Menti Code 9313 8690

ECS7024 Statistics for Artificial Intelligence and Data Science

Topic 12: Logistic Regression

William Marsh

See notebook on Logistic Regression

Outline

- Aim: introduce logistic regression

- Recap
  - Linear regression, continuous target
  - Odds ratio
- Predicting a binary variable
  - Logit function
- Accuracy, Confusion matrix and AUC
  - Rare class problem
- Extension to non-binary targets

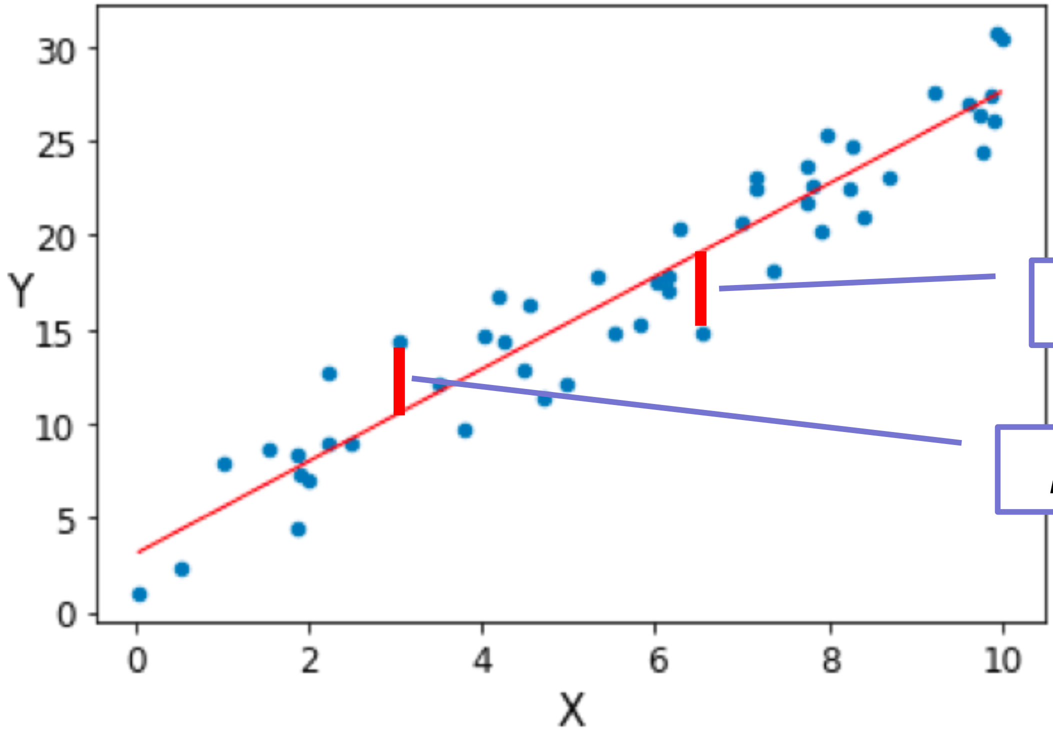
Recap

Regression Line

- Points are not exactly on a line

$$y_i = \beta_0 + \beta_1 x_{1i} + e_i$$

error



Error negative

Error positive

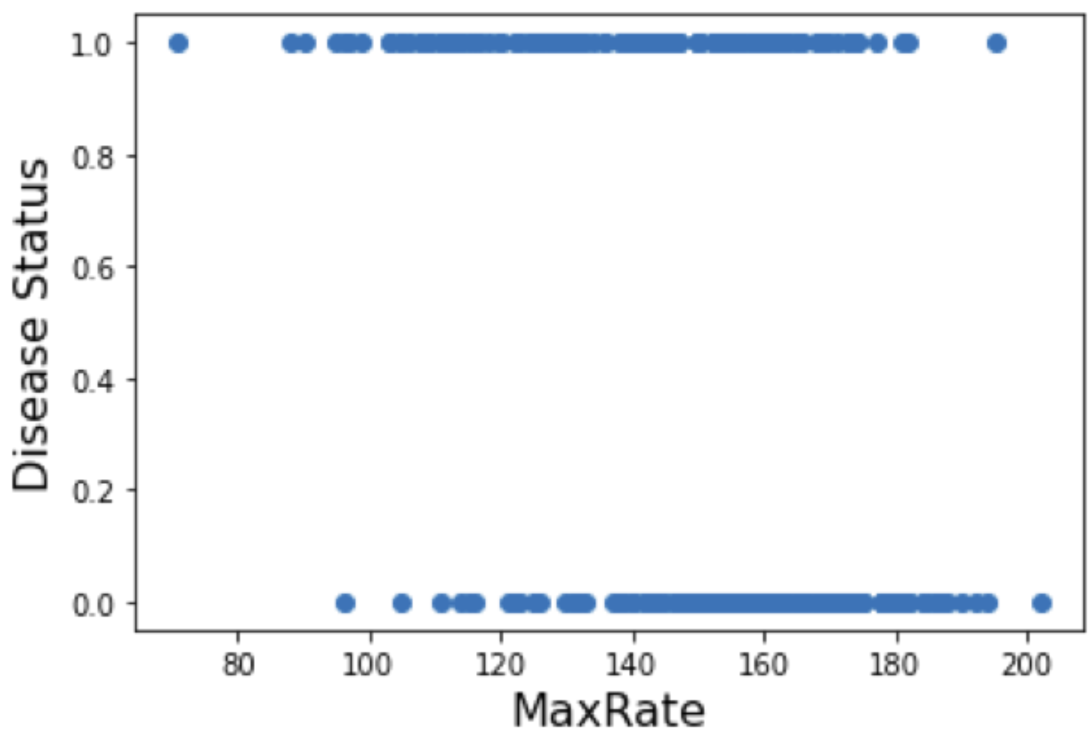
Odds is Another Way to Write a Probability

