four cells:
  - True Positives (TP)
  - False Positives (FP)
  - True Negatives (TN)
  - False Negatives (FN)
- Suppose we have 100 loan applicants
  - Ground truth (actual outcomes): 30 actually defaulted (Y=1), 70 did not default (Y=0).
  - A fitted logistic regression model (using credit score, loan amount, prior default, etc.) gives us:
    - Correctly flags **25** defaulters as default (**TP**), misses **5** defaulters (**FN**)
    - Incorrectly flags **10** non-defaulters as default (**FP**), and correctly classifies **60** as no default (**TN**)

Predicted Actual	Will default	Not default
Will default	25 (TP)	5 (FN)
Not default	10 (FP)	60 (TN)

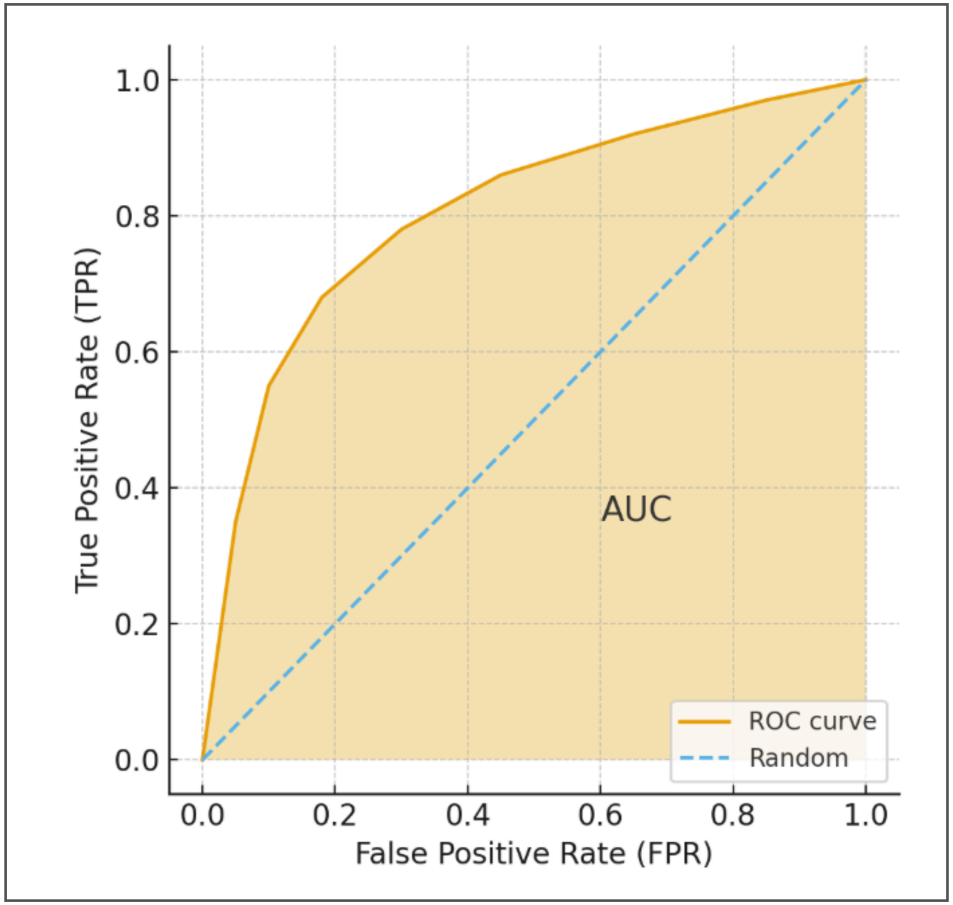
PERFORMANCE METRICS
(FOR CLASSIFICATION)

Predicted / Actual	Will default	Not default
Will default	**25 (TP)**	**5 (FN)**
Not default	**10 (FP)**	**60 (TN)**

