We started with a clear goal: use information about people (age, education, work hours, etc.) to predict whether they earn more than 50K a year, and then understand how well this model works and what the numbers mean in practice.

1. Getting and preparing the data

- **We loaded the Adult Census Income dataset from UCI, which has one row per person and 15 columns describing demographics, work, and income.**

- **We gave the columns readable names like `AGE`, `EDUCATIONNUM`, `HOURSPERWEEK`, `CAPITALGAIN`, `CAPITALLOSS`, and `INCOME`.**

- **We then created a new variable `ABOVE50K` that is 1 if income is `>50K` and 0 if it is `<=50K`, and dropped the original `INCOME` column so the model works with a clean binary target.**

- **When we checked the target distribution, we saw about 24,720 people earning <=50K and 7,841 earning >50K, so roughly 75% vs 25% – this is a class imbalance where low earners dominate the dataset.**

Why this matters: Class imbalance can cause a model to "play it safe" by predicting the majority class most of the time and still get high accuracy, which is misleading if we care about correctly identifying high earners.

2. Creating balanced training and test sets

- **We split the data into two groups: all rows where `ABOVE50K == 1` (high earners) and all rows where `ABOVE50K == 0` (not high earners).**

- **For the training data, we randomly took 70% of the high earners and then sampled the same number of low earners, so the training set has a 50/50 balance between classes.**

- **For the test data, we used the remaining high earners and a matching set of remaining low earners.**

Why we did this: Balancing the training data gives the model a fair chance to learn patterns for both classes instead of being dominated by the majority group, which helps it recognize high earners more reliably.

3. Building the logistic regression model

- **We fitted a logistic regression model where the outcome is `ABOVE50K` and the predictors are:**
    - **`AGE` – older vs younger**
    - **`EDUCATIONNUM` – more vs fewer years of education**
    - **`HOURSPERWEEK` – more vs fewer working hours**
    - **`CAPITALGAIN` – presence and size of investment gains**
    - **`CAPITALLOSS` – presence and size of losses.**
- **The model estimates, for each person, the probability that they earn more than 50K based on these variables.**
- **All five predictors came out highly statistically significant ($p < 2e\text{-}16$), and their coefficients were positive, meaning that higher values for each variable are associated with a higher chance of earning >50K.**
- **When we converted the coefficients to odds ratios using `exp(coef())`, we saw:**
    - **`AGE` odds ratio ≈ 1.052 → every extra year of age multiplies the odds of earning >50K by about 1.05, holding other variables constant.**

- **`EDUCATIONNUM` odds ratio ≈ 1.39 → each extra year of education multiplies the odds by about 1.39, a strong effect.**
- **`HOURSPERWEEK` odds ratio ≈ 1.047 → working more hours per week slightly increases the odds of high income.**
- **`CAPITALGAIN` and `CAPITALLOSS` have odds ratios just above 1, reflecting that any gain or loss is informative but measured on a large numeric scale, so small changes per unit look modest.**

What this tells us: People who are older, more educated, work longer hours, and have non-zero capital gains or losses are more likely to be in the >50K income group according to this model.

4. Predicting on the test set and using a 0.5 cutoff

- **We used the model to predict probabilities for the test set – for each person, we get a value between 0 and 1 representing the model's estimated probability of earning >50K.**
- **We then turned those probabilities into hard class predictions by using a cutoff of 0.5: if `p > 0.5`, predict 1 (>50K), otherwise predict 0 (<=50K).**
- **The confusion matrix with this cutoff looked like this:**
  - **Predicted 0 and actual 0: 14,732 (true negatives)**
  - **Predicted 1 and actual 0: 4,500 (false positives)**
  - **Predicted 0 and actual 1: 659 (false negatives)**
  - **Predicted 1 and actual 1: 1,694 (true positives).**
- **Overall accuracy with this default cutoff was about 76.1%.**

What this tells us: With the 0.5 threshold, the model gets roughly three-quarters of cases correct, but still misclassifies a non-trivial number of both high and low earners.

5. Evaluating with ROC curve and AUC

- **Instead of fixing a single cutoff, we looked at the model's behavior across all possible thresholds using an ROC curve: plotting the true positive rate (sensitivity) against the false positive rate (1 − specificity).**

- **The Area Under this Curve (AUC) summarizes overall discrimination performance; our model produced an AUC of 0.83.**

- **Using common interpretation guidelines, an AUC between 0.8 and 0.9 is considered "good," meaning the model is quite capable of ranking high earners above low earners in probability space.**

What this tells us: Even if accuracy at a specific cutoff doesn't look spectacular, the model is fundamentally good at separating the two classes when looked at over the full range of thresholds.

6. Finding the "best" cutoff and understanding the trade-off

- **We then asked: if we choose the cutoff that gives the highest accuracy instead of just using 0.5, what happens?**

- **We computed accuracy for many different probability thresholds and found the one that maximized it:**

    - **Best cutoff ≈ 0.876**

    - **Best accuracy at that cutoff ≈ 90.3%.**

- **Recomputing predictions with this higher cutoff dramatically changed the confusion matrix:**

    - **Sensitivity (true positive rate) dropped to around 0.249, meaning we only correctly catch about 25% of actual high earners.**

    - **Specificity (true negative rate) increased to about 0.983, meaning we classify low earners correctly almost all the time.**

Why this matters:

- **By pushing the cutoff up to ~0.876, we made the model very conservative about saying someone earns >50K – it only does so when it's very confident.**

- **This boosts overall accuracy and specificity, but comes at the cost of missing most high earners, which is reflected in the low sensitivity.**

- **In real applications, we usually balance these depending on what is more costly: flagging someone as high income when they are not (false positive) or failing to recognize someone who is high income (false negative).**

7. Big picture: what we achieved

- **We took raw census-style data and turned it into a structured prediction problem: "Will this person earn more than 50K?".**

- **We dealt with class imbalance by balancing the training set, so the model could learn from both high and low earners more fairly.**

- **We built a logistic regression model using interpretable numeric features and showed how each one affects the odds of high income.**

- **We evaluated the model both at a default cutoff (0.5) and at an accuracy-optimized cutoff (~0.876), and compared how this changes accuracy, sensitivity, and specificity.**

- **We used ROC and AUC to see that, overall, the model has strong discriminatory ability, even though specific cutoffs involve trade-offs depending on our objective.**

So in human terms: we taught a model to guess who earns more than 50K, checked that the variables it uses make intuitive sense, and then carefully examined how good those guesses are and what they imply for different types of errors we might care about.