- Two rules of probability
  - $0 \le p(A) \le 1$
  - $p(A) + p(\text{not } A) = 1$ (we write 'not A' as $\bar{A}$)
- Definition of odds: $o_A = {}^{p(A)}/_{p(\bar{A})}$
  - Odds ranges from zero upwards
  - $o_{\bar{A}} = {}^{1}/_{o_A}$ so that $o_A . o_{\bar{A}} = 1$
- Example: p(A) = 75% then $\text{odds}_A = 75/25 = 3$
  - Odds > 1 implies probability > 50%
  - Odds < 1 implies probability < 50%

Predicting a Binary Variable

Problem: Regression with Binary Target

- How to use a linear regression for target with 2 values?



Logistic Regression: Key Ideas

1. Predict a probability
   - Advantage: it's a number; use it to choose class
   - Problem: range 0 to 1
   - $p = f(\beta_0 + \beta_1 . x_1 + \beta_2 . x_2)$ – choose a suitable *f()*

Logistic Regression: Key Ideas

1. Predict a probability
   - Advantage: it's a number; use it to choose class
   - Problem: range 0 to 1
   - $p = f(\beta_0 + \beta_1 . x_1 + \beta_2 . x_2)$ – choose a suitable *f()*

2. Predict odds p(Y=true) / p(Y=false)
   - Advantage: range is 0 upwards
   - Problem: not linear; cannot be negative

Logistic Regression: Key Ideas

1. Predict a probability
   – Advantage: it's a number; use it to choose class
   – Problem: range 0 to 1
   – $p = f(\beta_0 + \beta_1 . x_1 + \beta_2 . x_2)$ – choose a suitable *f()*

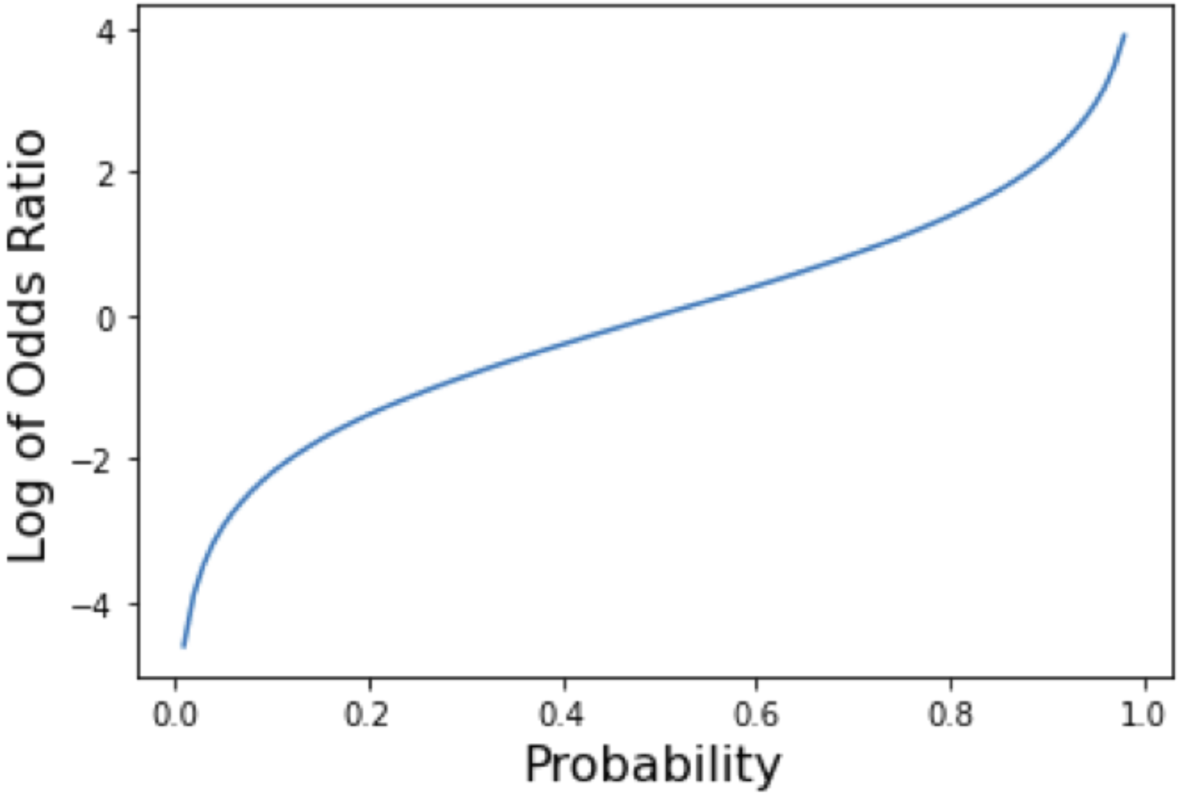
2. Predict odds p(Y=true) / p(Y=false)
   – Advantage: range is 0 upwards
   – Problem: not linear; cannot be negative

