PODS Lab 13: Supervised Learning, Classification, & Logistic Regression

Hamza Alshamy

Center for Data Science, NYU
ha2486@nyu.edu

04/25/25

1. Logistic Regression
2. Classification Metrics

Motivating Example: Loan Default Prediction

Scenario: You are a bank predicting whether a loan applicant will **default** ($y \in \{0, 1\}$) based on their **score** (x).

Motivating Example: Loan Default Prediction

Scenario: You are a bank predicting whether a loan applicant will **default** ($y \in \{0, 1\}$) based on their **score** (x).

The **score** combines factors like:

- Credit history, income, past repayments, etc.

Motivating Example: Loan Default Prediction

Scenario: You are a bank predicting whether a loan applicant will **default** ($y \in \{0, 1\}$) based on their **score** (x).

The **score** combines factors like:

- Credit history, income, past repayments, etc.

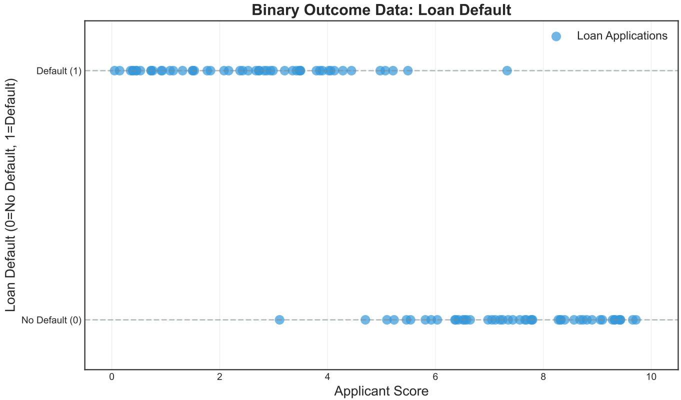
Idea:

- **High score:** $\Rightarrow$ stronger evidence the applicant **will not default**.
- **Low score:** $\Rightarrow$ greater risk of **default**.
- In the middle, we have **uncertainty** – we model this as a probability.

Goal: Use a model to map the score to a **probability of default**:

$$P(\text{default} \mid \text{score}) = \text{some function of the score}$$

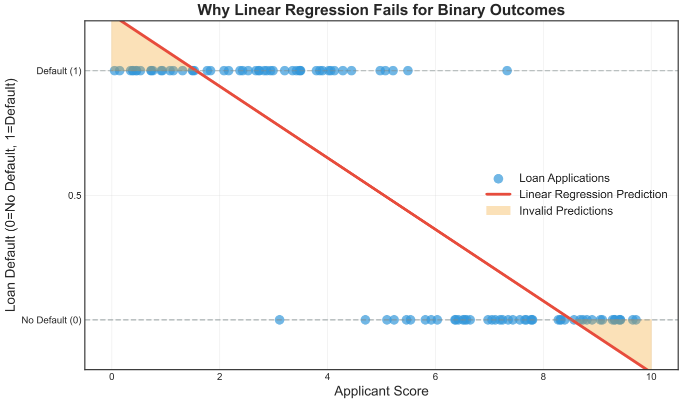
Visualizing Example: Loan Approval



Observation:

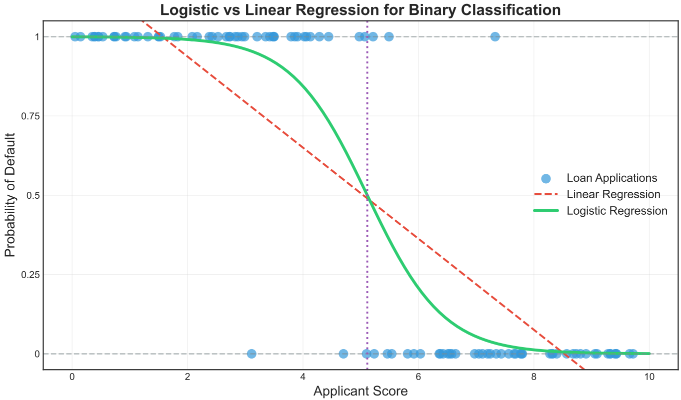
- **Outcome** is binary: default (1) or non-default (0).
- The score is continuous, so we need a model that maps it to a probability.

