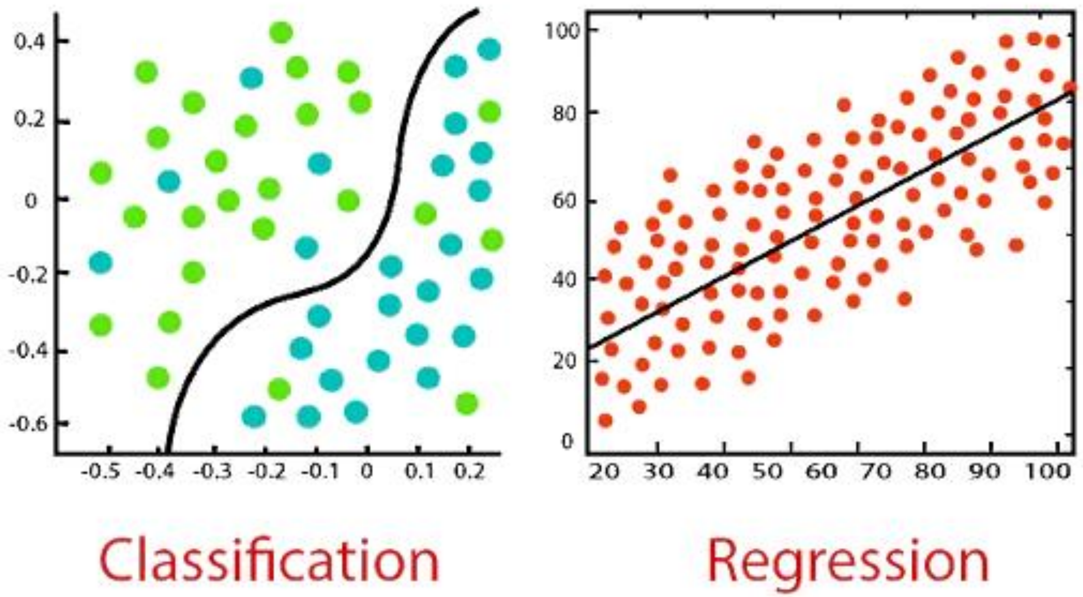
1. Review of classification
2. Classification metrics
3. k-NN, SVM, NB
4. Spam filter model!

Classification Models

DUKE
FUQUA
SCHOOL OF BUSINESS

Recap from Class/Module 1: In many settings, we want to estimate the probability of something happening

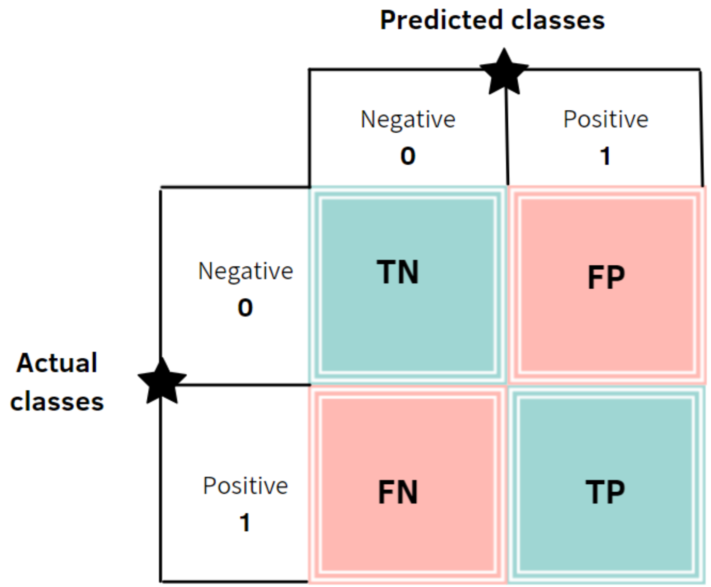
- This relates to classification (or *class probability estimation*).



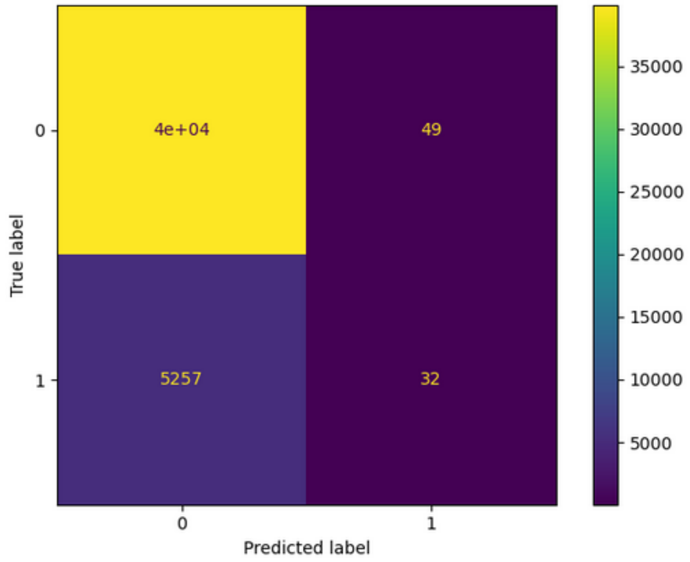
- Today, we will discuss some widely used classification algorithms

Confusion matrices summarize classification performance

Picture of confusion matrix:



From `sklearn` (bank data example):



- *Warning:* some people flip axes (true = horizontal, predicted = vertical)
- Classifier Accuracy = $\frac{\text{TN+TP}}{\text{TN+FN+TP+FP}}$ = (correct classifications)/(all classifications)

A targeted marketing example

- Your data analysts have developed two classification models (A and B) to predict which customers will buy a new product (balanced)
- On a training data set of 1,000 samples, the models produce the confusion matrices below.
- Which model is better?

