Statistics and Data Analysis
Unit 04 – Lecture 04 Notes
Polynomial Regression and Logistic Regression

Tofik Ali

February 17, 2026

Topic

Polynomial regression for curvature; logistic regression for classification; basic evaluation metrics.

How to Use These Notes

These notes are written for students who are seeing the topic for the first time. They follow the slide order, but add the missing 'why', interpretation, and common mistakes. If you get stuck, look at the worked exercises and then run the Python demo.

Course repository (slides, demos, datasets): `https://github.com/tali7c/Statistics-and-Data-Analysis`

Time Plan (55 minutes)

- 0–10 min: Attendance + recap of previous lecture
- 10–35 min: Core concepts (this lecture's sections)
- 35–45 min: Exercises (solve 1–2 in class, rest as practice)
- 45–50 min: Mini demo + interpretation of output
- 50–55 min: Buffer / wrap-up (leave 5 minutes early)

Slide-by-slide Notes

Title Slide

State the lecture title clearly and connect it to what students already know. Tell students what they will be able to do by the end (not just what you will cover).

Quick Links / Agenda

Explain the structure of the lecture and where the exercises and demo appear.

- Overview

- Polynomial Regression
- Logistic Regression
- Exercises
- Demo
- Summary

Learning Outcomes

- Explain polynomial features for modeling curvature
- Recognize overfitting risk with high degree
- Write logistic regression probability model (sigmoid)
- Compute precision and recall from a confusion matrix

Why these outcomes matter. **Regression** models a response variable Y as a function of predictor(s) X. It has direction (predictors -> response), produces a fitted equation, and lets you predict and explain. Regression is not automatically causal; causality needs design or strong assumptions. **Logistic regression** models the probability of a binary outcome. Its output is a value between 0 and 1, which you convert to a class label using a threshold (often 0.5, but not always). Use metrics like precision/recall and ROC when classes are imbalanced.

Polynomial Regression: Key Points

- Add features $x, x^2, x^3, \ldots$
- Still linear in parameters
- Choose degree using validation

Explanation. A **parameter** is a fixed (but usually unknown) number that describes the population (e.g., μ, σ). A **statistic** is a number computed from the sample (e.g., $\bar{x}$, s). Statistics vary from sample to sample, which is why we talk about uncertainty. **Regression** models a response variable Y as a function of predictor(s) X. It has direction (predictors -> response), produces a fitted equation, and lets you predict and explain. Regression is not automatically causal; causality needs design or strong assumptions.

Logistic Regression: Key Points

- Outputs probability in (0,1)
- Threshold converts probability to class label
- Evaluate using confusion matrix / ROC

Explanation. **Regression** models a response variable Y as a function of predictor(s) X. It has direction (predictors -> response), produces a fitted equation, and lets you predict and explain. Regression is not automatically causal; causality needs design or strong assumptions. **Logistic regression** models the probability of a binary outcome. Its output is a value between 0 and 1, which you convert to a class label using a threshold (often 0.5, but not always). Use metrics like precision/recall and ROC when classes are imbalanced. An **ROC curve** plots true positive rate vs false positive rate across all thresholds. It helps compare classifiers without committing to a single threshold. **AUC** summarizes the ROC curve: higher AUC generally indicates better ranking of positives above negatives.

Logistic Regression: Key Formula

$$P(y = 1 \mid x) = \frac{1}{1 + e^{-(\beta_0 + \beta^T x)}}$$

How to read the formula. **Regression** models a response variable Y as a function of predictor(s) X. It has direction (predictors -> response), produces a fitted equation, and lets you predict and explain. Regression is not automatically causal; causality needs design or strong assumptions. **Logistic regression** models the probability of a binary outcome. Its output is a value between 0 and 1, which you convert to a class label using a threshold (often 0.5, but not always). Use metrics like precision/recall and ROC when classes are imbalanced.

Exercises (with Solutions)

Attempt the exercise first, then compare with the solution. Focus on interpretation, not only arithmetic.

Exercise 1: Polynomial features

For degree-2 polynomial, what features do we use from x?

Solution

- Use $1, x, x^2$ (intercept + linear + quadratic).

Walkthrough. The **intercept** is the predicted value when $X = 0$. It may or may not be meaningful depending on whether $X = 0$ is realistic in your context. If $X = 0$ is outside the observed range, do not over-interpret the intercept.

Exercise 2: Precision/recall

TP=30 FP=10 FN=20 TN=40. Compute precision and recall.

Solution

- Precision=30/(30+10)=0.75
- Recall=30/(30+20)=0.60

Walkthrough. **Precision** answers: of all predicted positives, what fraction were truly positive? **Recall** answers: of all actual positives, what fraction did we catch? Improving precision often reduces recall and vice versa, so choose based on cost of false positives vs false negatives.

Exercise 3: Threshold effect

If threshold increases from 0.5 to 0.8, what tends to happen to precision and recall?

Solution

- Precision often increases, recall often decreases.

Walkthrough. **Precision** answers: of all predicted positives, what fraction were truly positive? **Recall** answers: of all actual positives, what fraction did we catch? Improving precision often reduces recall and vice versa, so choose based on cost of false positives vs false negatives.

