

CS M148 –

Data Science Fundamentals

Lecture #8: Logistic Regression

Baharan Mirzasoleiman
UCLA Computer Science

Announcements

Request for regrading Project 1

- Due Feb 5, 11pm

Project 2

- Due Wed Feb 9, 2pm

Lecture and discussions

- Online this week

Let's quickly review what we saw last time

Still here!

The Data Science Process

Ask an interesting question

Get the Data

Clean/Explore the Data

Model the Data

Communicate/Visualize the Results

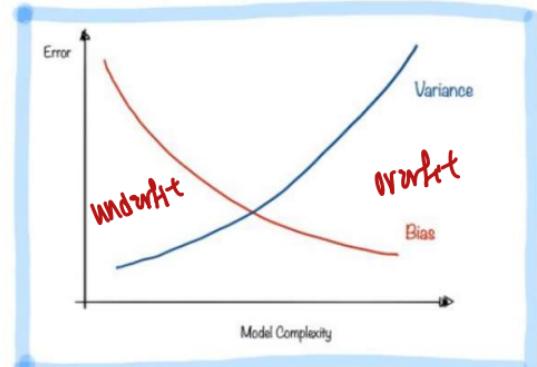
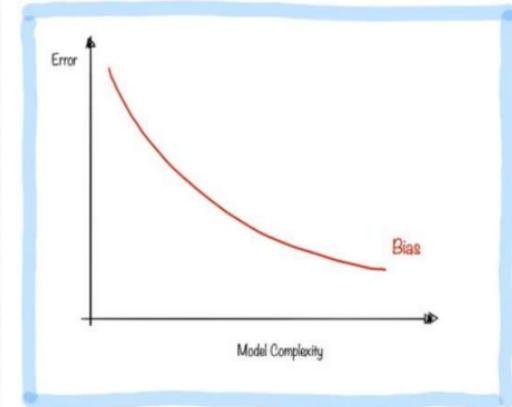


Regularization error, bias vs variance

The Bias-Variance Trade Off: Bias



undurkt



Overfitting

Overfitting occurs when a model corresponds too closely to the training set, and as a result, the model fails to fit additional data.

So far, we have seen that overfitting can happen when:

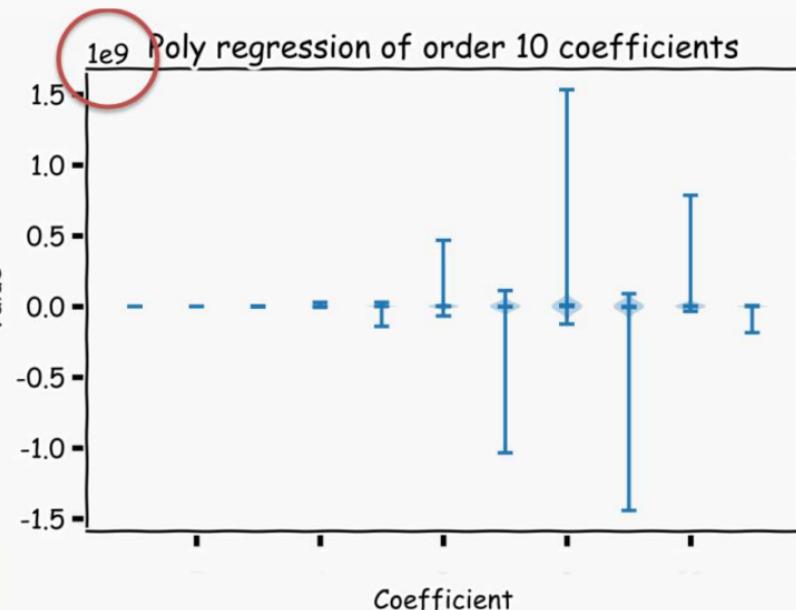
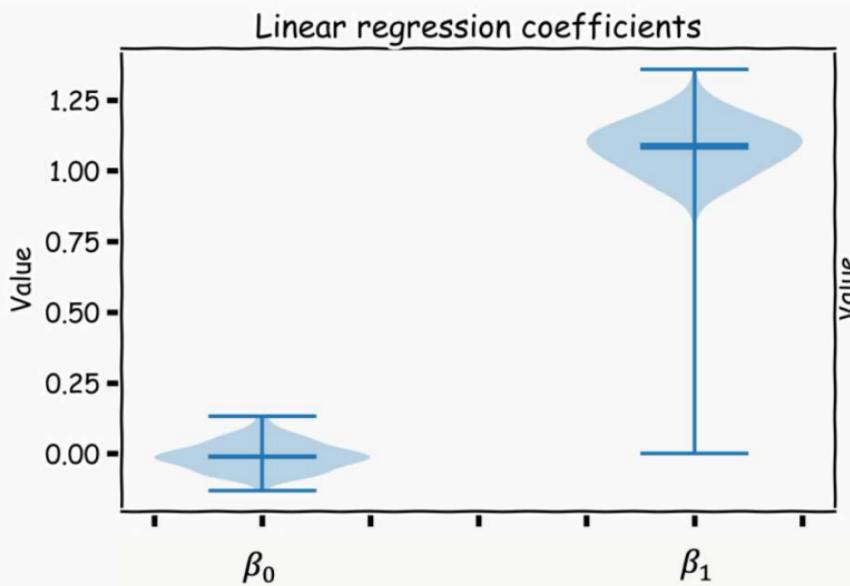
- Too many parameters
- Degree of the polynomial is too large
- Too many interaction terms

Next, we will see other evidence of overfitting, which will point to a way of avoiding overfitting: **Ridge and Lasso regressions**.

Bias vs Variance

Left: Linear regression coefficients

Right: Poly regression of order 10 coefficients



Regularization: An Overview

The idea of regularization revolves around modifying the loss function L ; in particular, we add a regularization term that penalizes some specified properties of the model parameters

$$L_{reg}(\beta) = L(\beta) + \lambda R(\beta)$$

where λ is a scalar that gives the weight (or importance) of the regularization term.

Fitting the model using the modified loss function L_{reg} would result in model parameters with desirable properties (specified by R).

LASSO Regression

Since we wish to discourage extreme values in model parameter, we need to choose a regularization term that penalizes parameter magnitudes. For our loss function, we will again use MSE.

Together our regularized loss function is:

$$L_{LASSO}(\beta) = \frac{1}{n} \sum_{i=1}^n |y_i - \boldsymbol{\beta}^\top \mathbf{x}_i|^2 + \lambda \sum_{j=1}^J |\beta_j|.$$

Note that $\sum_{j=1}^J |\beta_j|$ is the l_1 norm of the vector $\boldsymbol{\beta}$

$$\sum_{j=1}^J |\beta_j| = \|\boldsymbol{\beta}\|_1$$

Ridge Regression

Alternatively, we can choose a regularization term that penalizes the squares of the parameter magnitudes. Then, our regularized loss function is:

$$L_{Ridge}(\beta) = \frac{1}{n} \sum_{i=1}^n |y_i - \boldsymbol{\beta}^\top \mathbf{x}_i|^2 + \lambda \sum_{j=1}^J \beta_j^2.$$

Note that $\sum_{j=1}^J \beta_j^2$ is the l_2 norm of the vector $\boldsymbol{\beta}$

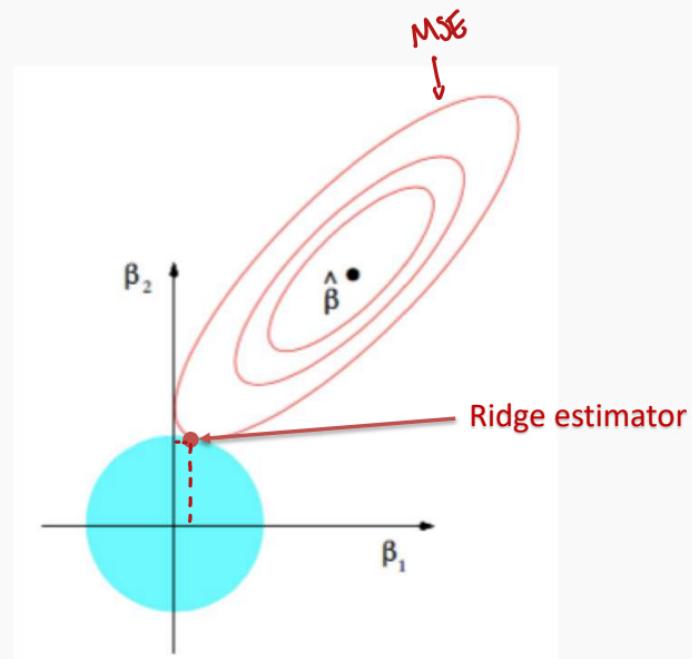
$$\sum_{j=1}^J \beta_j^2 = \|\boldsymbol{\beta}\|_2^2$$

