GENERAL ASSEMBLY

DATA SCIENCE

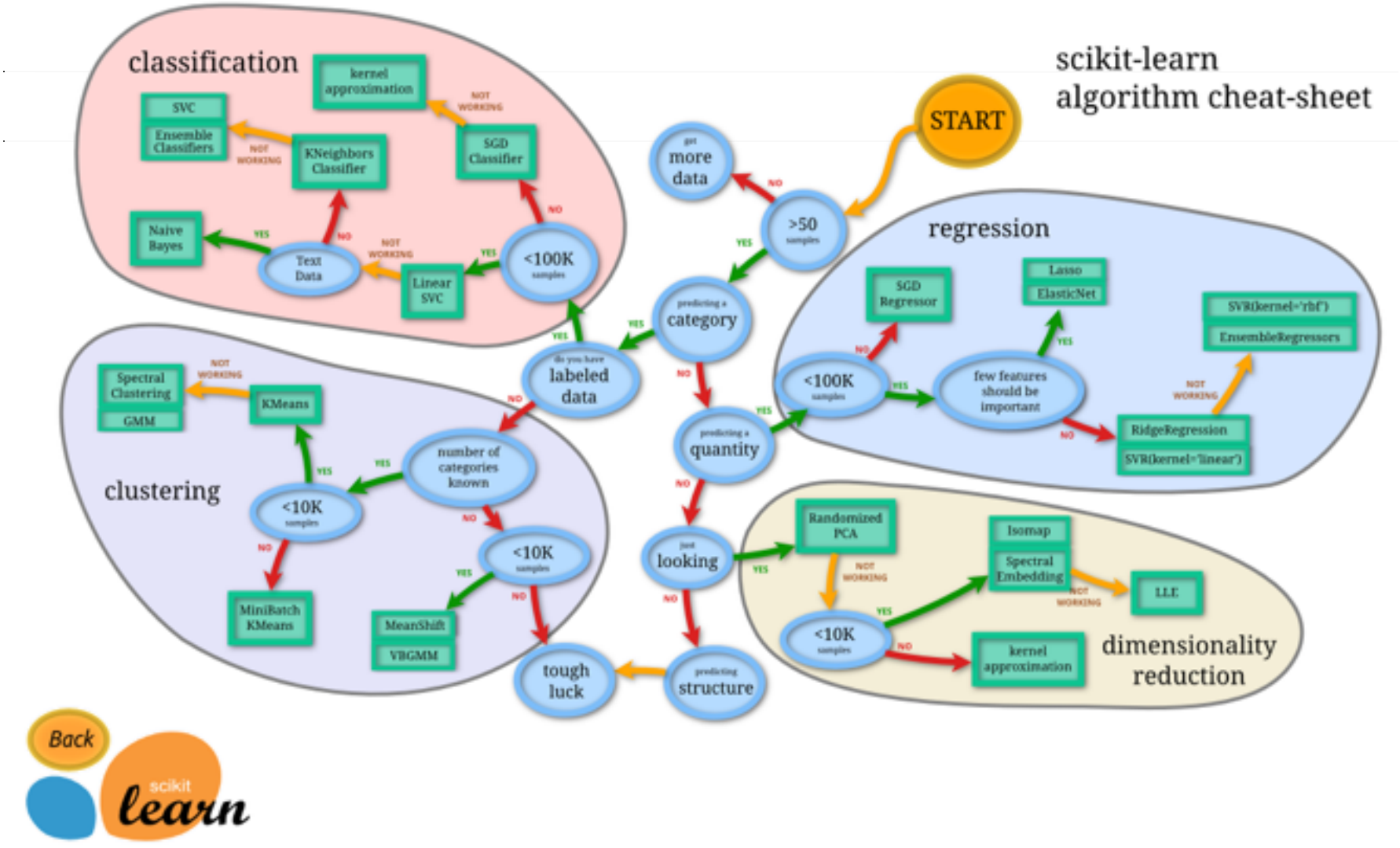
SYD DAT 6

Week 3 – Logistic Regression
Monday 24th October

1. Classification
2. What is Logistic Regression?
3. Why use Logistic Regression
4. Lab
5. Homework Review

DATA SCIENCE PART TIME COURSE

CLASSIFICATION


scikit-learn
algorithm cheat-sheet
START
classification
kernel approximation
NOT WORKING
SVC
Ensemble Classifiers
NOT WORKING
KNeighbors Classifier
SGD Classifier
NO
Naive Bayes
YES
Text Data
NO
NOT WORKING
Linear SVC
YES
<100K samples
NO
get more data
NO
>50 samples
YES
regression
SGD Regressor
Lasso
ElasticNet
SVR(kernel='rbf')
EnsembleRegressors
NO
<100K samples
YES
few features should be important
YES
NOT WORKING
RidgeRegression
NO
SVR(kernel='linear')
predicting a category
YES
NO
do you have labeled data
YES
NO
predicting a quantity
YES
NO
clustering
Spectral Clustering
NOT WORKING
GMM
KMeans
YES
number of categories known
YES
NO
<10K samples
NO
MiniBatch KMeans
<10K samples
YES
MeanShift
VBGMM
NO
just looking
YES
NO
dimensionality reduction
Randomized PCA
NOT WORKING
Isomap
Spectral Embedding
NOT WORKING
LLE
YES
<10K samples
NO
kernel approximation
tough luck
predicting structure
Back
scikit learn