3. Predict the log of the odds
   – Solution: range over -∞ to +∞

Logit: Log Odds

- Maps probability *p* to range -∞ to +∞

Log of odds ratio: $logit(p(x)) = \ln(\frac{p(x)}{1-p(x)})$



Not the only possible conversion

Getting the Probability & Class

- Logit regression
  - Linear regression on log odds
  - $logit(p(x)) = \beta_0 + \beta_1 . x_1 + \beta_2 . x_2$

Getting the Probability & Class

- Logit regression
  - Linear regression on log odds
  - $logit(p(x)) = \beta_0 + \beta_1 . x_1 + \beta_2 . x_2$
- Odds
  - Reverse the log: $\frac{p(x)}{1-p(x)} = e^{\beta_0 + \beta_1 . x_1 + \beta_2 . x_2}$

Getting the Probability & Class

- Logit regression
  - Linear regression on log odds
  - $logit(p(x)) = \beta_0 + \beta_1 . x_1 + \beta_2 . x_2$
- Odds
  - Reverse the log: $\frac{p(x)}{1-p(x)} = e^{\beta_0+\beta_1 . x_1+\beta_2 . x_2}$
- Probability
  - Reverse the odds: $p(x) = \frac{1}{1+e^{-(\beta_0+\beta_1 . x_1+\beta_2 . x_2)}}$

Getting the Probability & Class

- Logit regression
  - Linear regression on log odds
  - $logit(p(x)) = \beta_0 + \beta_1.x_1 + \beta_2.x_2$
- Odds
  - Reverse the log: $\frac{p(x)}{1-p(x)} = e^{\beta_0+\beta_1.x_1+\beta_2.x_2}$
- Probability
  - Reverse the odds: $p(x) = \frac{1}{1+e^{-(\beta_0+\beta_1.x_1+\beta_2.x_2)}}$

- Class: y is True if p > 50% (*a possible threshold*)

Menti Code 9313 8690

Quiz 1

Programming Topic: Indexing

... and c/w reminder

Indexing in Pandas: Old and New

- Recommend using 'new style' .loc and .iloc

Object Type	Indexers
Series	`s.loc[indexer]`
DataFrame	`df.loc[row_indexer,column_indexer]`

  - Column indexer: which column?
  - Row indexer: which row?
- Types of indexers
  - a value or list of values
  - a list of Boolean – i.e. a Boolean expression
- Lots of reading:
  - https://stackoverflow.com/questions/38886080/python-pandas-series-why-use-loc
  - https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html
  - https://docs.python.org/3/reference/datamodel.html#object.__getitem__

Most internet examples use df[...]

Indexing: What's a Column?

- A data frame column is a pd.Series

	Surname	Department	Years Service	Rating
0	Lovelace	Computer Science	5	5
1	Turing	Mathematics	7	8
2	Newton	Physics	3	10
3	Franklin	Chemistry	9	9

- `pd.Rating` # simple (no spaces)
- `df2.loc[:,'Rating']` # uses .loc
- `pd['Rating']` # old style

Slice

Demo of notebook 1 answers

- Collapse code
- Restart kernel

Example

Based on Heart Data

Kaggle Data: Heart Disease I

Variable	Meaning	Type
Age	The person's age in years	Continuous
Sex	1 = male, 0 = female	Categorical
ChestPain	The chest pain experienced (1: typical angina, 2: atypical angina, 3: non-anginal pain, 4: asymptomatic)	Categorical
RestBP	The person's resting blood pressure (mm Hg on admission to the hospital)	Continuous
Chol	The person's cholesterol measurement in mg/dl	Continuous
Bsugar	The person's fasting blood sugar (> 120 mg/dl, 1 = true; 0 = false)	Binary
RestECG	Resting electrocardiographic measurement (0 = normal, 1 = having ST-T wave abnormality, 2 = showing probable e left ventricular hypertrophy)	Ordinal (?)