Regularization Parameter with a Validation Set

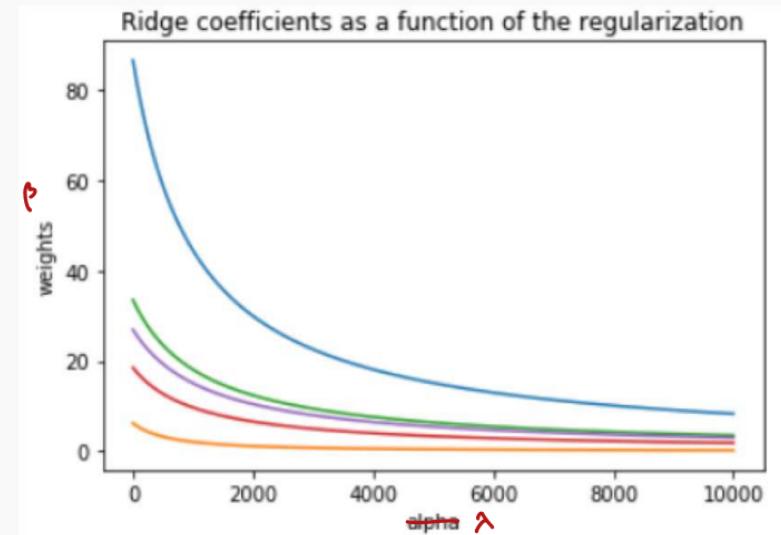
The solution of the Ridge/Lasso regression involves three steps:

- Select λ
- Find the minimum of the ridge/Lasso regression loss function (using the formula for ridge) and record the MSE **on the validation set.**
- Find the λ that gives the **smallest MSE on the validation set.**

Ridge visualized

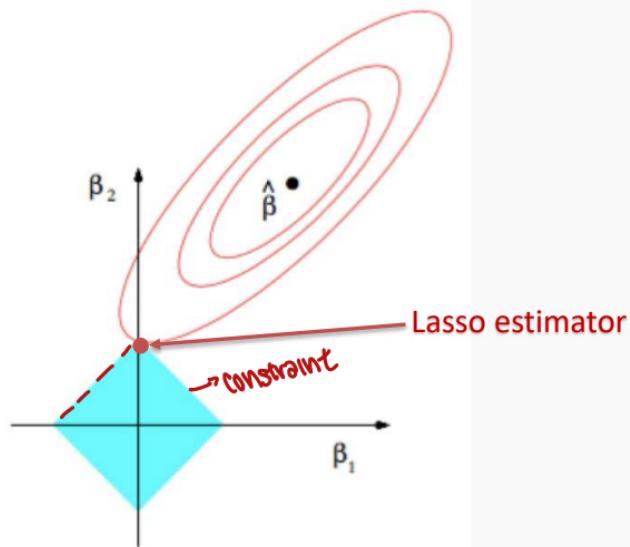


The ridge estimator is where the constraint and the loss intersect.

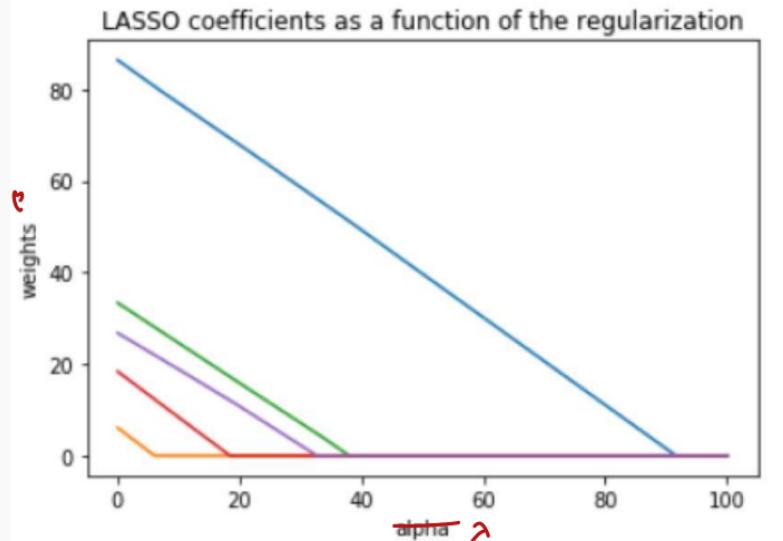


The values of the coefficients decrease as lambda increases, but they are not nullified.

LASSO visualized



The Lasso estimator tends to zero out parameters as the OLS loss can easily intersect with the constraint on one of the axes.



The values of the coefficients decrease as lambda increases, and are nullified fast.

Variable Selection as Regularization

Question: What are the pros and cons of the two approaches?

Since LASSO regression tends to **produce zero estimates** for a number of model parameters - we say that LASSO solutions are **sparse** - we consider LASSO to be a **variable selection method**.

Ridge is faster to compute but many prefer using LASSO for variable selection (as well as for suppressing extreme parameter values) and therefore easier to **interpret**.
has closed form
doesn't have closed form

Classification

Lecture Outline

- Introduction to Classification
- k -NN for Classification
- Why not Linear Regression?
- Binary Response & Logistic Regression
 - Estimating the Simple Logistic Model
 - Classification using the Logistic Model
 - Multiple Logistic Regression

Advertising Data (from earlier lectures)

The diagram illustrates the structure of the advertising data. A large bracket on the left indicates n observations, and a bracket at the bottom indicates p predictors. Two speech bubbles define the terms: one for predictors (X) and one for the outcome variable (Y).

X
predictors
features
covariates

Y
outcome
response variable
dependent variable

TV	radio	newspaper	sales
230.1	37.8	69.2	22.1
44.5	39.3	45.1	10.4
17.2	45.9	69.3	9.3
151.5	41.3	58.5	18.5
180.8	10.8	58.4	12.9

Heart Data

response variable **Y**
is Yes/No

Age	Sex	ChestPain	RestBP	Chol	Fbs	RestECG	MaxHR	ExAng	Oldpeak	Slope	Ca	Thal	AHD
63	1	typical	145	233	1	2	150	0	2.3	3	0.0	fixed	No
67	1	asymptomatic	160	286	0	2	108	1	1.5	2	3.0	normal	Yes
67	1	asymptomatic	120	229	0	2	129	1	2.6	2	2.0	reversible	Yes
37	1	nonanginal	130	250	0	0	187	0	3.5	3	0.0	normal	No
41	0	nontypical	130	204	0	2	172	0	1.4	1	0.0	normal	No

Heart Data

These data contain a binary outcome HD for 303 patients who presented with chest pain. An outcome value of:

- **Yes** indicates the presence of heart disease based on an angiographic test,
- **No** means no heart disease.

There are 13 predictors including:

- Age
- Sex (0 for women, 1 for men)
- Chol (a cholesterol measurement),
- MaxHR
- RestBP

and other heart and lung function measurements.

Classification

Up to this point, the methods we have seen have centered around modeling and the prediction of a **quantitative** response variable (ex, number of taxi pickups, number of bike rentals, etc). Linear **regression** (and Ridge, LASSO, etc) perform well under these situations

When the response variable is **categorical**, then the problem is no longer called a regression problem but is instead labeled as a **classification problem**.

The goal is to attempt to classify each observation into a category (aka, class or cluster) defined by Y , based on a set of predictor variables X .

k-NN for Classification

k-NN for Classification

How can we modify the *k*-NN approach for classification?

The approach here is the same as for *k*-NN regression: use the other available observations that are most similar to the observation we are trying to predict (classify into a group) based on the predictors at hand.

How do we classify which category a specific observation should be in based on its nearest neighbors?

k -NN for Classification: formal definition

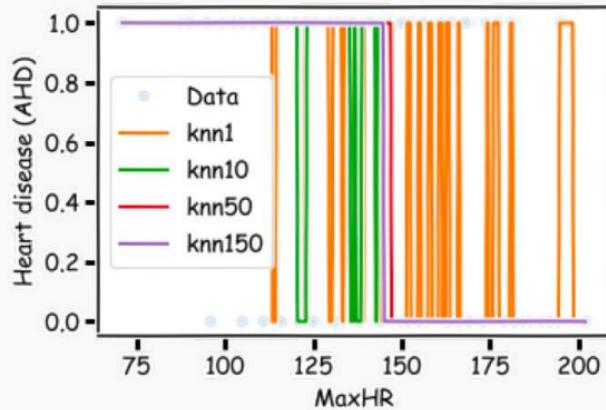
The k -NN classifier first identifies the k points in the training data that are closest to x_0 , represented by \mathcal{N}_0 . It then estimates the conditional probability for class j as the fraction of points in \mathcal{N}_0 whose response values equal j :

$$P(Y = j | X = x_0) = \frac{1}{k} \sum_{i \in \mathcal{N}_0} I(y_i = j)$$

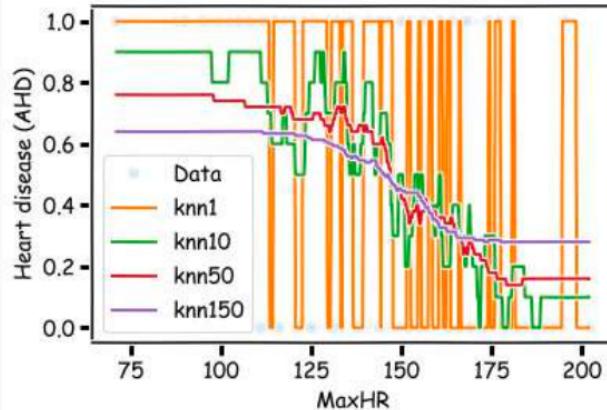
Then, the k -NN classifier predicts this new observation, x_0 , to be in the class with largest estimated probability.