If the y variable is numeric then we have a regression problem - we are trying to predict a continuous number

If the y variable is a category (for example trying to predict a type of flower) the we have a classification problem - we are trying to classify what group that y belongs to.

$$Y = \begin{cases} 0 & \text{if No} \\ 1 & \text{if Yes.} \end{cases}$$

WHAT IS LOGISTIC REGRESSION?

We want to build a classifier that correctly identifies which class our target variable y belongs to given our input variable x.

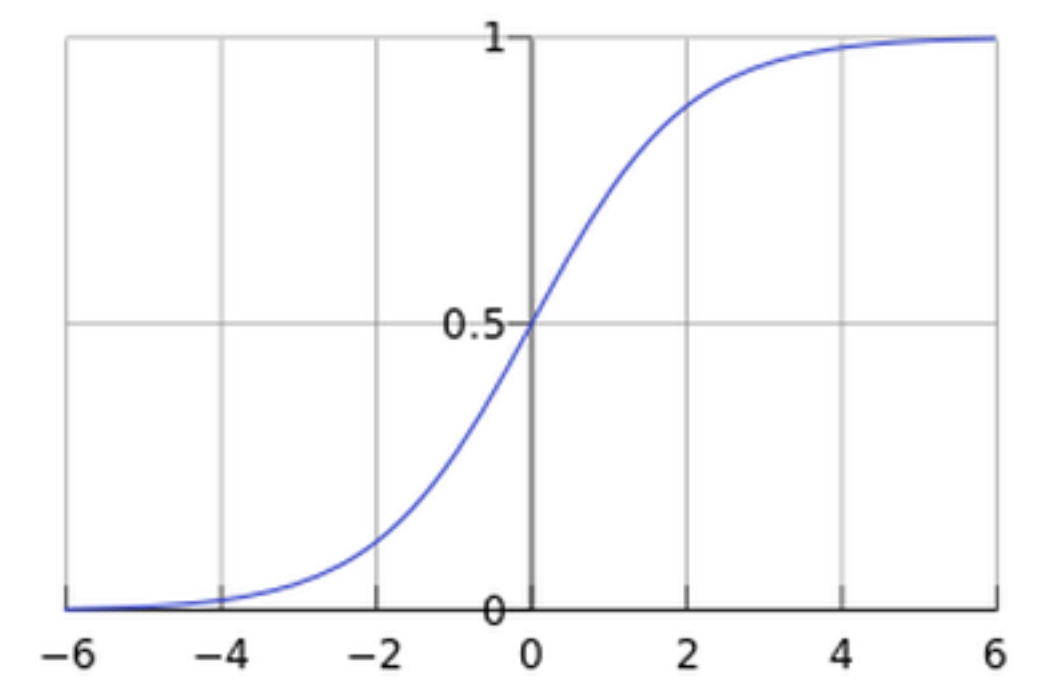
Why not use the linear regression model?

$$y = X\beta + \epsilon$$

- If we only have a binary response variable (0 or 1) it might make sense… BUT we can have our estimated value of $y > 1$ or $y < 0$ … which doesn't make sense.
- What of the case where we have more than one class? Linear regression cannot easily handle these cases.
- We want a classification method that can handle these cases and give us results we can easily interpret.

$$p(Y=1|X) = \beta_0 + \beta_1 X.$$

- This is a good starting point but we still have the problem of $p(Y)$ being outside the 0,1 range.
- We need to model $p(Y=1|X)$ using a function that gives outputs between 0 and 1.
- Basically we want something that looks like the following