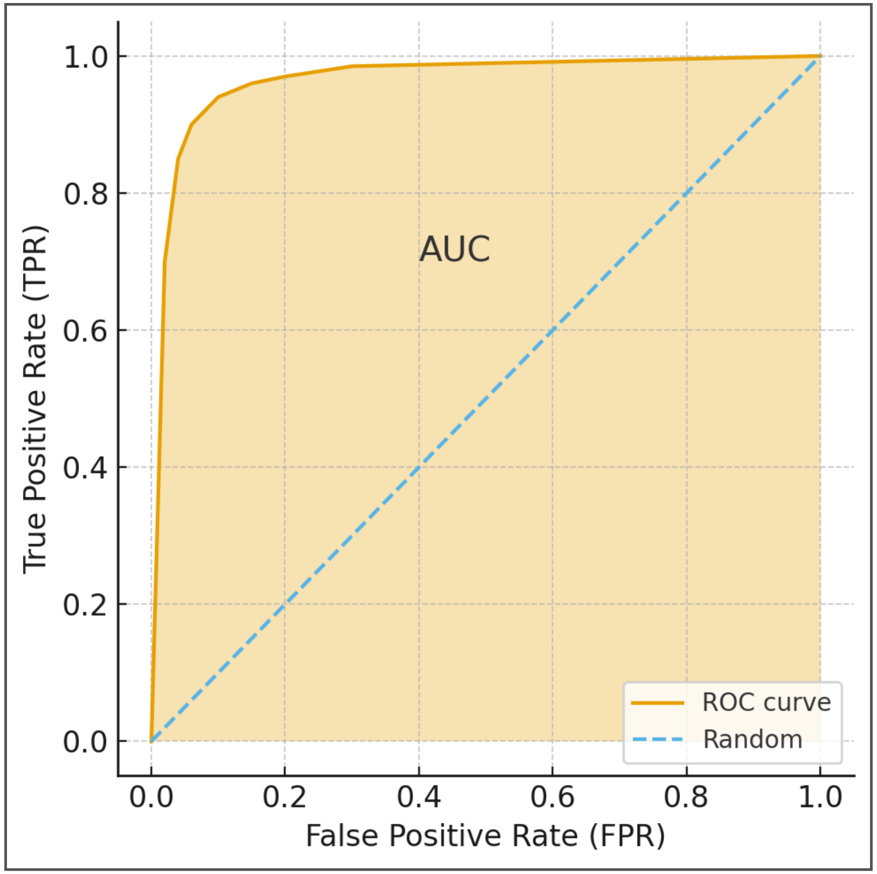
- **Accuracy:** what fraction did we get **correct**
  - (TP + TN) / (Total = TP + FP + TN + FN) = (25+60)/100 = 0.85
- **Sensitivity (Recall)**: how often we catch true positives
  - TP / (TP + FN) = 25/(25+5) = 0.83
- **Specificity:** how often we correctly identify true negatives
  - TN / (TN + FP) = 60/(60+10) = 0.86

THE **ROC CURVE**

- **ROC curve:** plot of (sensitivity) vs. (1 – specificity) as the *threshold* varies
  - **Threshold** (for ROC): The threshold is the cutoff on the *predicted probability of default* above which we label a case as default (1) and below which we label it as no default (0)
  - **True Positive Rate (TPR):** Among all loans that *actually defaulted*, the proportion that the model correctly labels as default. (Sensitivity / recall.)
  - **False Positive Rate (FPR):** Among all loans that *actually did not default*, the proportion that the model incorrectly labels as default.
- Essentially, ROC shows **the tradeoff** between the true positive rate and the false positive rate
- **Area Under Curve (AUC)**: probability the model ranks a random positive case higher than a random negative case
  - Values closer to 1 indicate better discrimination
  - 0.5 is no better than random guessing



ROC CURVE: **IDEALLY .....**



- If your ROC curve is “hugging the top left corner” = you have a very good model for binary classification !

CHOOSING A CLASSIFICATION THRESHOLD

- Default threshold 0.5 is not always appropriate
- If missing a positive case is very costly, you might lower the threshold
- If false alarms are costly, you might raise the threshold
- The "best" threshold depends on the context and relative costs of errors

MODEL LIMITATIONS AND DIAGNOSTICS

- Logistic regression assumes a linear relationship between predictors and log-odds
- Strong nonlinearity may require transformations or interaction terms
- Separation issues occur if predictors perfectly predict the outcome
  - Let's say *every* loan with `prior_default = 1` defaults, and *no* loan with `prior_default = 0` defaults
  - In that case, logistic regression tries to push the coefficient to $\pm\infty$, in order to get probabilities 0 or 1

EXAMPLE DATASET: LOAN DEFAULT RISK DATA

- We will use a simple, hypothetical dataset called loans_df
- Outcome:
  - default (1 = loan default, 0 = no default)
- Predictors:
  - credit_score, loan_amount, prior_default (1/0), has_coapplicant (1/0)
- Goal: fit a logistic regression model and interpret results