Pure Classifications in k -NN Classification Models

Pure Classifications:



Estimated Probabilities:



What will these plots look like when there are 2 predictors? Or 3+ predictors?

This is our first glimpse at a **classification boundary** (more to come).

k-NN with Multiple Predictors

How could we extend *k*-NN (both regression and classification) when there are multiple predictors?

We would need to define a measure of distance for observations in order to which are the most similar to the observation we are trying to predict.

Euclidean distance is a good option. To measure the distance of a new observation, \mathbf{x}_0 from each observation in the data set, \mathbf{x}_i :

$$D^2(\mathbf{x}_i, \mathbf{x}_0) = \sum_{j=1}^P (x_{i,j} - x_{0,j})^2$$

k -NN with Multiple Predictors (cont.)

But what must we be careful about when measuring distance?

1. Differences in variability in our predictors!
2. Having a mixture of quantitative and categorical predictors.

So what should be good practice? To determine closest neighbors when $p > 1$, you should first standardize the numeric predictors! And you can even standardize the binaries if you want to include them.

How else could we determine closeness in this multi-dimensional setting?

$$\frac{x_i - M}{\sigma} \in [0,1]$$

Parametric Modeling: Why not Linear Regression?

Simple Classification Example

Given a dataset:

$$\{(\mathbf{x}_1, y_1), (\mathbf{x}_2, y_2), \dots, (\mathbf{x}_N, y_N)\}$$

where the y are categorical (sometimes referred to as *qualitative*), we would like to be able to predict which category y takes on given x .

A categorical variable y could be encoded to be quantitative. For example, if y represents concentration of UCLA undergrads, then y could take on the values:

$$y = \begin{cases} 1 & \text{if Computer Science (CS)} \\ 2 & \text{if Statistics} \\ 3 & \text{otherwise} \end{cases}.$$

Linear regression **does not work well**, or is not appropriate at all,
in this setting.

Simple Classification Example (cont.)

A linear regression could be used to predict y from x . What would be wrong with such a model?

The model would imply a specific ordering of the outcome, and would treat a one-unit change in y equivalent. The jump from $y = 1$ to $y = 2$ (**CS** to **Statistics**) should not be interpreted as the same as a jump from $y = 2$ to $y = 3$ (**Statistics** to **everyone else**).

Similarly, the response variable could be reordered such that $y = 1$ represents **Statistics** and $y = 2$ represents **CS**, and then the model estimates and predictions would be fundamentally different.

If the categorical response variable was ***ordinal*** (had a natural ordering, like class year: Freshman, Sophomore, etc.), then a linear regression model would make some sense but is still not ideal.

Even Simpler Classification Problem: Binary Response

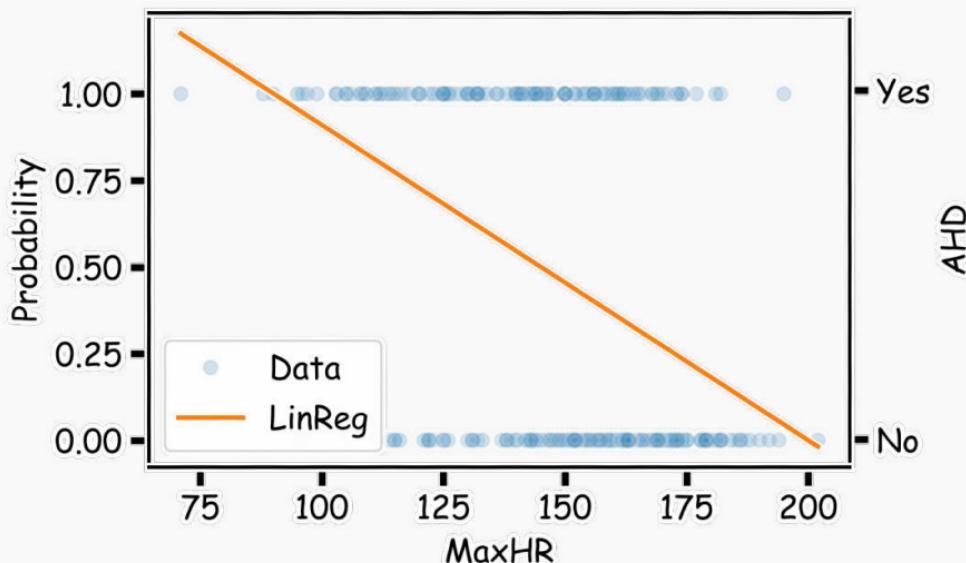
The simplest form of classification is when the response variable y has only two categories, and then an ordering of the categories is natural. For our example, a patient in the ICU could be categorized as having [atherosclerotic] heart disease (AHD) or not (note, the $y = 0$ category is a "catch-all" so it would involve those patients with lots of other diseases or diagnoses):

$$y = \begin{cases} 1 & \text{if patient has heart disease} \\ 0 & \text{otherwise.} \end{cases}$$

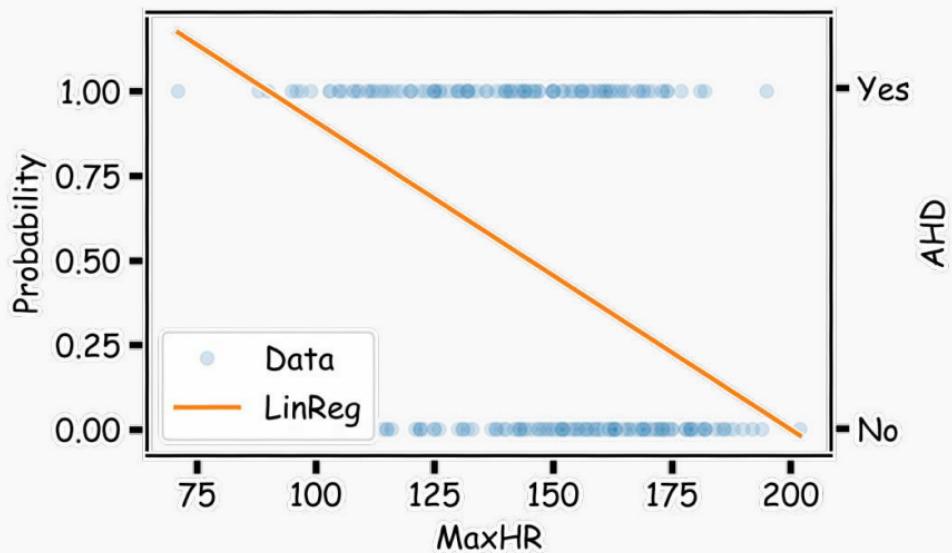
Linear regression could be used to predict y directly from a set of covariates (like sex, age, resting HR, etc.), and if $\hat{y} \geq 0.5$, we could predict the patient to have AHD and predict not to have heart disease if $\hat{y} < 0.5$.

Even Simpler Classification Problem: Binary Response (cont)

What could go wrong with this linear regression model?



Even Simpler Classification Problem: Binary Response (cont)

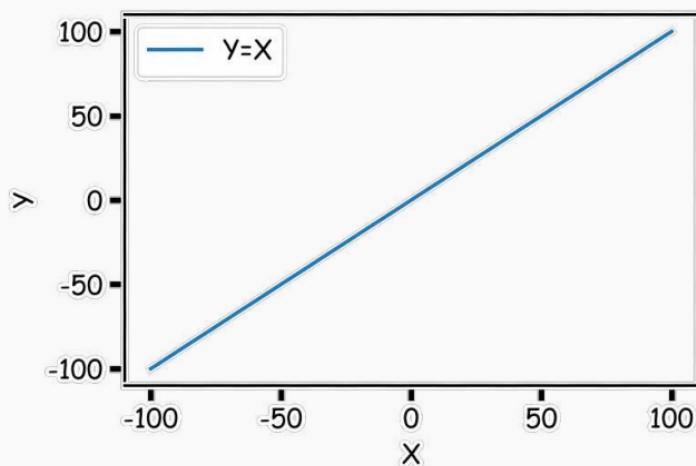


The main issue is you could get non-sensical values for y . Since this is modeling $P(y = 1)$, values for \hat{y} below 0 and above 1 would be at odds with the natural measure for y . Linear regression can lead to this issue.