Accuracy = 80%

Model A

True Label \ Prediction Label	0	1
0	300	200
1	0	500

false positives

Accuracy = 80%

Model B

True Label \ Prediction Label	0	1
0	500	0
1	200	300

false negatives

Example continued

- Suppose within the population that only 10% of customers actually buy the product
- On a data set of 1,000 samples representative of this population, the models produce the confusion matrices below.
- Now which model is better?

Model A Accuracy = 64%

True Label \ Prediction Label	0	1
0	540	360
1	0	100

false positives

Model B Accuracy = 96%

True Label \ Prediction Label	0	1
0	900	0
1	40	60

false negatives

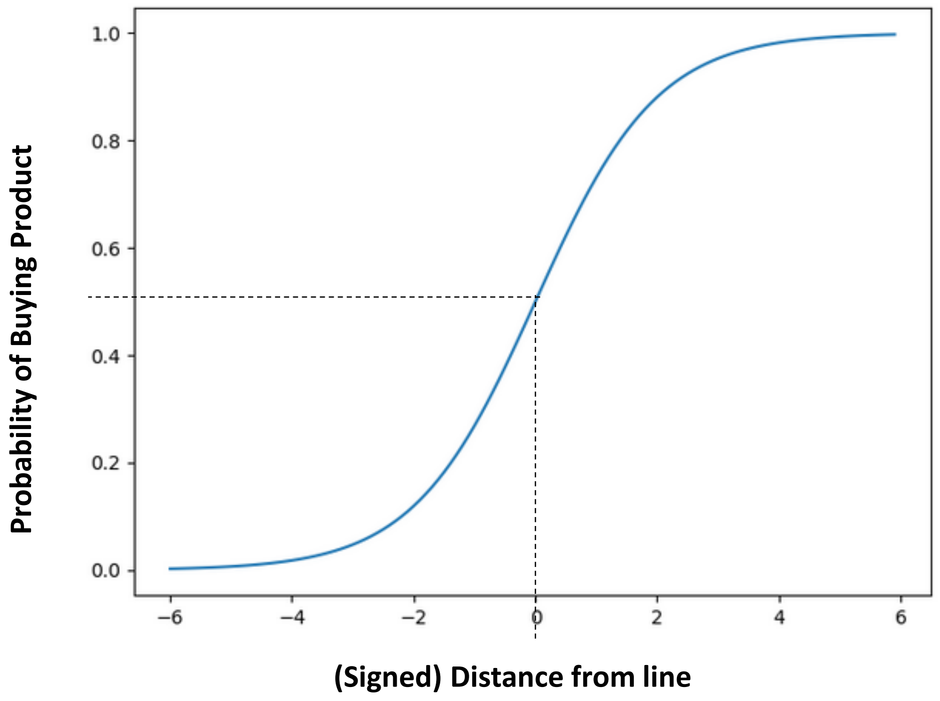
David's Model

	0	1
0	900	0
1	100	0

Accuracy = 90%

Classification scores

- Classification models usually provide a *score* for each set of features (e.g., a potential customer in the targeted marketing example)
- In many settings, these scores are useful to rank alternatives
- Logistic regression illustration:



Using classification models to rank alternatives

- If our estimates of both (a) the class probabilities from a model and (b) costs (or profits) are accurate, making decisions using a classification model is usually straightforward:
    - E.g., "mail a catalog to the customer whenever the expected profit of doing so is positive"
    - E.g., "offer medical treatment whenever the expected benefit (e.g., in QALYs) is positive"
- Even when this is not true, however, a classification model may nonetheless provide valuable information
- We can use a *single* classification model to generate a continuum of rules:
    - "Predict a 1 (e.g., buy) whenever the classification score is larger than a threshold T. Otherwise, predict a 0 (e.g., no buy)."
    - We can see how performance changes as we vary the threshold T

“ROC” curves provide a visualization of classification performance as we change the threshold