Kaggle Data: Heart Disease II

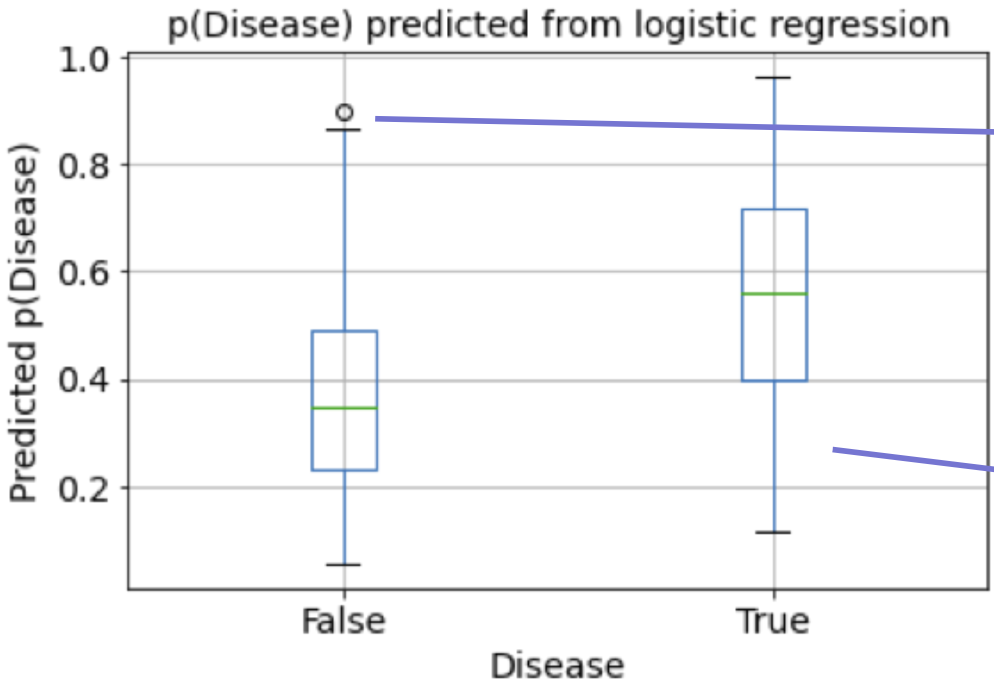
Variable	Meaning	Type
MaxRate	The person's maximum heart rate achieved	Continuous
Angina	Exercise induced angina (1 = yes; 0 = no)	Binary
ECG_ST_d	ST depression induced by exercise relative to rest ('ST' relates to positions on the ECG plot)	Continuous
ECG_ST_slope	The slope of the peak exercise ST segment (1: upsloping, 2: flat, 3: downsloping)	Categorical
Vessels	The number of major vessels (0-3) coloured by fluoroscopy	Ordinal
Thallium	Thallium update test (0 = normal; 1 = fixed defect; 2 = reversible defect)	Categorical
Disease	Heart disease (0 = no, 1 = yes)	Binary

Predict Disease Status (Age, MaxRate)

- Using continuous variables as predictors
  - Predicts p(Disease = True)

Predictor	Beta
Intercept	2.868
Age	0.020
MaxRate	-0.042

Probability of heart disease increases with age: each year increases odd ratio by ~2%



Outlier

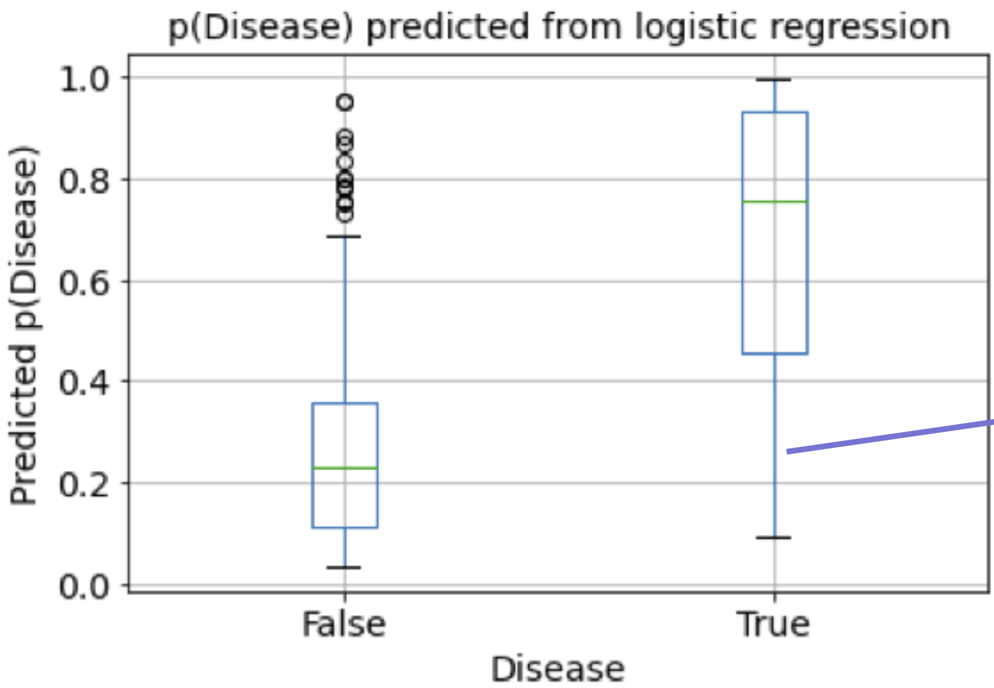
Overlap: some predicted probability in true cases lower than false cases