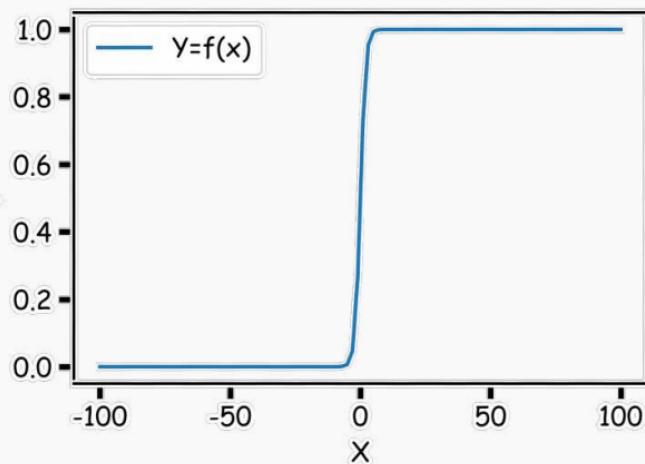
Binary Response & Logistic Regression

Logistic Regression

Think of a function that would do this for us



$$Y = f(x)$$



Logistic Regression

Logistic Regression addresses the problem of estimating a probability, $P(y = 1)$, to be outside the range of [0,1]. The logistic regression model uses a function, called the *logistic* function, to model $P(y = 1)$:

$$P(Y = 1) = \frac{e^{\beta_0 + \beta_1 X}}{1 + e^{\beta_0 + \beta_1 X}} = \frac{1}{1 + e^{-(\beta_0 + \beta_1 X)}}$$

Logistic Regression

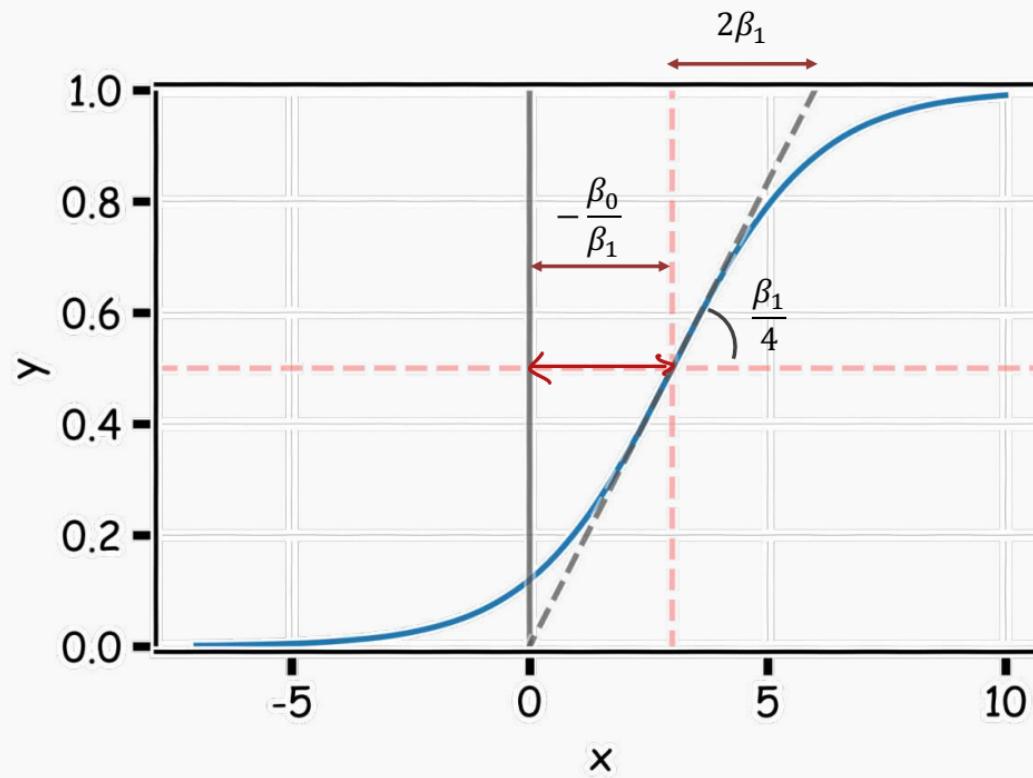
As a result the model will predict $P(y = 1)$ with an *S*-shaped curve, which is the general shape of the logistic function.

β_0 shifts the curve right or left by $c = -\frac{\beta_0}{\beta_1}$.

β_1 controls how steep the *S*-shaped curve is. Distance from $\frac{1}{2}$ to almost 1 or $\frac{1}{2}$ to almost 0 is $\frac{2}{\beta_1}$

Note: if β_1 is positive, then the predicted $P(y = 1)$ goes from zero for small values of X to one for large values of X and if β_1 is negative, then the $P(y = 1)$ has opposite association.

Logistic Regression



Logistic Regression: interpretation

With a little bit of algebraic work, the logistic model can be rewritten as:

$$\ln \left(\frac{P(Y = 1)}{1 - P(Y = 1)} \right) = \beta_0 + \beta_1 X.$$

The value inside the natural log function $\frac{P(Y=1)}{1-P(Y=1)}$, is called the **odds**, thus logistic regression is said to model the **log-odds** with a linear function of the predictors or features, X . This gives us the natural interpretation of the estimates similar to linear regression: a one unit change in X is associated with a β_1 change in the log-odds of success ($Y = 1$); or better yet, a **one unit change in X is associated with an e^{β_1} multiplicate change in the odds of success ($Y = 1$)**.

Estimating the Simple Logistic Model

Estimation in Logistic Regression

Unlike in linear regression where there exists a closed-form solution to finding the estimates, $\hat{\beta}_j$'s, for the true parameters, logistic regression estimates cannot be calculated through simple matrix multiplication.

Questions:

- In linear regression what loss function was used to determine the parameter estimates? $y = f(x) + \epsilon$
 $MS\bar{E}$
- What was the probabilistic perspective on linear regression? $y \sim N(f(x), \sigma^2)$
- Logistic Regression also has a likelihood based approach to estimating parameter coefficients.

Estimation in Logistic Regression

Probability Mass Function (PMF):

$$\left\{ \begin{array}{l} P(Y = 1) = p \\ P(Y = 0) = 1 - p \\ P(Y = y) = p^y(1 - p)^{(1-y)} \end{array} \right.$$

← Prob of 1 data point being $y|0$ or 1

where:

$$p = P(Y = 1|X = x) = \frac{1}{1 + e^{-(\beta_0 + \beta_1 X)}}$$

and therefore p depends on X .

Thus not every p_i is the same for each individual measurement.

Likelihood

The likelihood of a single observation for p given x and y is:

$$L(p_i|Y_i) = P(Y_i = y_i) = p_i^{y_i} (1 - p_i)^{1-y_i}$$

Given the observations are independent, what is the likelihood function for p ?

likelihood & data
maximizer

$$L(p|Y) = \prod_i P(Y_i = y_i) = \prod_i p_i^{y_i} (1 - p_i)^{1-y_i}$$

likelihood for point i

easier to solve

$$l(p|Y) = -\log L(p|Y) = -\sum_i y_i \log p_i + (1 - y_i) \log(1 - p_i)$$

this maximization turns into minimization turns Σ into Σ

Loss Function

$$l(p|Y) = - \sum_i \left[y_i \log \frac{1}{1 + e^{-(\beta_0 + \beta_1 X_i)}} + (1 - y_i) \log \left(1 - \frac{1}{1 + e^{-(\beta_0 + \beta_1 X_i)}} \right) \right]$$

How do we minimize this?

Differentiate, equate to zero and solve for it!

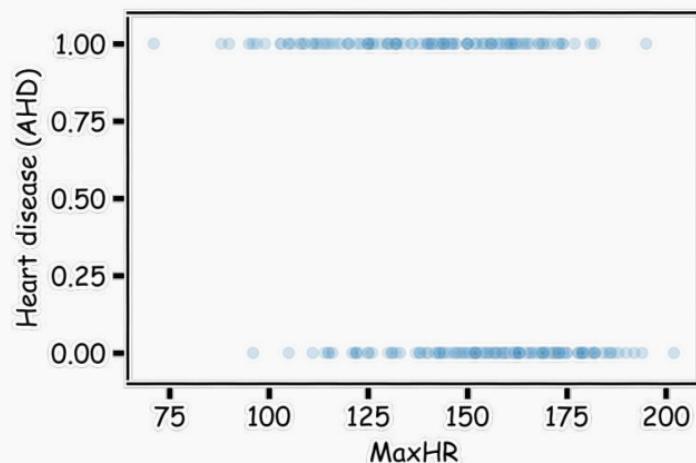
But jeeze does this look messy?! It will not necessarily have a closed form solution.

So how do we determine the parameter estimates? Through an iterative approach (we will talk about this *at length* in future lectures).