- Example from Module 1: bank data

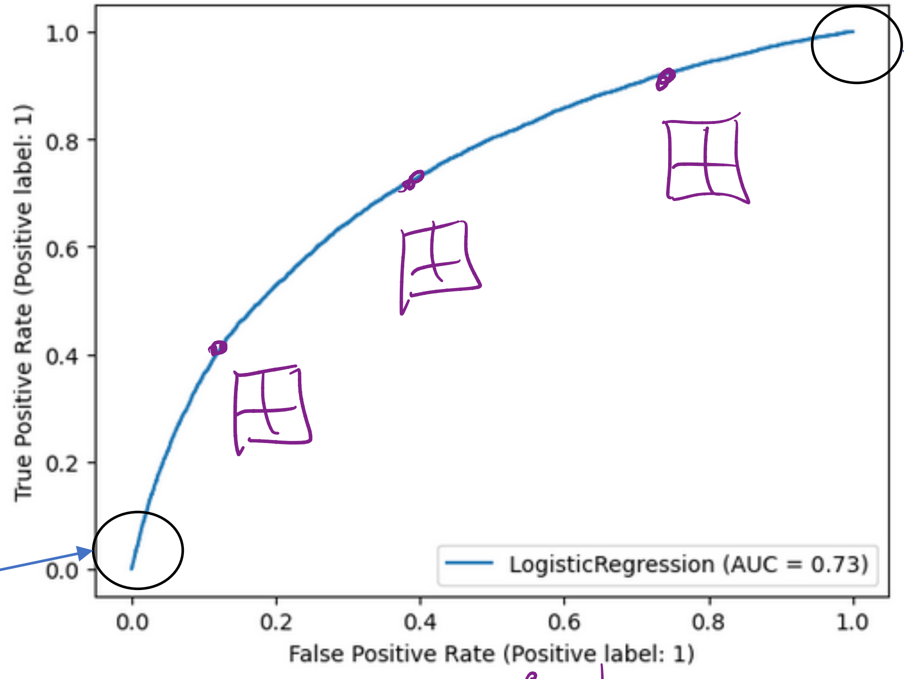
	age	job	marital	education	default	balance	housing	loan	contact	campaign	previous	y
0	58	management	married	tertiary	no	2143	yes	no	unknown	1	0	no
1	44	technician	single	secondary	no	29	yes	no	unknown	1	0	no
2	33	entrepreneur	married	secondary	no	2	yes	yes	unknown	1	0	no
3	47	blue-collar	married	unknown	no	1506	yes	no	unknown	1	0	no
4	33	unknown	single	unknown	no	1	no	no	unknown	1	0	no

- ROC curve for the logistic regression model:

```
_ = RocCurveDisplay.from_estimator(logistic_regression_sklearn, X, Y)
```

sensitivity (=recall)

(logistic thresh = 0)



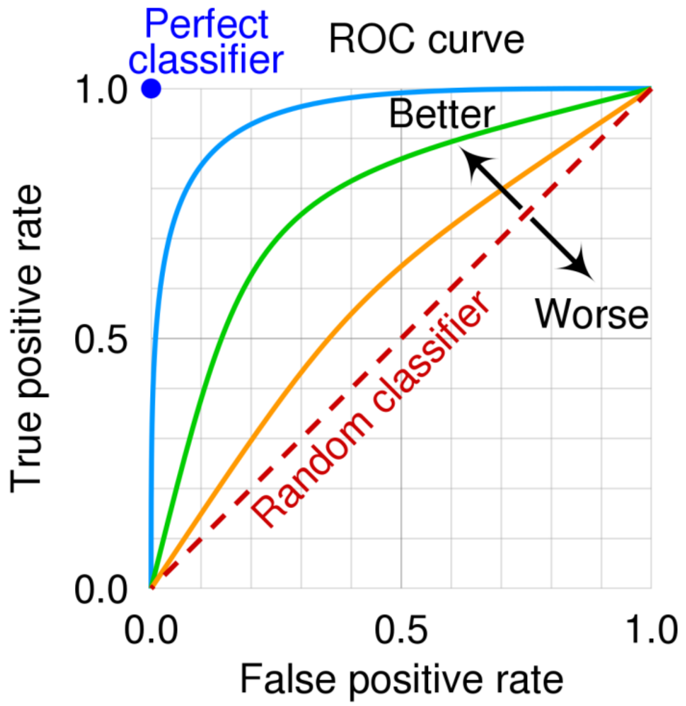
1-specificity

Threshold (T) = $-\infty$
Classify everything as a positive (1), i.e., nothing as a negative (0)

(logistic thresh = 1)

Threshold (T) = $+\infty$
Classify nothing as a positive (1), i.e., everything as a negative (0)