Visualizing Example: Linear Regression doesn't cut it



- **Unbounded Predictions:** Linear regression can output values below 0 or above 1 – invalid predictions.
- **Linearity Assumption:** Assumes each unit increase in score has the same effect on the outcome.

Visualizing Example: Logistic Regression



- **Non-linear:** When score = 5, Little changes in the score have big impacts on our prediction.
- **Bounded:** Predictions stay between 0 and 1 – perfect for modeling probabilities.

Logistic Regression

Key Characteristics:

- **Supervised learning**: learns from labeled data.
- **Classification task**: predicts discrete outcomes (e.g., default vs non-default).
- **Goal**: estimate the *probability* of a class given input features.

Logistic Regression

Key Characteristics:

- **Supervised learning**: learns from labeled data.
- **Classification task**: predicts discrete outcomes (e.g., default vs non-default).
- **Goal**: estimate the *probability* of a class given input features.

Main Idea:

Logistic regression maps continuous predictors to discrete outcomes by applying the **logistic (sigmoid) function**.

Logistic Regression

Key Characteristics:

- **Supervised learning**: learns from labeled data.
- **Classification task**: predicts discrete outcomes (e.g., default vs non-default).
- **Goal**: estimate the *probability* of a class given input features.

Main Idea:

Logistic regression maps continuous predictors to discrete outcomes by applying the **logistic (sigmoid) function**.

1. When **score is in the middle:** Little changes in these core have big impacts on our probability.
    - Moving from score 5 to 4 is a *huge* change in relative terms.

Logistic Regression

Key Characteristics:

- **Supervised learning**: learns from labeled data.
- **Classification task**: predicts discrete outcomes (e.g., default vs non-default).
- **Goal**: estimate the *probability* of a class given input features.

Main Idea:

Logistic regression maps continuous predictors to discrete outcomes by applying the **logistic (sigmoid) function**.

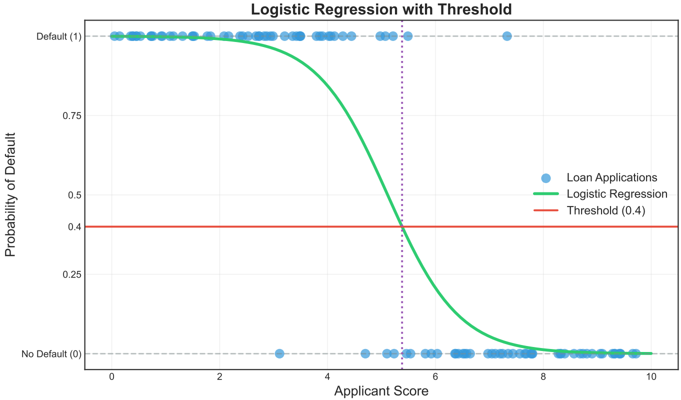
1. When **score is in the middle:** Little changes in these core have big impacts on our probability.
   - Moving from score 5 to 4 is a *huge* change in relative terms.
2. When **score is large:** When the score is 9, I have very high evidence that he will not default.
   - So, jumping from 9 to 10 does not change out probability that much.

Classification via Logistic Regression – Thresholding

- Logistic regression outputs a **probability** between 0 and 1.
- **Q:** How do we turn this into a binary prediction?

Classification via Logistic Regression – Thresholding

- Logistic regression outputs a **probability** between 0 and 1.
- **Q:** How do we turn this into a binary prediction?
- **A:** Apply a **threshold** (e.g., classify as 1 if probability > 0.4).



- Any applicant with $P(\text{Default}) > 0.4$ is classified as **default** (1).