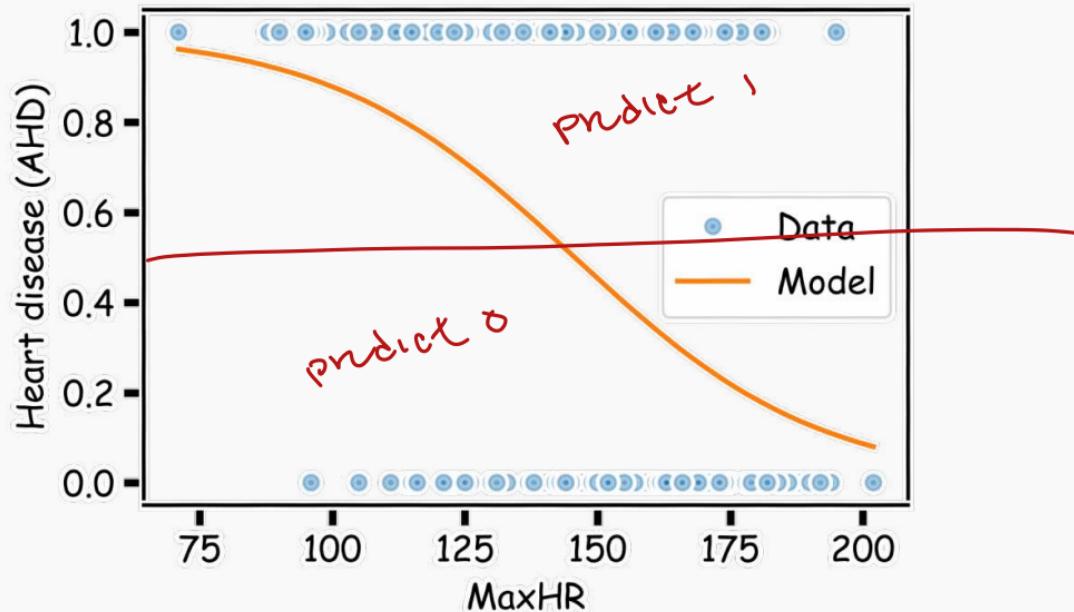
Heart Data: logistic estimation

We'd like to predict whether or not a person has a heart disease. And we'd like to make this prediction, for now, just based on the MaxHR.

How should we visualize these data?



Heart Data: logistic estimation



Heart Data: logistic estimation

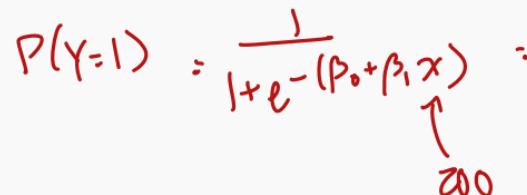
There are various ways to fit a logistic model to this data set in Python. The most straightforward in `sklearn` is via `linear_model.LogisticRegression`.

```
from sklearn.linear_model import LogisticRegression  
logreg = LogisticRegression(C=100000, fit_intercept=True)  
logreg.fit(data_x.values.reshape(-1,1), data_y); fit w/ l1  
  
print('Estimated beta1: \n', logreg.coef_)  
print('Estimated beta0: \n', logreg.intercept_)
```

```
Estimated beta1:  
[[-0.04326016]]  
Estimated beta0:  
[ 6.30193148]
```

Heart Data: logistic estimation

Answer some questions:

- Write down the logistic regression model. $\ln\left(\frac{P(Y=1)}{P(Y=0)}\right) = 6.3 - 0.04X$
- Interpret $\hat{\beta}_1$. $\ln 2$ unit incr. in HR \rightarrow 4% incr. in log
- Estimate the probability of heart disease for someone with MaxHR ≈ 200 ?
$$P(Y=1) = \frac{1}{1+e^{-(\beta_0 + \beta_1 x)}} =$$

- If we were to use this model purely for classification, how would we do so? See any issues?

Categorical Predictors

Just like in linear regression, when the predictor, X , is binary, the interpretation of the model simplifies (and there is a quick closed form solution here).

In this case, what are the interpretations of $\hat{\beta}_0$ and $\hat{\beta}_1$?

For the heart data, let X be the indicator that the individual is a female ($X = 0$) or male ($X = 1$). What is the interpretation of the coefficient estimates in this case?

The observed percentage of HD for women is 26% while it is 55% for men.

Calculate the estimates for $\hat{\beta}_0$ and $\hat{\beta}_1$ if the indicator for HD was predicted from the gender indicator (work on next page).

Solving for the estimates mathematically:

$$P(Y = 1|X = 0) = 0.26, \quad P(Y = 1|X = 1) = 0.55$$

$$\ln\left(\frac{P(Y = 1)}{1 - P(Y = 1)}\right) = \beta_0 + \beta_1 X$$

finals ($x=0$): $\ln\left(\frac{P(Y=1)}{P(Y=0)}\right) = \ln\left(\frac{0.26}{1-0.26}\right) \therefore \beta_0 = -1.04$

more ($x=1$): $\ln\left(\frac{P(Y=1)}{P(Y=0)}\right) = \ln\left(\frac{0.55}{1-0.55}\right) \therefore \beta_0 + \beta_1 = 0.2$

$$P(Y=1|X=0) = \frac{1}{1+e^{-\beta_0}}$$

Statistical Inference in Logistic Regression

The uncertainty of the estimates $\hat{\beta}_0$ and $\hat{\beta}_1$ can be quantified and used to calculate both confidence intervals and hypothesis tests.

The estimate for the standard errors of these estimates, likelihood-based, is based on a quantity called Fisher's Information (beyond the scope of this class), which is related to the curvature of the log-likelihood function.

Due to the nature of the underlying Bernoulli distribution, if you estimate the underlying proportion p_i , you get the variance for free! Because of this, the inferences will be based on the normal approximation (and not t -distribution based). Use statsmodels to perform inferences!

Of course, you could always **bootstrap** the results to perform these inferences as well.

Classification using the Logistic Model

Using Logistic Regression for Classification

How can we use a logistic regression model to perform classification?

That is, how can we predict when $Y = 1$ vs. when $Y = 0$?

We mentioned before, we can classify all observations for which $\hat{P}(Y = 1) \geq 0.5$ to be in the group associated with $Y = 1$ and then classify all observations for which $\hat{P}(Y = 0) < 0.5$ to be in the group associated with $Y = 0$.

Using such an approach is called the standard **Bayes classifier**.

The Bayes classifier takes the approach that assigns each observation to the most likely class, given its predictor values.

Using Logistic Regression for Classification

When will this Bayes classifier be a good one? When will it be a poor one?

The Bayes classifier is the one that minimizes the overall classification error rate.

That is, it ~~minimizes~~:

maximizes

$$\frac{1}{n} \sum_i^n I(y_i = \hat{y}_i)$$

label *predicted*

Is this a good Loss function to minimize? Why or why not?

The Bayes classifier may be a poor indicator within a group. Think about the Heart Data scatter plot...

Multiple Logistic Regression

Multiple Logistic Regression

It is simple to illustrate examples in logistic regression when there is just one predictors variable.

But the approach ‘easily’ generalizes to the situation where there are multiple predictors.

A lot of the same details as linear regression apply to logistic regression.

Interactions can be considered. Multicollinearity is a concern. So is overfitting.

Etc...

two predictors w/ high corr → interpretation not unique

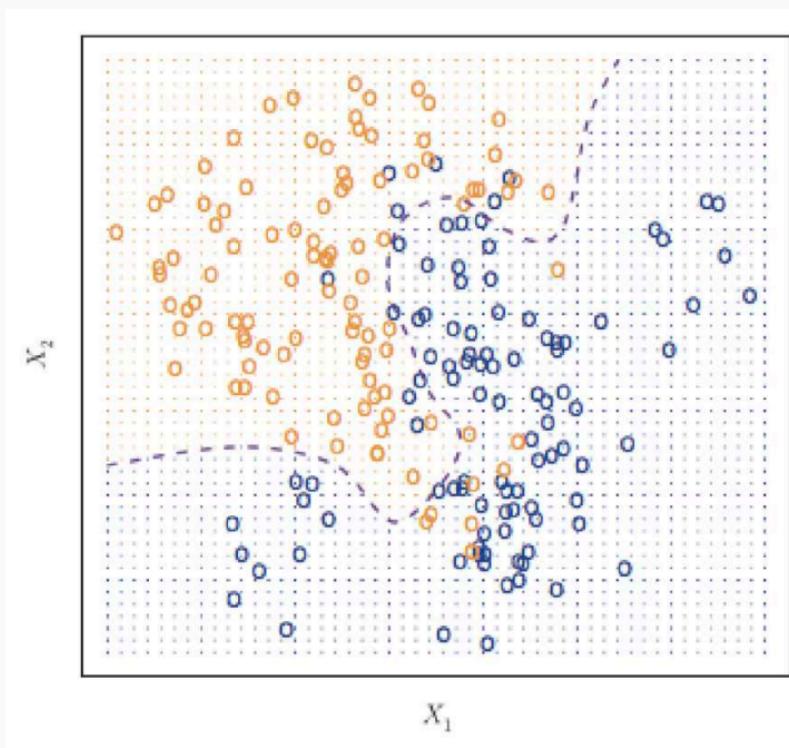
So how do we correct for such problems?