Predict Disease Status (More Predictors)

- Using continuous variables as predictors
  - Predicts p(Disease = True)

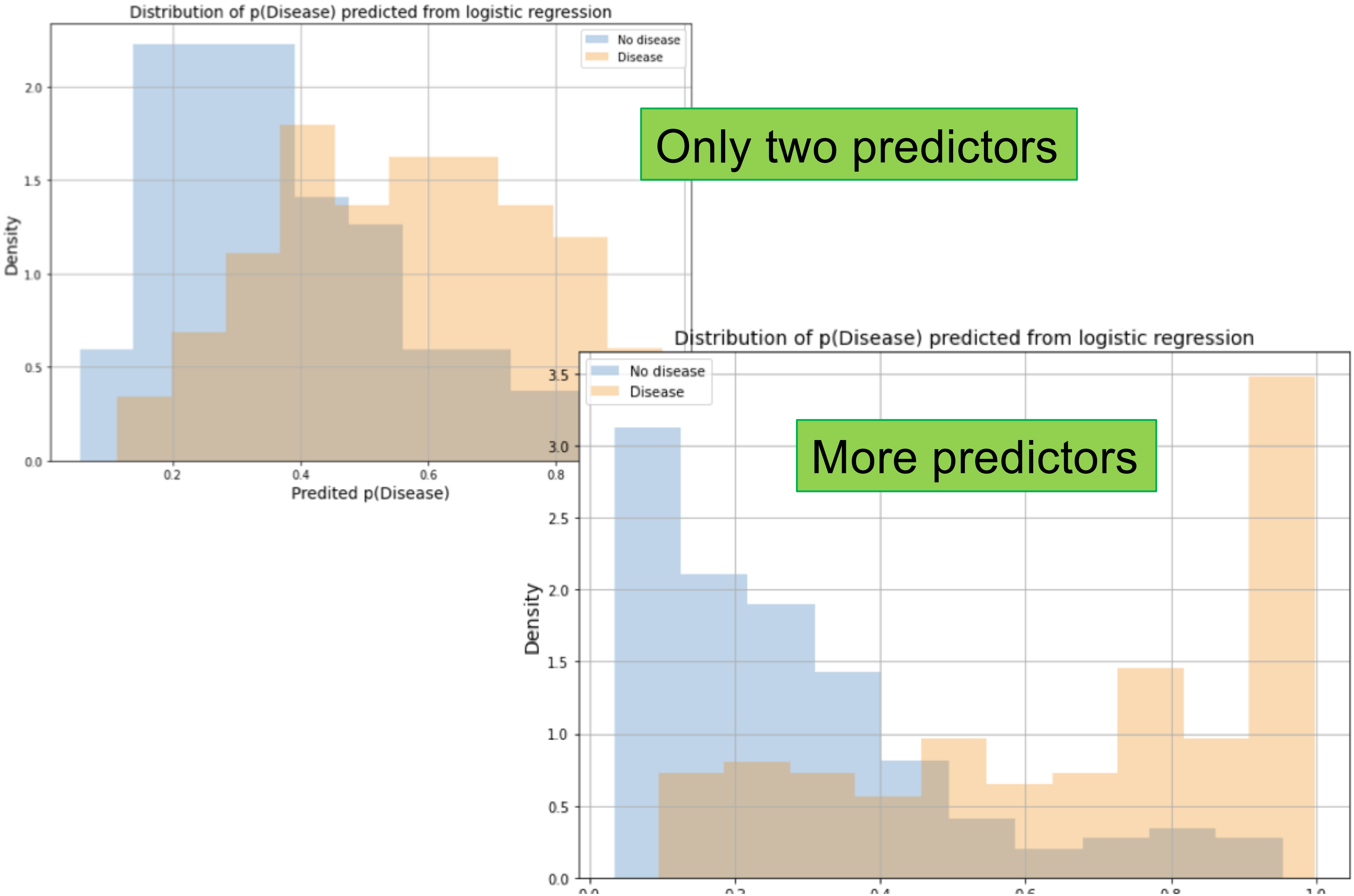
Predictor	Beta
Intercept	2.868
Age	-0.036
RestBP	0.018
Chol	0.003
MaxRate	-0.037
ECG_ST_d	0.624
Vessel	1.213

Sign has changed



Still some overlap

Distribution of Predicted Probabilities



Accuracy, Confusion Matrix and AUC

Is there an R^2 Equivalent?