Mini Demo (Python)

Run from the lecture folder:

```
python demo/demo.py
```

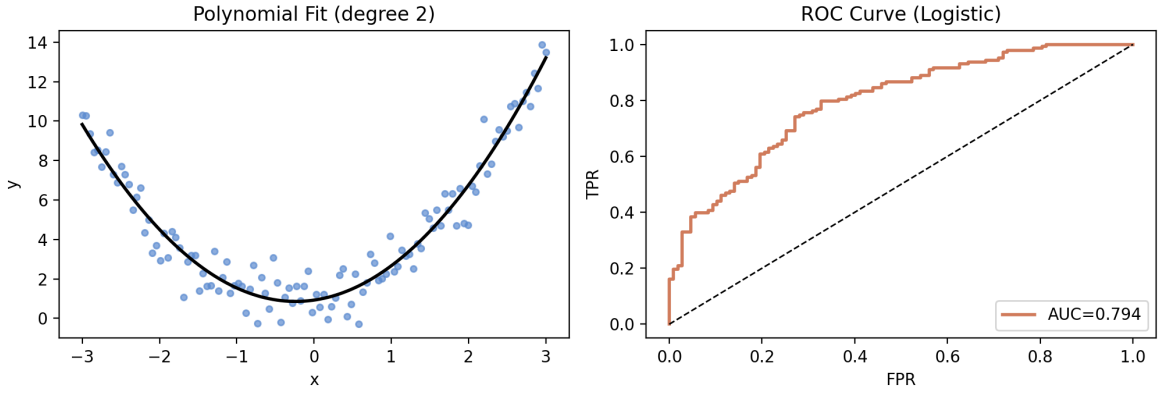
Output files:

- `images/demo.png`
- `data/results.txt`

What to show and say.

- Demonstrates polynomial regression (curvature) and a simple logistic classification boundary.
- Use it to discuss overfitting risk with higher-degree polynomials.
- Connect classification output to precision/recall trade-offs.

Demo Output (Example)



Summary

- Key definitions and the main formula.
- How to interpret results in context.
- How the demo connects to the theory.

Exit Question

Why is ROC curve useful when classes are imbalanced?

Suggested answer (for revision). ROC curves compare performance across thresholds and are less sensitive to class imbalance than accuracy at one threshold.

References

- Montgomery, D. C., & Runger, G. C. *Applied Statistics and Probability for Engineers*, Wiley.
- Devore, J. L. *Probability and Statistics for Engineering and the Sciences*, Cengage.
- McKinney, W. *Python for Data Analysis*, O'Reilly.

Appendix: Slide Deck Content (Reference)

The material below is a reference copy of the slide deck content. Exercise solutions are explained in the main notes where applicable.

Title Slide

https://github.com/tali7c/Statistics-and-Data-Analysis

Quick Links

Overview Polynomial Regression Logistic Regression Exercises Demo Summary

Agenda

- Overview
- Polynomial Regression
- Logistic Regression
- Exercises
- Demo
- Summary

Learning Outcomes

- Explain polynomial features for modeling curvature
- Recognize overfitting risk with high degree
- Write logistic regression probability model (sigmoid)
- Compute precision and recall from a confusion matrix

Polynomial Regression: Key Points

- Add features $x, x^2, x^3, \ldots$
- Still linear in parameters
- Choose degree using validation

Logistic Regression: Key Points

- Outputs probability in (0,1)
- Threshold converts probability to class label
- Evaluate using confusion matrix / ROC

Logistic Regression: Key Formula

$$P(y = 1 \mid x) = \frac{1}{1 + e^{-(\beta_0 + \beta^T x)}}$$

Exercise 1: Polynomial features

For degree-2 polynomial, what features do we use from x?

Solution 1

- Use $1, x, x^2$ (intercept + linear + quadratic).

Exercise 2: Precision/recall

TP=30 FP=10 FN=20 TN=40. Compute precision and recall.

Solution 2

- Precision=30/(30+10)=0.75
- Recall=30/(30+20)=0.60

Exercise 3: Threshold effect

If threshold increases from 0.5 to 0.8, what tends to happen to precision and recall?

Solution 3

- Precision often increases, recall often decreases.

Mini Demo (Python)

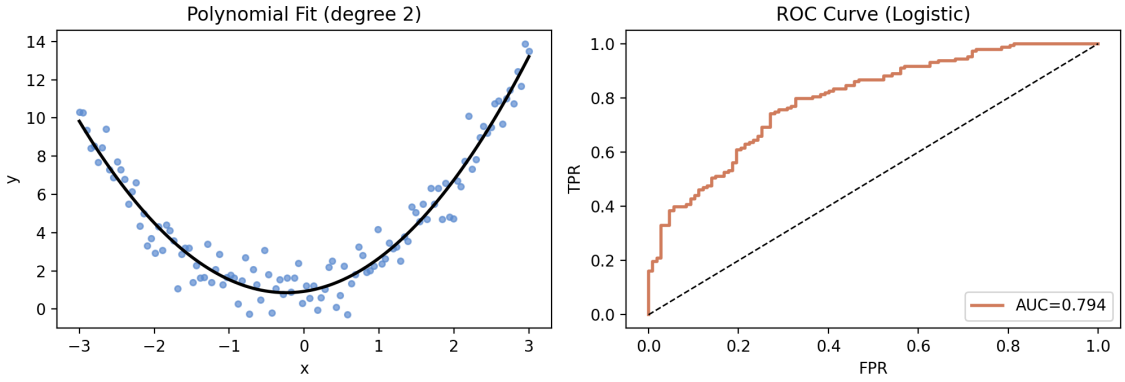
Run from the lecture folder:

```
python demo/demo.py
```

Outputs:

- `images/demo.png`
- `data/results.txt`

Demo Output (Example)



Summary

- Key definitions and the main formula.
- How to interpret results in context.
- How the demo connects to the theory.

Exit Question

Why is ROC curve useful when classes are imbalanced?