Regularization and checking though train, test, and cross-validation!

We will get into the details of this, along with other extensions of logistic regression, in the next lecture.

Classifier with two predictors

How can we estimate a classifier, based on logistic regression, for the following plot?



Multiple Logistic Regression

Earlier we saw the general form of *simple* logistic regression, meaning when there is just one predictor used in the model. What was the model statement (in terms of linear predictors)?

$$\cancel{\ln} \left(\frac{P(Y = 1)}{1 - P(Y = 1)} \right) = \beta_0 + \beta_1 X$$

Multiple logistic regression is a generalization to multiple predictors. More specifically we can define a multiple logistic regression model to predict $P(Y = 1)$ as such:

$$\cancel{\ln} \left(\frac{P(Y = 1)}{1 - P(Y = 1)} \right) = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \dots + \beta_p X_p$$

Fitting Multiple Logistic Regression

The estimation procedure is identical to that as before for simple logistic regression:

- a likelihood approach is taken, and the function is maximized across all parameters $\beta_0, \beta_1, \dots, \beta_p$ using an iterative method like Newton-Raphson or Gradient Descent.

The actual fitting of a Multiple Logistic Regression is easy using software (of course there's a python package for that) as the iterative maximization of the likelihood has already been hard coded.

In the `sklearn.linear_model` package, you just have to create your multidimensional design matrix X to be used as predictors in the `LogisticRegression` function.

Interpretation of Multiple Logistic Regression

Interpreting the coefficients in a multiple logistic regression is similar to that of linear regression.

Key: since there are other predictors in the model, the coefficient $\hat{\beta}_j$ is the association between the j^{th} predictor and the response (on log odds scale). But what do we have to say? _____.

We are trying to attribute the partial effects of each predictor controlling for the others (aka, controlling for possible *confounders*).

Y=HD ↑
X=mkt ↑

Interpreting Multiple Logistic Regression: an Example

Let's get back to the Heart Data. We are attempting to predict whether someone has HD based on MaxHR and whether the person is female or male. The simultaneous effect of these two predictors can be brought into one model.

Recall from earlier we had the following estimated models:

$$\log \left(\frac{\widehat{P(Y = 1)}}{1 - \widehat{P(Y = 1)}} \right) = 6.30 - 0.043 \cdot X_{MaxHR}$$

$$\log \left(\frac{\widehat{P(Y = 1)}}{1 - \widehat{P(Y = 1)}} \right) = -1.06 + 1.27 \cdot X_{gender}$$

Interpreting Multiple Logistic Regression: an Example

The results for the multiple logistic regression model are:

```
data_x = df_heart[['MaxHR', 'Sex']]
data_y = df_heart['AHD']

logreg = LogisticRegression(C=100000, fit_intercept=True)
logreg.fit(data_x, data_y);

print('Estimated beta1: \n', logreg.coef_)
print('Estimated beta0: \n', logreg.intercept_)
```

Estimated beta1:

~~MaxHR~~ [[-0.04496354 1.40079047]]
Estimated beta0:
[5.58662464]

$$\ln P \left(\frac{P(Y=1)}{P(Y=0)} \right) = 5.5 - 0.04 \text{MaxHR} + 1.4 \text{Sex}$$

Some questions

1. Estimate the odds ratio of HD comparing men to women using this model.
2. Is there any evidence of multicollinearity in this model?
Compare coll & mult. log & singl log. → no relation b/w predictors, the coeffs should not change
3. Is there any confounding in this problem?

Rest of the Lecture Outline

- Interpreting interactions in logistic regression
- Classification Boundaries
- Regularization in Logistic Regression
- Bayes Theorem and Misclassification Rates
- ROC Curves

Interactions in Multiple Logistic Regression

Just like in linear regression, interaction terms can be considered in logistic regression. An **interaction terms** is incorporated into the model the same way, and the interpretation is very similar (on the log-odds scale of the response of course).

Write down the model for the Heart data for the 2 predictors plus the interactions term based on the output on the next slide.

Interpreting Multiple Logistic Regression: an Example

The results for the multiple logistic regression model are:

```
df_heart['Age_Sex'] = df_heart.Age*df_heart.Sex  
  
data_x = df_heart[['Age', 'Sex', 'Age_Sex']]  
  
logit3 = LogisticRegression(penalty='none', max_iter = 1000)  
logit3.fit(data_x, df_heart['AHD'])  
  
print("Logistic Regression Estimated Betas (B0,B1,B2,B3):",  
      np.round(logit3.intercept_,4),np.round(logit3.coef_,5))
```

```
Logistic Regression Estimated Betas (B0,B1,B2,B3): [-4.2742] [[0.05654 0.77549 0.01273]]
```

Some questions

```
Logistic Regression Estimated Betas (B0,B1,B2,B3): [-4.2742] [[0.05654 0.77549 0.01273]]
```

1. Write down the complete model. Break this down into the model to predict log-odds of heart disease (HD) based on Age for women and the same model for men.

Some questions

Logistic Regression Estimated Betas (B0,B1,B2,B3): [2.90173899] [[-0.0265903 5.41531302 -0.02707269]]

2. Interpret the results of this model. What does the coefficient for the interaction term represent?

3. Estimate the odds ratio of HD comparing men to women using this model [trick question].

Classification Boundaries

Classification boundaries

Recall that we could attempt to purely classify each observation based on whether the estimated $P(Y = 1)$ from the model was greater than 0.5.

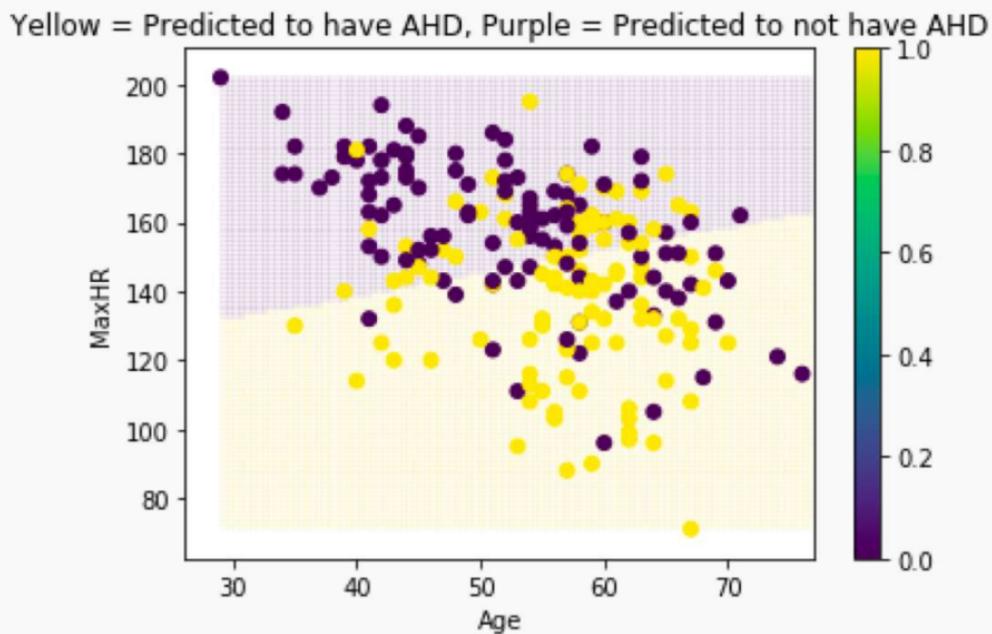
Recall, the logistic regression model statement from last week:

$$\ln\left(\frac{P(Y = 1)}{P(Y = 0)}\right) = \beta_0 + \beta_1 X_1 + \cdots + \beta_p X_p$$

If $P(Y = 1) = 0.5$, then the odds is 1, and the log-odds is 0. Thus the classification boundary that separates the estimated probabilities above 0.5 from below 0.5 is defined when we set: $\beta_0 + \beta_1 X_1 + \cdots + \beta_p X_p = 0$.

Thus logistic regression is said to lead to a **linear classification boundary**. There is actually a lot more freedom in logistic regression's classification boundary (the geometry is defined by the parameterization of your predictors!)