Applies to any binary classifier

Errors: false positive and false negative

- Is the prediction correct?

	Disease	Predicted-disease	Predicted_class
0	False	0.604642	True
1	False	0.831805	True
2	False	0.786860	True
3	True	0.379604	False
4	False	0.264561	False
...	...	...	...
292	True	0.134812	False
293	True	0.454062	False
294	True	0.635814	True
295	False	0.228263	False
296	True	0.257882	False

Incorrectly predicted True

Incorrectly predicted False

Confusion Matrix

- Compare actual and predicted

		Predicted Disease Status	
		Positive	**Negative**
True Disease Status	**Positive**	True positive (TP)	False negative (FN)
	Negative	False positive (FP)	True negative (TN)

- Classification depends on probability threshold
- Are both types of error equal?

Measure of Performance I

- Condition positive: P = TP + FN
- Condition negative: N = TN + FP

Measure of Performance I

- Condition positive: P = TP + FN
- Condition negative: N = TN + FP

- True positive rate (TPR) = TP / P
  - Also ‘Sensitivity’, ‘Recall’
  - *How many positive cases are found?*
- False positive rate (FPR) = FP / N

Measure of Performance I

- Condition positive: P = TP + FN
- Condition negative: N = TN + FP

- True positive rate (TPR) = TP / P
  - Also ‘Sensitivity’, ‘Recall’
  - *How many positive cases are found?*
- False positive rate (FPR) = FP / N

- True negative rate (TNR) = TN / N = 1 – FPR
  - Also Specificity
  - How many negative cases were found?
- False negative rate (FNR) = FP / P = 1 – TPR

Measures of Performance II

- Accuracy
  - *Proportion of correct predictions*
  - (TP + TN) / (N+P)

- Precision
  - *How many predicted positives are correct?*
  - TP / (TP + FP)

Confusion Matrix: Heart Disease

- Heart disease status:

		Predicted Disease Status	
		Positive	**Negative**
True Disease Status	**Positive**	95 (TP)	42 (FN)
	Negative	22 (FP)	138 (TN)

- Depends on the threshold probability - 50%

Performance of Heart Disease

- Sensitivity (Recall, TPR)
  - *How many positive cases are found?*
  - TP / (TP + FN) = 95 / (95+42) = 69%

Performance of Heart Disease

- Sensitivity (Recall, TPR)
  - *How many positive cases are found?*
  - TP / (TP + FN) = 95 / (95+42) = 69%
- Specificity (TNR)
  - *How many negative cases were found?*
  - TN / (TN + FP) = 138 / (138+22) = 86%

Performance of Heart Disease

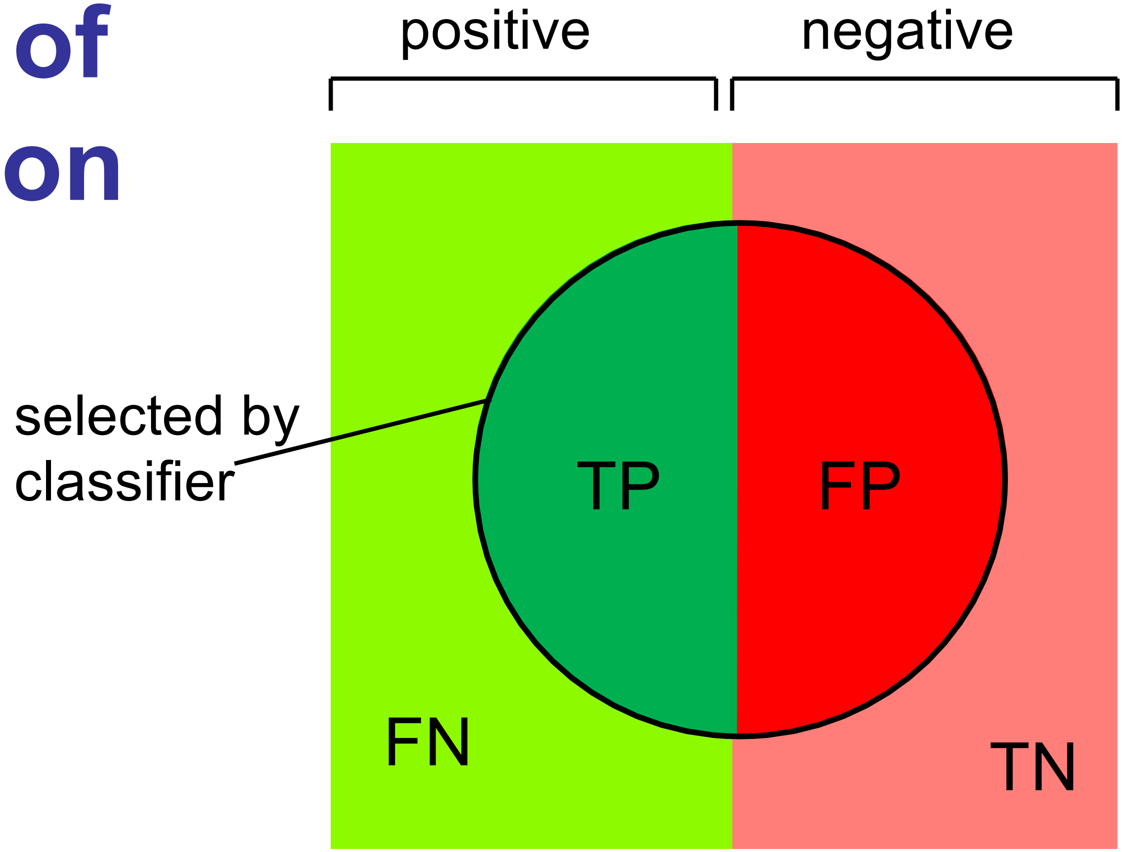
- Sensitivity (Recall, TPR)
  - *How many positive cases are found?*
  - TP / (TP + FN) = 95 / (95+42) = 69%
- Specificity (TNR)
  - *How many negative cases were found?*
  - TN / (TN + FP) = 138 / (138+22) = 86%
- Accuracy
  - (TP + TN) / Total = (95 + 138) / 297 = 78%