Standard logistic sigmoid function

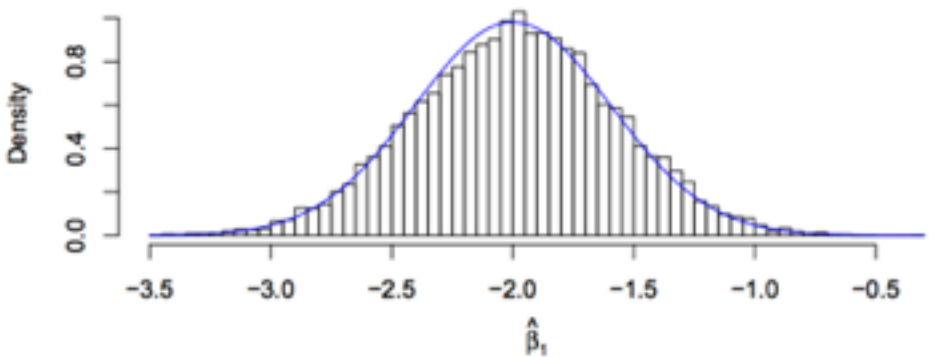
$$\log\left(\frac{p}{1-p}\right) = \beta_0 + \beta_1 x$$

- This is the logit function,
- We can see that it this function is linear in X
- $\frac{p}{1-p}$ is called the 'odds' and can be any value from 0 to ∞
- $\log\left(\frac{p}{1-p}\right)$ is called the 'log-odds' or 'logit'

Ordinary Least Squares does not work now that the function includes the logic transformation

A common method is Maximum Likelihood:

This likelihood gives the probability of the observed zeros and ones in the data. We pick β0 and β1 to maximise the likelihood of the observed data.



DATA SCIENCE PART TIME COURSE

MULTICLASS LOGISTIC REGRESSION

- Also known as ‘multinomial’ logistic regression.
- Concepts applies to other classification algorithms.

One vs Rest

DEMONSTRATION

SYNCHING YOUR FORK WITH THE COURSE REPO

1. re-name your labs with lab_name.<yourname>.ipynb (to prevent a conflict)
2. cd <path to the root of your SYD_DAT_6 local repo>
3. commit your changes ahead of sync
   - git status
   - git add .
   - git commit -m "descriptive label for the commit"
   - git status
4. download new material from official course repo (upstream) and merge it
   - git checkout master (ensures you are in the master branch)
   - git fetch upstream
   - git merge upstream/master

DATA SCIENCE PART TIME COURSE

CONFUSION MATRIX

Confusion Matrix: table to describe the performance of a classifier

n=165	Predicted: NO	Predicted: YES
Actual: NO	50	10
Actual: YES	5	100

Example: Test for presence of disease

NO = negative test = False = 0

YES = positive test = True = 1

- *How many classes are there?*
- *How many patients?*
- *How many times is disease predicted?*
- *How many patients actually have the disease?*

$$\text{sensitivity} = \frac{\text{number of true positives}}{\text{number of true positives} + \text{number of false negatives}}$$

$$\text{specificity} = \frac{\text{number of true negatives}}{\text{number of true negatives} + \text{number of false positives}}$$

n=165	Predicted: NO	Predicted: YES	
Actual: NO	TN = 50	FP = 10	60
Actual: YES	FN = 5	TP = 100	105
	55	110	

False Positive Rate:

- *When actual value is **negative**, how often is prediction **wrong**?*
- *FP / actual no = 10/60 = 0.17*

Sensitivity:

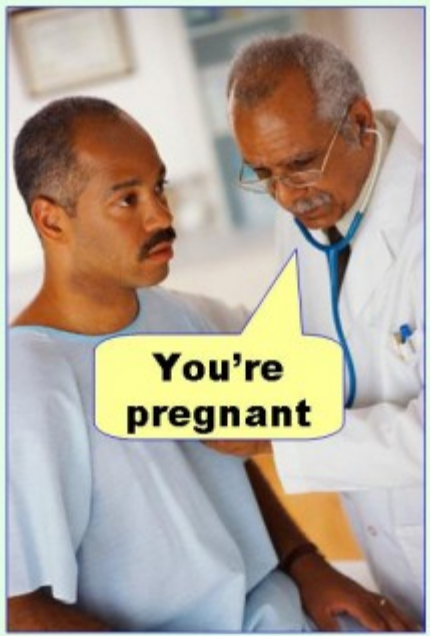
- *When actual value is **positive**, how often is prediction **correct**?*
- *TP / actual yes = 100/105 = 0.95*
- *"True Positive Rate" or "Recall"*

Specificity:

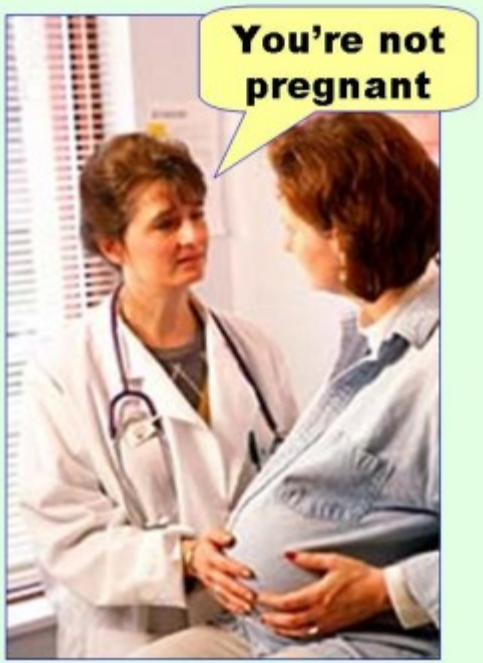
- *When actual value is **negative**, how often is prediction **correct**?*
- *TN / actual no = 50/60 = 0.83*

		Predicted condition			
	Total population	Predicted Condition positive	Predicted Condition negative	Prevalence $= \frac{\Sigma \text{ Condition positive}}{\Sigma \text{ Total population}}$	
True condition	condition positive	**True positive**	**False Negative** (Type II error)	True positive rate (TPR), Sensitivity, Recall $= \frac{\Sigma \text{ True positive}}{\Sigma \text{ Condition positive}}$	False negative rate (FNR), Miss rate $= \frac{\Sigma \text{ False negative}}{\Sigma \text{ Condition positive}}$
	condition negative	**False Positive** (Type I error)	**True negative**	False positive rate (FPR), Fall-out $= \frac{\Sigma \text{ False positive}}{\Sigma \text{ Condition negative}}$	True negative rate (TNR), Specificity (SPC) $= \frac{\Sigma \text{ True negative}}{\Sigma \text{ Condition negative}}$
	Accuracy (ACC) $= \frac{\Sigma \text{ True positive} + \Sigma \text{ True negative}}{\Sigma \text{ Total population}}$	Positive predictive value (PPV), Precision $= \frac{\Sigma \text{ True positive}}{\Sigma \text{ Test outcome positive}}$	False omission rate (FOR) $= \frac{\Sigma \text{ False negative}}{\Sigma \text{ Test outcome negative}}$	Positive likelihood ratio (LR+) $= \frac{\text{TPR}}{\text{FPR}}$	Diagnostic odds ratio (DOR) $= \frac{\text{LR+}}{\text{LR−}}$
		False discovery rate (FDR) $= \frac{\Sigma \text{ False positive}}{\Sigma \text{ Test outcome positive}}$	Negative predictive value (NPV) $= \frac{\Sigma \text{ True negative}}{\Sigma \text{ Test outcome negative}}$	Negative likelihood ratio (LR−) $= \frac{\text{FNR}}{\text{TNR}}$	

Type I error
(false positive)



Type II error
(false negative)



LAB

DATA SCIENCE – Week 3 Day 2

DISCUSSION TIME

- Review of last week
- Further Reading for Logistic Regression
- Check in with homework/course project

DATA SCIENCE – Week 3 Day 2

DISCUSSION TIME

An Introduction to Statistical Learning

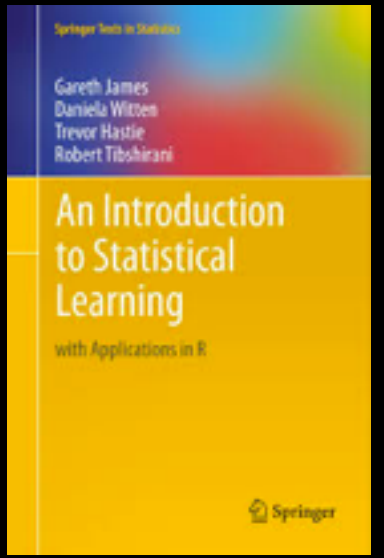
- Chapter 4 – Logistic Regression

Logistic Regression applied to loan applications

- https://github.com/nborwankar/LearnDataScience

Odds Ratio in Logistic Regression

- http://www.ats.ucla.edu/stat/mult_pkg/faq/general/odds_ratio.htm



DISCUSSION TIME

Homework/Course Project

- Work on course project ideas
- Read