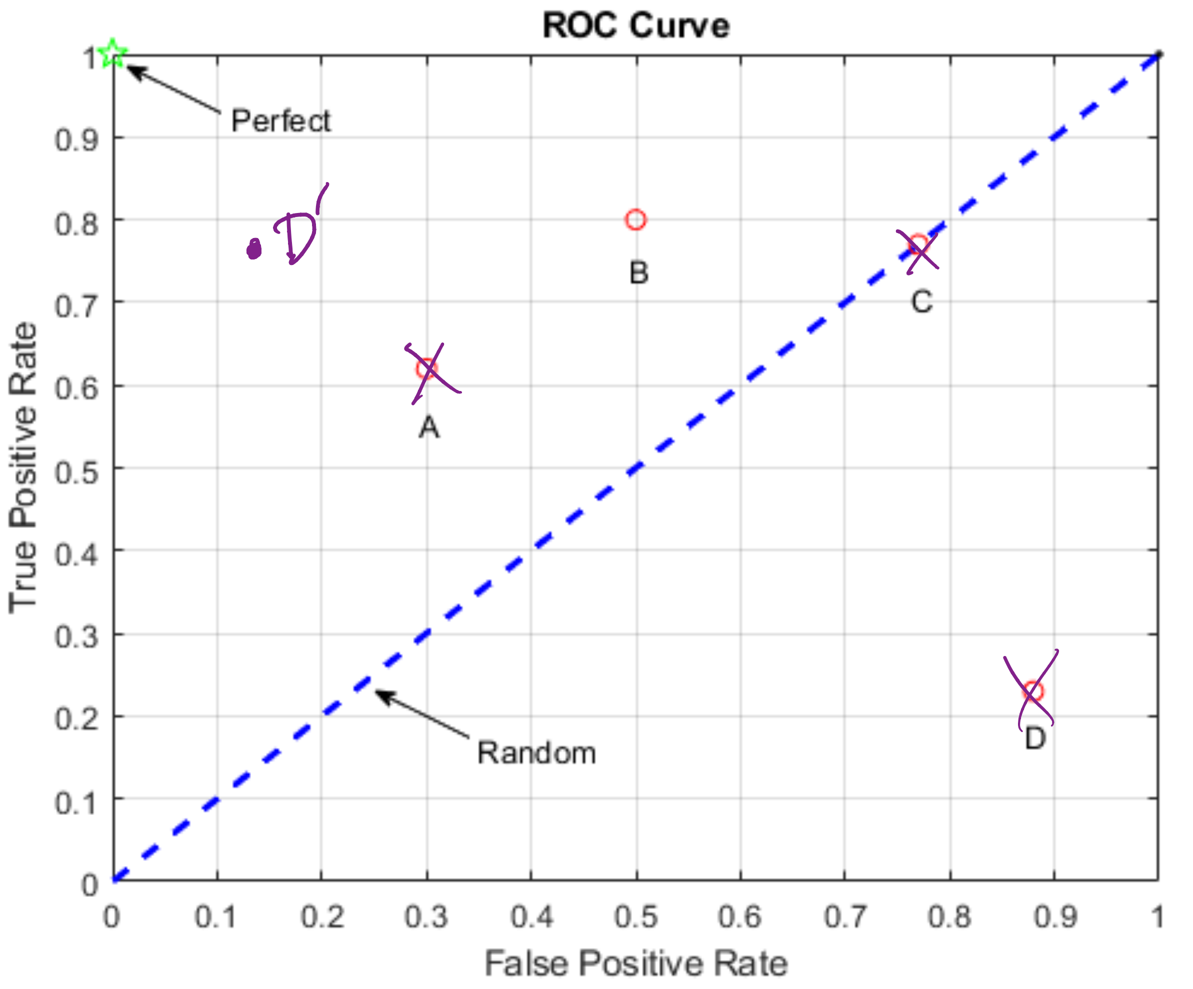
Properties of ROC curves



Source: Wikipedia

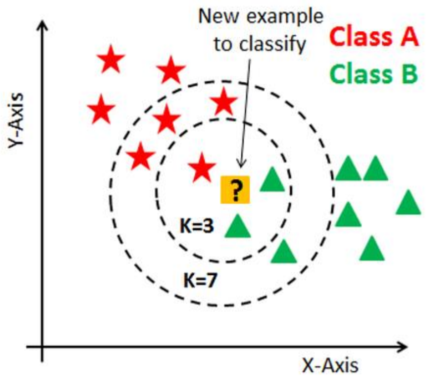
- ROC curves do not depend on the baseline positive rate
- The *area under the curve (AUC)* is a common summary statistic:
  - All else equal, a higher AUC is better
  - The AUC is equivalent to the probability that a randomly chosen positive (1) instance is ranked higher than a randomly chosen negative (0) instance

Which classification model would you choose?



Nearest neighbor ("k-NN") classification

- Nearest neighbor classification is an example of a *non-parametric predictive model*
- Basic idea:
  - Choose a positive integer k, which ranges from 1 to N (= size of training set)
  - Classification rule for a new instance: predict the majority target label from the k training points closest to the new instance



Source: https://ai.plainenglish.io/introduction-to-k-nearest-neighbors-knn-algorithm-e8617a448fa8

- How to choose k?
- Incredibly simple yet remarkably flexible! (Some caveats, though…)

A joke related to k-NN…

Chet Haase
@chethaase

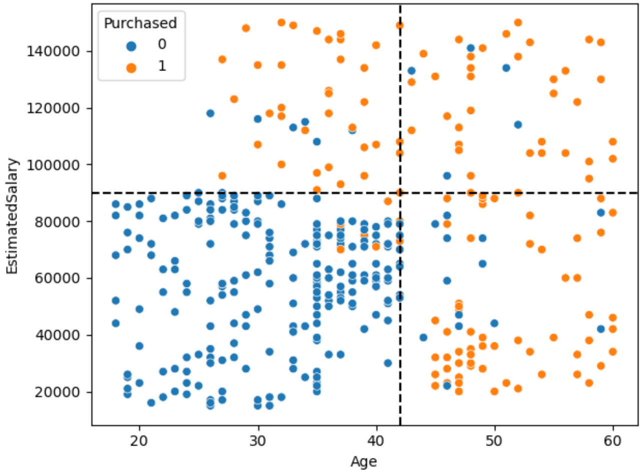
A Machine Learning algorithm walks into a bar.
The bartender asks, "What'll you have?"
The algorithm says, "What's everyone else having?"