Linear Classification Boundary Example

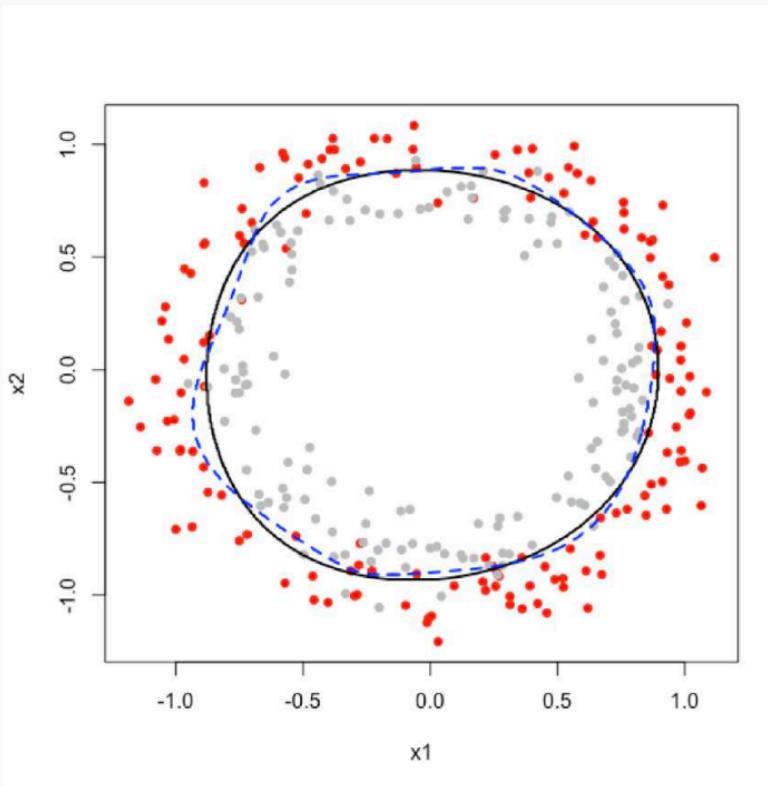


Classification Boundary: a complicated example

```
Logistic Regression Estimated Betas (B0,B1,B2,B3): [-4.2742] [[0.05654 0.77549 0.01273]]
```

4. Using the results of the model above, at what ages will males be predicted to have a heart attack in a classification? At what ages will females be predicted to have a heart attack?

Geometry of 2D Classification in Logistic Regression: an Example



2D Classification in Logistic Regression: an Example

Would a logistic regression model perform well in classifying the observations in this example?

What would be a good logistic regression model to classify these points?

Based on these predictors, two separate logistic regression model were considered that were based on different ordered polynomials of X_1, X_2 and their interactions. The ‘circles’ represent the boundary for classification.

How can the classification boundary be calculated for a logistic regression?

Regularization in Logistic Regression

Review: Regularization in Linear Regression

Based on the Likelihood framework, a loss function can be determined based on the log-likelihood function.

We saw in linear regression that maximizing the log-likelihood is equivalent to minimizing the sum of squares error:

$$\arg \min \sum_{i=1}^n (y_i - \hat{y}_i)^2 = \arg \min \sum_{i=1}^n (y_i - (\beta_0 + \beta_1 x_{1i} + \dots + \beta_p x_{pi}))^2$$

Review: Regularization in Linear Regression

And a regularization approach was to add a penalty factor to this equation. Which for Ridge Regression becomes:

$$\arg \min \left[\sum_{i=1}^n \left(y_i - \left(\beta_0 + \sum_{j=1}^n \beta_j x_{ji} \right) \right)^2 + \lambda \sum_{j=1}^n \beta_j^2 \right]$$

Note: this penalty *shrinks* the estimates towards zero, and had the analogue of using a Normal prior centered at zero in the Bayesian paradigm.

Recall: Loss function in Logistic Regression

A similar approach can be used in logistic regression. Here, maximizing the log-likelihood is equivalent to minimizing the following loss function:

$$\operatorname{argmin}_{\beta_0, \beta_1, \dots, \beta_p} \left[- \sum_{i=1}^n (y_i \ln(p_i) + (1 - y_i) \ln(1 - p_i)) \right]$$

$$\text{where } p_i = \frac{1}{1 - e^{-(\beta_0 + \beta_1 X_{1,i} + \dots + \beta_p X_{p,i})}}$$

Why is this a good loss function to minimize? Where does this come from?

The log-likelihood for independent $Y_i \sim \text{Bern}(p_i)$.

Regularization in Logistic Regression

A penalty factor can then be added to this loss function and results in a new loss function that penalizes large values of the parameters:

$$\operatorname{argmin}_{\beta_0, \beta_1, \dots, \beta_p} \left[- \sum_{i=1}^n (y_i \ln(p_i) + (1 - y_i) \ln(1 - p_i)) + \lambda \sum_{j=1}^p \beta_j^2 \right]$$

The result is just like in linear regression: shrink the parameter estimates towards zero.

In practice, the intercept is usually not part of the penalty factor.

Note: the sklearn package uses a different tuning parameter: instead of λ they use a constant that is essentially $C = \frac{1}{\lambda}$.

Regularization in Logistic Regression: an Example

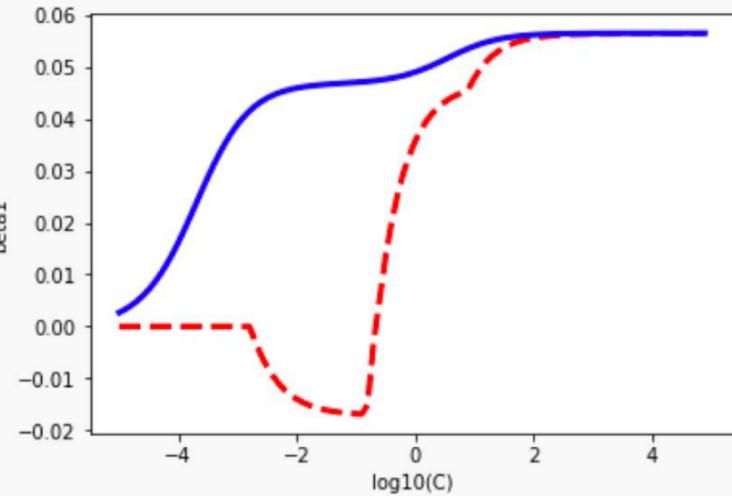
Let's see how this plays out in an example in logistic regression.

```
beta1_l1 = []
beta1_l2 = []
Cs = np.power(10,np.arange(0,10,0.1))/np.power(10,5)

data_x = df_heart[['Age', 'Sex', 'Age_Sex']]
data_y = df_heart['AHD']

for C in Cs:
    logitm_l1 = LogisticRegression(C = C, penalty = "l1",
                                    solver='liblinear',max_iter=200)
    logitm_l1.fit (data_x, data_y)
    logitm_l2 = LogisticRegression(C = C, penalty = "l2",
                                    solver='lbfgs', max_iter=200)
    logitm_l2.fit (data_x, data_y)
    beta1_l1.append(logitm_l1.coef_[0][0])
    beta1_l2.append(logitm_l2.coef_[0][0])

plt.plot(np.log10(Cs), beta1_l1, "--", color='red', lw=3)
plt.plot(np.log10(Cs), beta1_l2, color='blue', lw=3)
plt.xlabel ("log10(C)")
plt.ylabel("beta1")
plt.show()
```



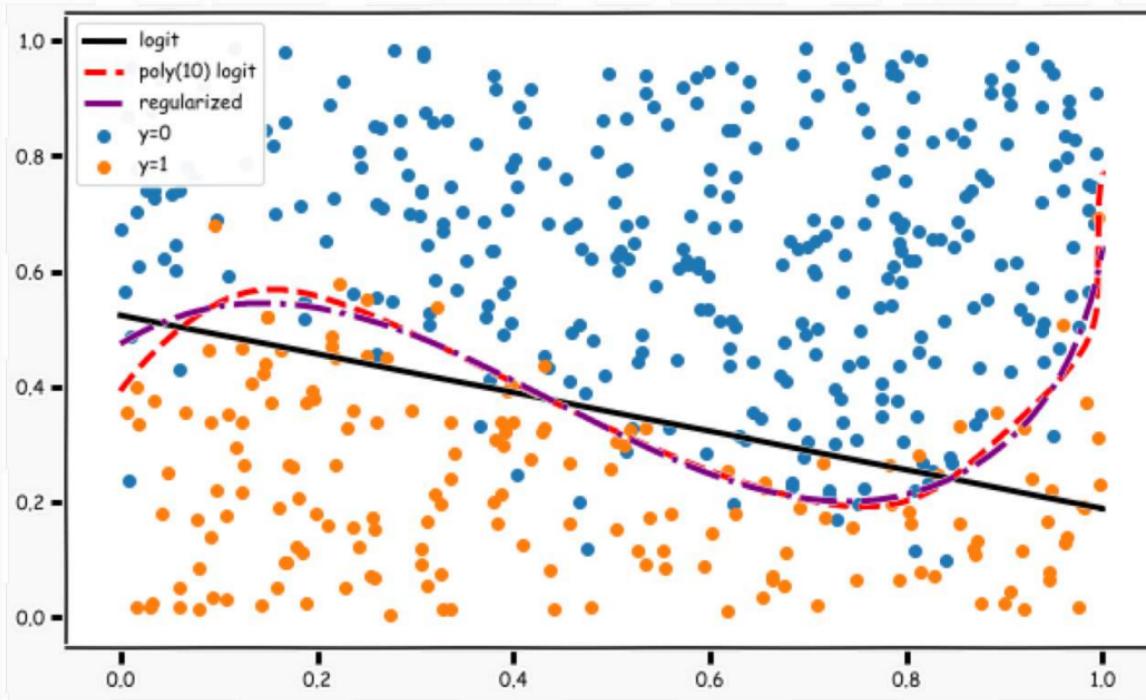
Regularization in Logistic Regression: tuning λ

Just like in linear regression, the shrinkage factor must be chosen.
How should we go about doing this?