From Linear Regression to Logistic Regression

- Logistic regression is built from two key components:
  1. **Linear model:**
  $$z = \beta_0 + \beta_1 x_1 + \cdots + \beta_n x_n$$
  2. **Sigmoid function:**
  $$\sigma(z) = \frac{1}{1 + e^{-z}}$$

From Linear Regression to Logistic Regression

- Logistic regression is built from two key components:
  1. **Linear model:**
  $$z = \beta_0 + \beta_1 x_1 + \cdots + \beta_n x_n$$
  2. **Sigmoid function:**
  $$\sigma(z) = \frac{1}{1+e^{-z}}$$
- Plug the linear model into the sigmoid:

Logistic Regression!

$$P(y = 1 \mid \mathbf{x}) = \frac{1}{1+e^{-(\beta_0+\beta_1 x_1+\cdots+\beta_n x_n)}}$$

Properties of the Sigmoid Function:

- **Bounded:** Always stays between 0 and 1 (Probabilistic Output!).
- **Non-linear:** Captures uncertainty and saturation effects.
- **Continuous.**

Classification Evaluation Metrics

- In linear regression, we evaluate model performance using metrics like:
  1. Root Mean Squared Error (RMSE)
  2. Coefficient of Determination (R^2)
- **But what about classification models, like Logistic Regression?**
- Since the output is categorical (e.g., 0 or 1), we need **different evaluation metrics**.

Classification Metrics from Confusion Matrix

		Actual	
		Positive (1)	**Negative (0)**
Predicted	**Positive (1)**	True Positive (TP)	False Positive (FP)
	Negative (0)	False Negative (FN)	True Negative (TN)

1. **Accuracy:** Proportion of correct predictions.
$$\text{Accuracy} = \frac{TP+TN}{TP+TN+FP+FN}$$
2. **Precision:** Proportion of positive predictions that were actually correct.
$$\text{Precision} = \frac{TP}{TP+FP}$$
3. **Recall (Sensitivity):** Ability to identify all *positive* instances.
$$\text{Recall} = \frac{TP}{TP+FN}$$
4. **Specificity:** Ability to identify all *negative* instances.
$$\text{Specificity} = \frac{TN}{TN+FP}$$

How to Choose the Relevant Metric?

- **Precision:** When the cost of predicting a positive incorrectly is high (e.g., falsely predicting guilt)
- **Recall:** When the cost of missing a positive instance is high (e.g., failing to detect cancer)
- **Specificity:** When the cost of missing a negative instance is high (e.g., When the cost of falsely diagnosing a healthy person as sick is high – prescribing expensive treatment to someone who doesn't need it)

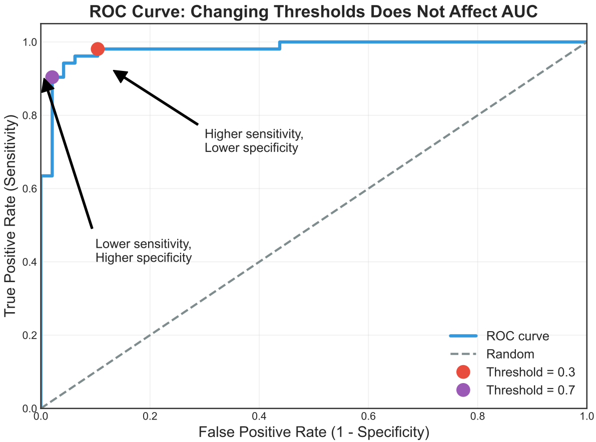
How to Choose the Relevant Metric?

- **Precision:** When the cost of predicting a positive incorrectly is high (e.g., falsely predicting guilt)
- **Recall:** When the cost of missing a positive instance is high (e.g., failing to detect cancer)
- **Specificity:** When the cost of missing a negative instance is high (e.g., When the cost of falsely diagnosing a healthy person as sick is high – prescribing expensive treatment to someone who doesn't need it)

Considerations:

- **Suppose** you are trying to detect fraud.
- Out of 100 cases, only 4 are fraudulent.
- If you predict *all* cases as non-fraud, your **accuracy** is 96%.
- **Accuracy can be misleading in *imbalanced* datasets.**

Area under the Receiver Operator Curve (AUROC)



- The ROC curve plots **True Positive Rate (Recall)** vs. **False Positive Rate**.
- The **(AUC)** measures overall classification performance.