9:24 AM · Nov 1, 2017

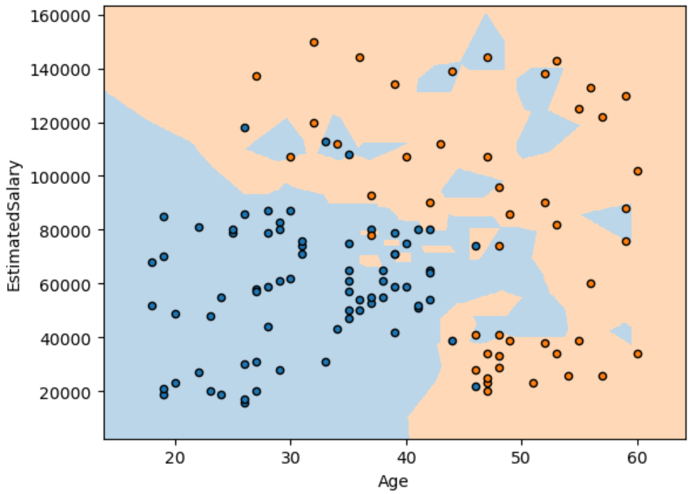
3,833 Retweets **218** Quote Tweets **7,850** Likes

k-NN on the social ads data

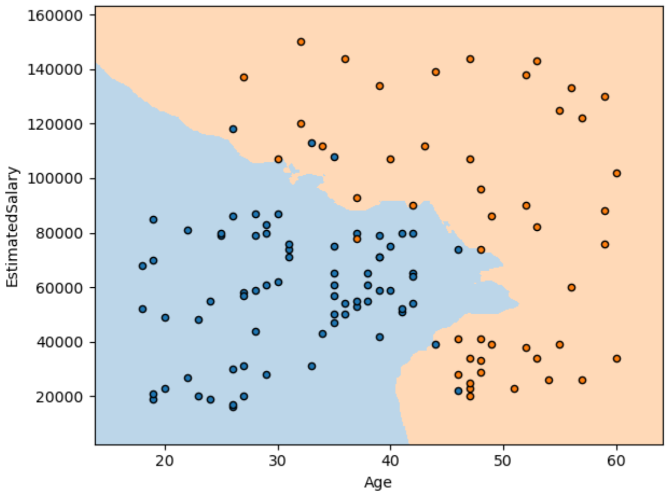
- Original data:



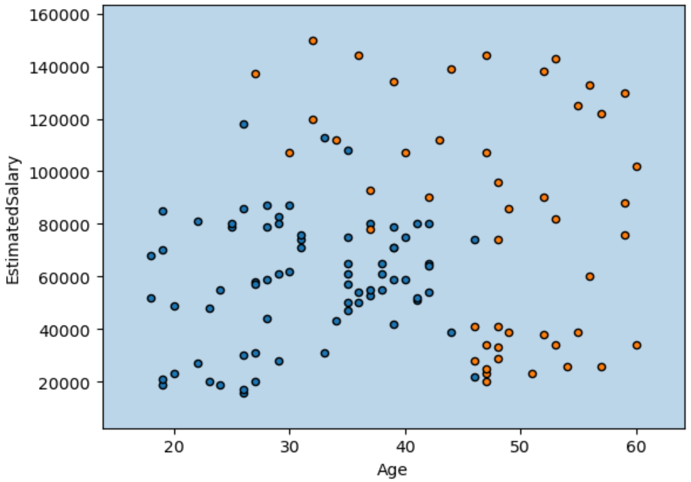
- k-NN results on test data:



k = 1



k = 9



k = 280

Pros and cons of nearest neighbor methods

- “Incredibly simple yet remarkably flexible! (Some caveats, though…)”
  - Three slides ago

- The caveats:

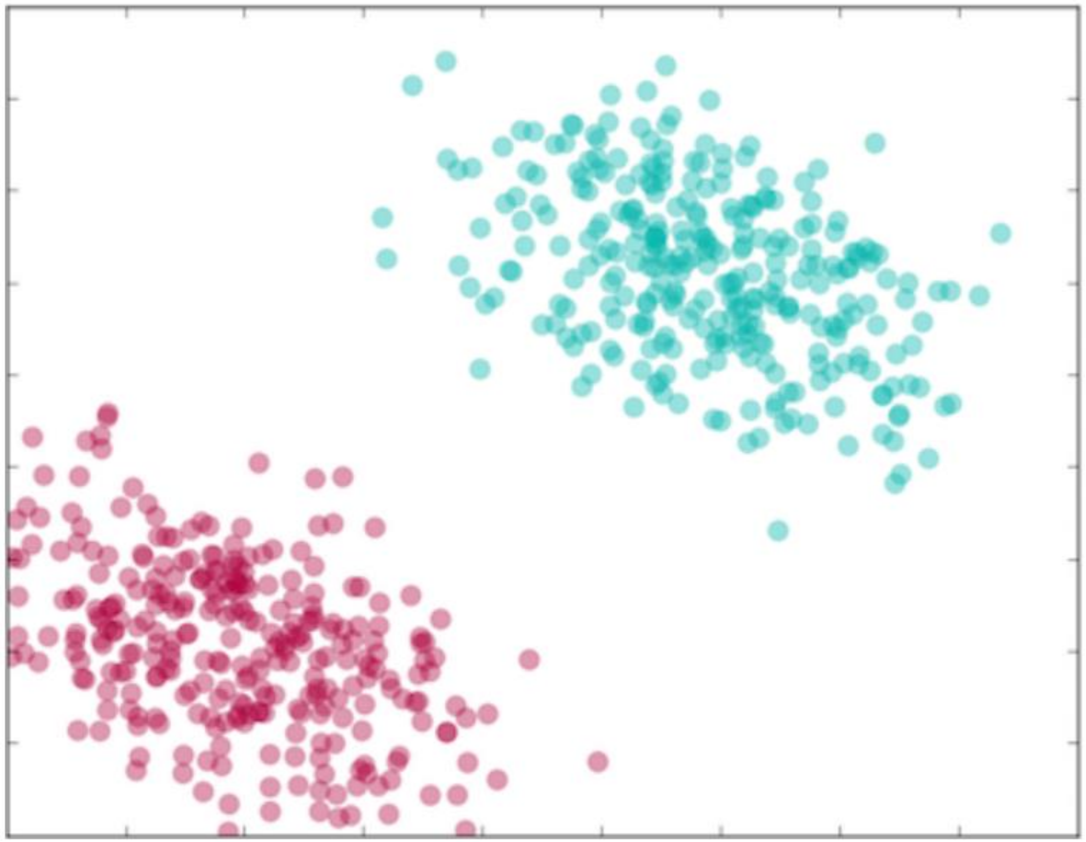
  1. Interpretability

  2. What does “distance” mean?

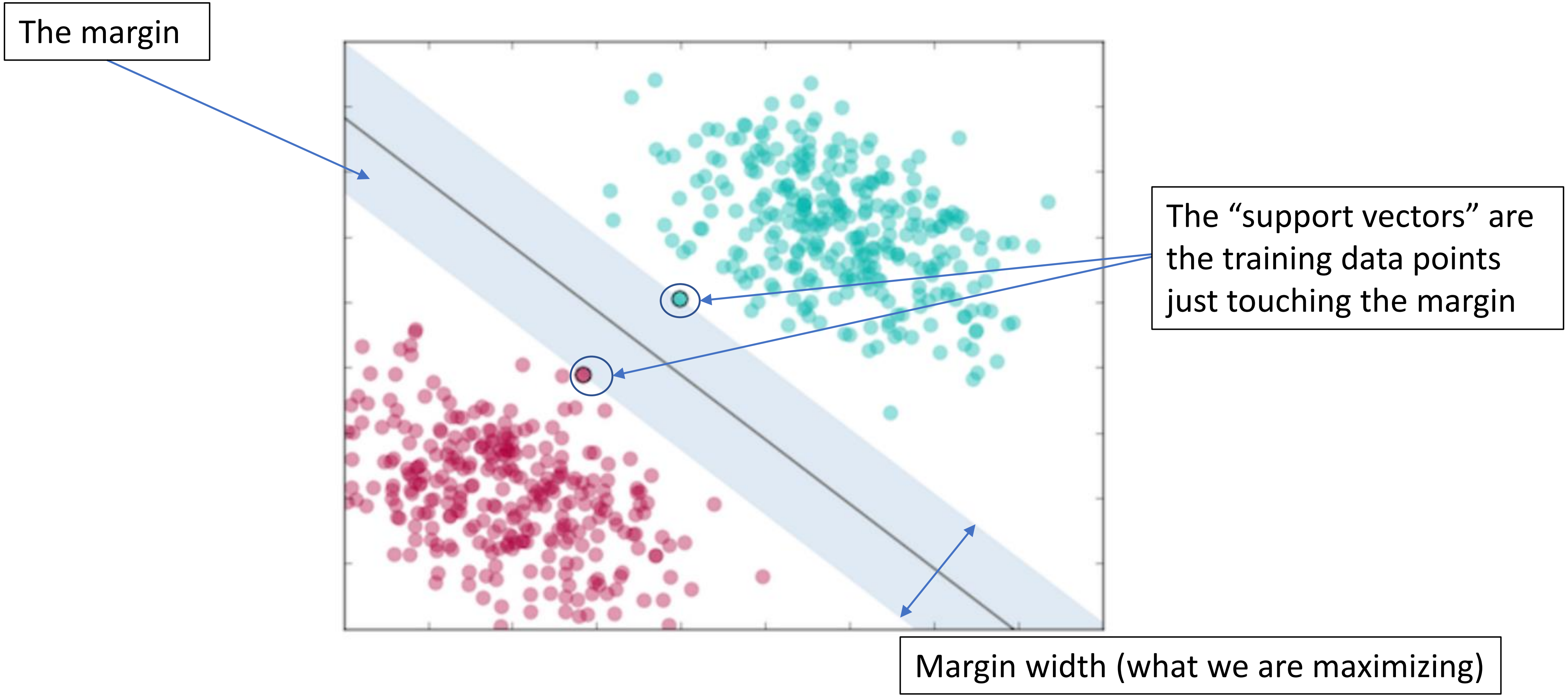
  3. Computational complexity

Support vector machines (SVM)

- What line (or “linear discriminant”) best separates this data?



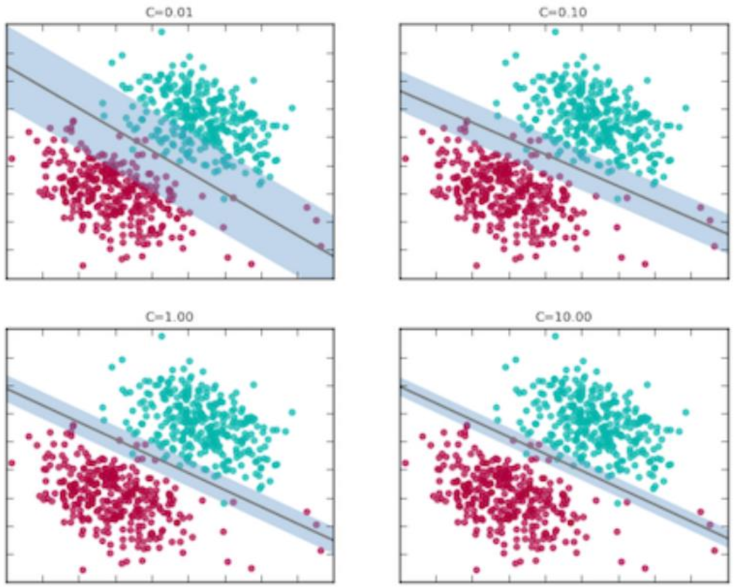
Maximum margin classifier



- This data is linearly separable (“hard margin”)

Data that is not linearly separable ("soft margin")

- There is a natural tradeoff between:
  a) The width of the margin
  b) The number of misclassified points, and how far they are from the margin ("hinge loss")