GLIMPSE AT THE DATA

```
> mean(loans_df$default)   # overall default rate
[1] 0.176
> head(loans_df)
  default credit_score loan_amount prior_default has_coapplicant p_default
1       1          639       37900             1               1 0.4306990
2       0          653        8300             0               0 0.1059305
3       0          731       42500             0               0 0.2395770
4       0          682       10500             1               0 0.2709121
5       0          657       26700             0               0 0.1951323
6       0          669       15700             0               0 0.1281965
>
```

FIT A LOGISTIC REGRESSION MODEL

```
fit <- glm(
  default ~ credit_score + loan_amount + prior_default + has_coapplicant,
  data   = loans_df,
  family = binomial
)

# Inspect coefficient estimates, standard errors, z-statistics, and p-values
summary(fit)
```

```
Call:
glm(formula = default ~ credit_score + loan_amount + prior_default +
    has_coapplicant, family = binomial, data = loans_df)

Coefficients:
                  Estimate Std. Error z value Pr(>|z|)
(Intercept)     -3.179e-01  1.801e+00  -0.177  0.85987
credit_score    -4.181e-03  2.622e-03  -1.595  0.11081
loan_amount      4.696e-05  1.080e-05   4.349 1.37e-05 ***
prior_default    1.488e+00  2.740e-01   5.431 5.61e-08 ***
has_coapplicant -8.835e-01  2.944e-01  -3.001  0.00269 **
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 465.27  on 499  degrees of freedom
Residual deviance: 404.02  on 495  degrees of freedom
AIC: 414.02

Number of Fisher Scoring iterations: 5
```

INTERPRETING THE R OUTPUT

- summary(fit) shows:
  - Estimate: $\hat{\beta}$ for each predictor (on the log-odds scale)
  - Std. Error, z value, and Pr(>|z|) for hypothesis tests.
- Also shows null deviance, residual deviance, and AIC for model fit
- AIC (Akaike Information Criterion)
  - For a fitted model
    - $AIC = -2(\text{log-likelihood}) + 2k$,
      - where k = number of estimated parameters (including the intercept)
  - Essentially
    - The **–2 log-likelihood** part rewards **good fit** (higher likelihood → smaller –2 logL)
    - The **+ 2k** part **penalizes model complexity** (more parameters → larger penalty)

COMPUTE **ODDS RATIOS**

```
# Coefficients on the log-odds (logit) scale
coef(fit)

# Convert to odds ratios
or <- exp(coef(fit))
or

# Example: interpret the loan_amount coefficient
or["loan_amount"]
```

```
> # Coefficients on the log-odds (logit) scale
> coef(fit)
   (Intercept)   credit_score    loan_amount  prior_default has_coapplicant
 -3.179299e-01  -4.181242e-03   4.695878e-05   1.488311e+00   -8.834543e-01
>
> # Convert to odds ratios
> or <- exp(coef(fit))
> or
   (Intercept)   credit_score    loan_amount  prior_default has_coapplicant
     0.7276538      0.9958275      1.0000470      4.4296059       0.4133526
>
> # Example: interpret the loan_amount coefficient
> or["loan_amount"]
loan_amount
   1.000047
```

COEFFICIENTS AND ODDS

- credit_score (coef = –0.00418, OR = 0.99583)
  - Each 1-point increase in credit score multiplies the odds of default by 0.9958 (about a 0.4% decrease in the odds), holding other variables fixed
- loan_amount (coef = 0.00004696, OR ≈ 1.000047 per $1)
  - Each extra $1 slightly increases the odds of default (odds × 1.000047).
    More interpretable: an increase of $1,000 multiplies the odds of default by about 1.05 (≈ 5% higher odds), holding other variables fixed
- prior_default (coef = 1.488, OR = 4.43)
  - Borrowers with a prior default have odds of default that are about 4.4 times higher than borrowers without a prior default, holding other variables fixed
- has_coapplicant (coef = –0.883, OR = 0.41)
  - Having a co-applicant multiplies the odds of default by 0.41 (about a 59% reduction in the odds) compared to not having a co-applicant, holding other variables fixed

PREDICTED PROBABILITIES

```
# Get predicted probabilities of default for each loan
loans_df$phat <- predict(fit, type = "response")

# Quick checks
head(loans_df$phat)        # first few predicted probabilities
```

```
> head(loans_df$phat)          # first few predicted probabilities
[1] 0.35316646 0.06546563 0.20122274 0.23358833 0.14049237 0.08487136
```