Performance of Heart Disease

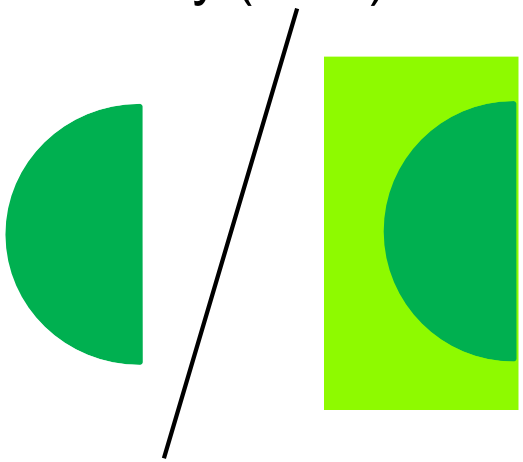
- Sensitivity (Recall, TPR)
  - *How many positive cases are found?*
  - TP / (TP + FN) = 95 / (95+42) = 69%
- Specificity (TNR)
  - *How many negative cases were found?*
  - TN / (TN + FP) = 138 / (138+22) = 86%
- Accuracy
  - (TP + TN) / Total = (95 + 138) / 297 = 78%
- Precision
  - *How many predicted positive are correct?*
  - TP / (TP + FP) = 95 / (95+22) = 81%

Picture of Confusion

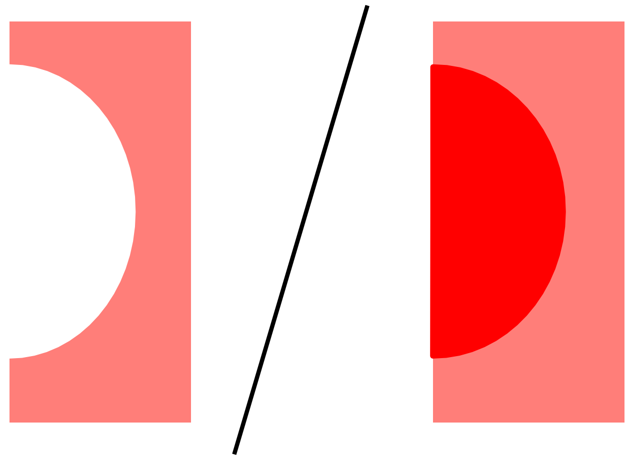
- What happens if you expand the selected region?



Sensitivity (TPR)



Specificity (TNR)