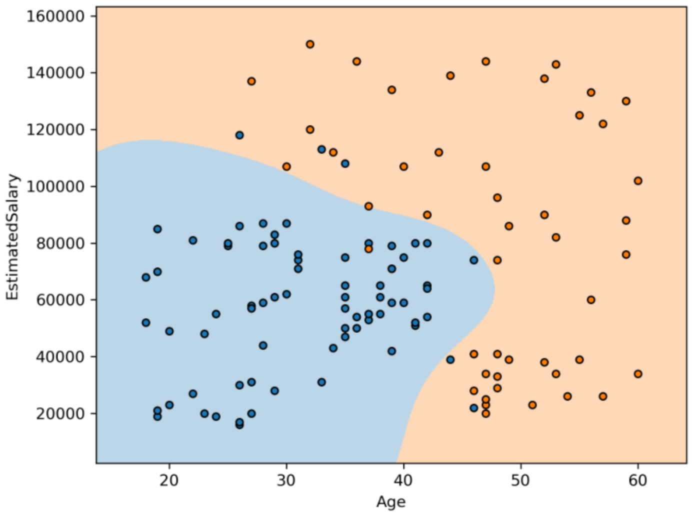


- The parameter C is the weight on component b) above (how do we choose?)
- This is effectively the same as Ridge regression with a different way of measuring errors

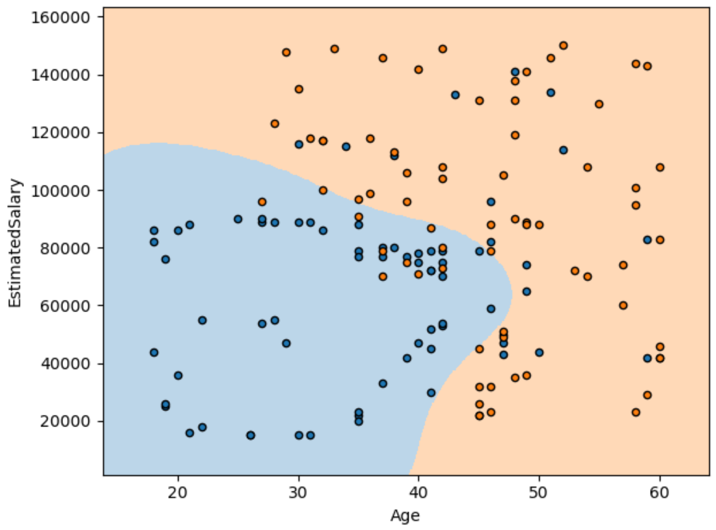
Nonlinear SVM

- We can augment SVM with nonlinear transformations of the training data ("the kernel trick") to obtain nonlinear classifiers
- Many such "kernels" are possible – "radial basis" kernels are widely used and flexible
- In the social ads data, we use radial bases and tune the scale (γ) and the regularization strength (C) using grid search and cross validation