Through building multiple training and test sets (hooray, cross-validation!), we can select the best shrinkage factor to mimic out-of-sample prediction.

How could we measure how well each model fits the validation sets?
We could measure this based on some loss function (more coming later)!

Regularized Decision Boundaries



Bayes Theorem, Misclassification Rates,
False Positives and Negatives

Bayes' Theorem

We defined conditional probability as:

$$P(B|A) = P(B \text{ and } A)/P(A)$$

And using the fact that $P(B \text{ and } A) = P(A|B)P(B)$ we get the simplest form of Bayes' Theorem:

$$P(B|A) = \frac{P(A|B)P(B)}{P(A)}$$

Another version of Bayes' Theorem is found by substituting in the Law of Total Probability (LOTP) into the denominator:

$$P(B|A) = \frac{P(A|B)P(B)}{P(A|B)P(B) + P(A|B^C)P(B^C)}$$

Where have we seen Bayes' Theorem before? Why do we care?

Diagnostic Testing

In the diagnostic testing paradigm, one cares about whether the results of a test (like a classification test) matches truth (the true class that observation belongs to). The simplest version of this is trying to detect disease ($D+$ vs. $D-$) based on a diagnostic test ($T+$ vs. $T-$).

Medical examples of this include various screening tests: breast cancer screening through (i) self-examination and (ii) mammography, prostate cancer screening through (iii) PSA tests, and Colo-rectal cancer through (iv) colonoscopies.

These tests are a little controversial because of poor predictive probability of the tests.

Diagnostic Testing (cont.)

Bayes' theorem can be rewritten for diagnostic tests:

$$P(D+|T+) = \frac{P(T+|D+)P(D+)}{P(T+|D+)P(D+) + P(T+|D-)P(D-)}$$

These probability quantities can then be defined as:

- *Sensitivity*: $P(T+|D+)$
- *Specificity*: $P(T-|D-)$
- *Prevalence*: $P(D+)$
- *Positive Predictive Value*: $P(D+|T+)$
- *Negative Predictive Value*: $P(D-|T-)$

How do positive and negative predictive values relate? Be careful...

Diagnostic Testing

We mentioned that these tests are a little controversial because of their poor predictive probability. When will these tests have poor positive predictive probability?

When the disease is not very prevalent, then the number of 'false positives' will overwhelm the number of true positive. For example, PSA screening for prostate cancer has sensitivity of about 90% and specificity of about 97% for some age groups (men in their fifties), but prevalence is about 0.1%.

What is positive predictive probability for this diagnostic test?

Why do we care?

As data scientists, why do we care about diagnostic testing from the medical world? (hint: it's not just because Kevin is a trained biostatistician!)

Because classification can be thought of as a diagnostic test. Let $Y_i = k$ be the event that observation i truly belongs to category k , and let $\hat{Y}_i = k$ the event that we correctly predict it to be in class k . Then Bayes' rule states that our *Positive Predictive Value* for classification is:

$$P(Y_i = k|\hat{Y}_i = k) = \frac{P(\hat{Y}_i = k|Y_i = k)P(Y_i = k)}{P(\hat{Y}_i = k|Y_i = k)P(Y_i = k) + P(\hat{Y}_i = k|Y_i \neq k)P(Y_i \neq k)}$$

Thus the probability of a predicted outcome truly being in a specific group depends on what? The proportion of observations in that class!

Error in Classification

There are 2 major types of error in classification problems based on a binary outcome. They are:

False positives: incorrectly predicting $\hat{Y} = 1$ when it truly is in $Y = 0$.

False negative: incorrectly predicting $\hat{Y} = 0$ when it truly is in $Y = 1$.

The results of a classification algorithm are often summarized in two ways: (1) a **confusion matrix**, sometimes called a **contingency table**, or a 2×2 table (more generally $(k \times k)$ table) and (2) a receiver operating characteristics (ROC) curve.

Confusion matrix

When a classification algorithm (like logistic regression) is used, the results can be summarized in a ($k \times k$) table as such:

	Predicted no AHD ($\hat{Y} = 0$)	Predicted AHD ($\hat{Y} = 1$)
Truly no AHD ($Y = 0$)	110	54
Truly AHD ($Y = 1$)	53	86

The table above was a classification based on a logistic regression model to predict AHD based on “3” predictors: X_1 = Age, X_2 = Sex, and X_3 = interaction between Age and Sex.

What are the false positive and false negative rates for this classifier?

Bayes' Classifier Choice

A classifier's error rates can be tuned to modify this table. How?

The choice of the Bayes' classifier level will modify the characteristics of this table.

If we thought it was more important to predict AHD patients correctly (fewer false negatives), what could we do for our Bayes' classifier level?

We could classify instead based on:

$$\hat{P}(Y = 1) > \pi$$

and we could choose π to be some level other than 0.50.

Let's see what the table looks like if π were 0.40 or 0.60 instead. What should happen to the False Positive and False Negative frequencies?

Other Confusion tables

Based on $\pi = 0.4$:

	Predicted no AHD ($\hat{Y} = 0$)	Predicted AHD ($\hat{Y} = 1$)
Truly no AHD ($Y = 0$)	93	71
Truly AHD ($Y = 1$)	38	101

What has improved? What has worsened?

Based on $\pi = 0.6$:

	Predicted no AHD ($\hat{Y} = 0$)	Predicted AHD ($\hat{Y} = 1$)
Truly no AHD ($Y = 0$)	138	26
Truly AHD ($Y = 1$)	74	65

Which should we choose? Why?

ROC Curves and Area-Under-the-Curve

ROC Curves

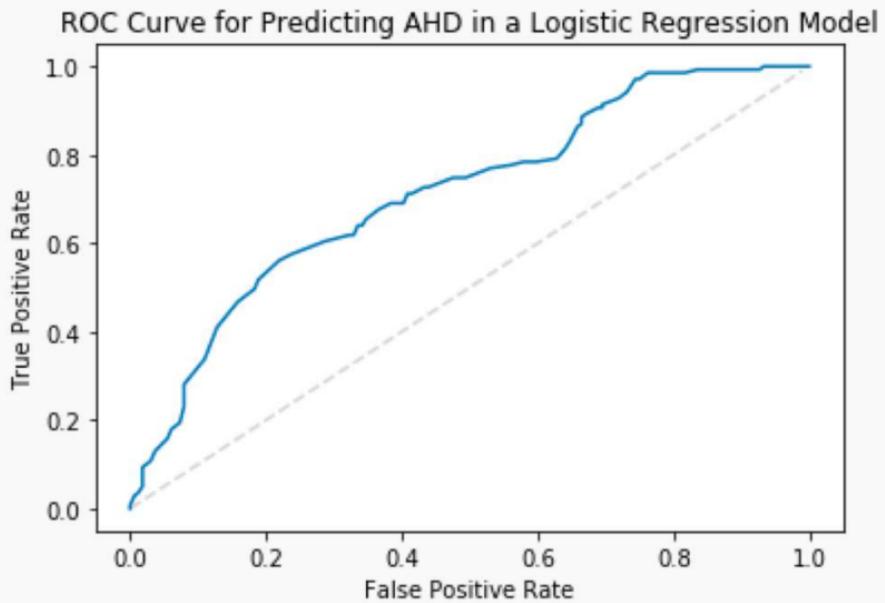
The Radio Operator Characteristics (ROC) curve illustrates the trade-off for all possible thresholds chosen for the two types of error (or correct classification).

The vertical axis displays the true positive predictive value and the horizontal axis depicts the true negative predictive value.

What is the shape of an ideal ROC curve?

See next slide for an example.

ROC Curve Example



AUC for measuring classifier performance

The overall performance of a classifier, calculated over all possible thresholds, is given by the **area under the ROC curve** (AUC).

An ideal ROC curve will hug the top left corner, so the larger the AUC the better the classifier.

What is the worst case scenario for AUC? What is the best case? What is AUC if we independently just flip a [biased] coin to perform classification?

This AUC then can be used to compare various approaches to classification: Logistic regression, k -NN, Decision Trees (to come), etc.