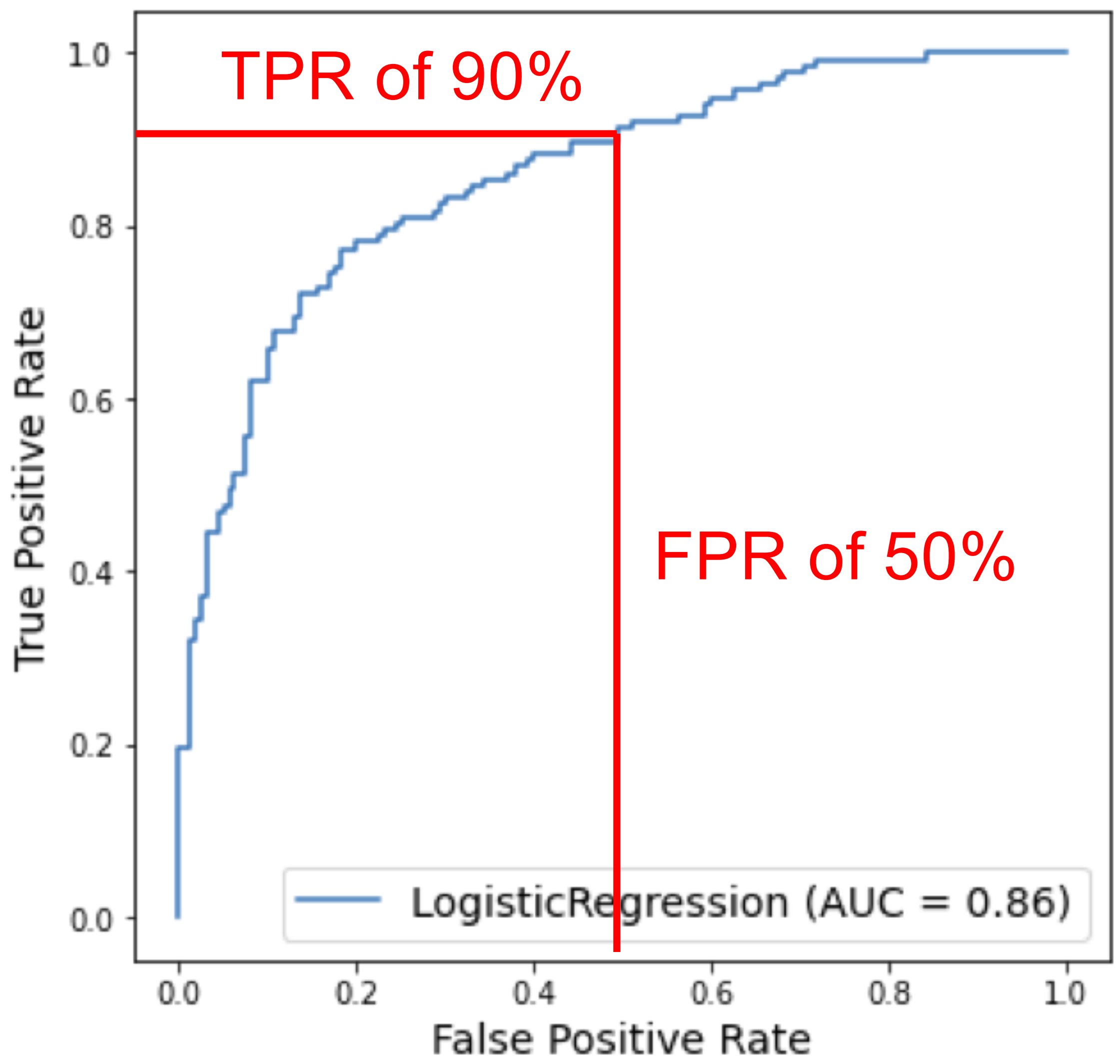
The Rare Class (Low Rate) Problem

- What if the ‘true’ state in the classification is rare?
  - Perhaps 0.1%
  - Then **always predicting false** is 99.9% accurate
  - ... but TPR = 0% useless

- Lower accuracy and higher FPR more useful

ROC: Sensitivity v Specificity

- Y axis
  - TPR (Sensitivity)
- X axis
  - FPR (1 – Specificity)
- Curve
  - Possible operating points
  - Given by threshold
- AUC: measure of performance



Menti Code 9313 8690

Quiz 2

Extension to Non-Binary Target Variables

More than 2 Values

- Multinomial
  - *Applies to categorical variable with > 2 values*
  - For N values, N-1 regression equations
  - ... results in a probability for each value
  - Handled by `sklearn.linear_model`
- Ordinal
  - *Applies to ordinal variables (categories with order)*
  - Either like multinomial or more as continuous
  - Less well supported in Python

Generalised Linear Models

- Extends linear regression
  - Logistic regression is one of many examples
- Key ideas
  - Response variable from different distributions
  - Link function (*logit* is an example)
    - Result of linear regression ‘linked’ to response variable

- Also multi-level models

How Logistic Regression Works

- Linear regression
  - Ordinary least squares
  - Closed-form solution
- Logistic regression (and other GLM)
  - No closed-form solution
  - Optimisation problem: search for a solution

- Maximum likelihood estimation (MLE)
  - Search for parameters that make data most probable
  - ‘Best fit’
  - Rare class problem: over- and under- sampling

Summary

- Logistic regression
  - Applies to binary (or categorical) response variables
  - An example of ‘generalised’ linear models
  - Find parameters using MLE

- Linear regression to predict the log of the odds

- Regression is surprisingly fundamental