Results on test data (accuracy = 95%)



Support vectors (training data): 141 of them!




```
svc_temp=grid_search_cv.best_estimator_[1]
inds = svc_temp.support_
X_supp = X_train.iloc[inds,:]
y_supp = Y_train.iloc[inds]
plot_decision_boundary(grid_search_cv.best_estimator_, X_supp, y_supp, n_points=500,
                       x_label='Age', y_label='EstimatedSalary')
plt.tight_layout()
```

Example: classifying text as spam or not (“ham”)

- Data from https://archive.ics.uci.edu/ml/datasets/sms+spam+collection
- 5,572 text messages collected by National University of Singapore; 13.4% of these are spam
- Data frame, pre-cleaning:

	Label	SMS
0	ham	Go until jurong point, crazy.. Available only ...
1	ham	Ok lar... Joking wif u oni...
2	spam	Free entry in 2 a wkly comp to win FA Cup fina...
3	ham	U dun say so early hor... U c already then say...
4	ham	Nah I don't think he goes to usf, he lives aro...
5	spam	FreeMsg Hey there darling it's been 3 week's n...
6	ham	Even my brother is not like to speak with me. ...
7	ham	As per your request 'Melle Melle (Oru Minnamin...
8	spam	WINNER!! As a valued network customer you have...
9	spam	Had your mobile 11 months or more? U R entitle...
10	ham	I'm gonna be home soon and i don't want to tal...
11	spam	SIX chances to win CASH! From 100 to 20,000 po...
12	spam	URGENT! You have won a 1 week FREE membership ...
13	ham	I've been searching for the right words to tha...
14	ham	I HAVE A DATE ON SUNDAY WITH WILL!!

- Can we use this data to build a classifier to predict whether a new message is ham or spam?

We can use Bayes' Rule to classify messages

- Probability that a given message is ham:

$$= \frac{P(\text{Ham and Message})}{P(\text{Message})} =$$

$$\mathrm{P}(\mathrm{Ham}|\mathrm{Message}) = \frac{\mathrm{P}(\mathrm{Message}|\mathrm{Ham})\mathrm{P}(\mathrm{Ham})}{\mathrm{P}(\mathrm{Message}|\mathrm{Spam})\mathrm{P}(\mathrm{Spam}) + \mathrm{P}(\mathrm{Message}|\mathrm{Ham})\mathrm{P}(\mathrm{Ham})}$$

- Probability that a given message is spam:

$$\mathrm{P}(\mathrm{Spam}|\mathrm{Message}) = \frac{\mathrm{P}(\mathrm{Message}|\mathrm{Spam})\mathrm{P}(\mathrm{Spam})}{\mathrm{P}(\mathrm{Message}|\mathrm{Spam})\mathrm{P}(\mathrm{Spam}) + \mathrm{P}(\mathrm{Message}|\mathrm{Ham})\mathrm{P}(\mathrm{Ham})}$$

- How can we estimate $\mathrm{P}(\mathrm{Message}|\mathrm{Ham})$ and $\mathrm{P}(\mathrm{Message}|\mathrm{Spam})$?

Let's be naïve!

Probability of including words in message if ham

Probability of excluding all other words if ham

- We assume that:

$$P(\text{Message M} \mid \text{Ham}) = \prod_{\text{i in M}} P(W_i \mid \text{Ham}) \times \prod_{\text{i not in M}} (1 - P(W_i \mid \text{Ham}))$$

and

$$P(\text{Message M} \mid \text{Spam}) = \prod_{\text{i in M}} P(W_i \mid \text{Spam}) \times \prod_{\text{i not in M}} (1 - P(W_i \mid \text{Spam}))$$

- Sometimes called a "bag of words" assumption (why "naïve?")
- *Note:* in the Module 3 video, we discussed Gaussian naïve Bayes: same idea, but the data are continuously distributed, not categorical

Last step: estimating word probabilities in ham & spam

- We make a "vocabulary" of all unique words (7,783 words) in data set then count how many times each word in vocabulary appeared in ham or spam messages

	Label	SMS	quickly	07099833605	lyrics	yrs	unlike	increase	closer	next	...	original	annoying	aiyo	largest	previews	some	finds
0	ham	[yep, by, the, pretty, sculpture]	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0
1	ham	[yes, princess, are, you, going, to, make, me,...	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0
2	ham	[welp, apparently, he, retired]	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0
3	ham	[havent]	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0
4	ham	[i, forgot, 2, ask, ü, all, smth, there, s, a,...	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0

5 rows × 7785 columns

Number of times word appears in spam messages

Number of words across all spam messages

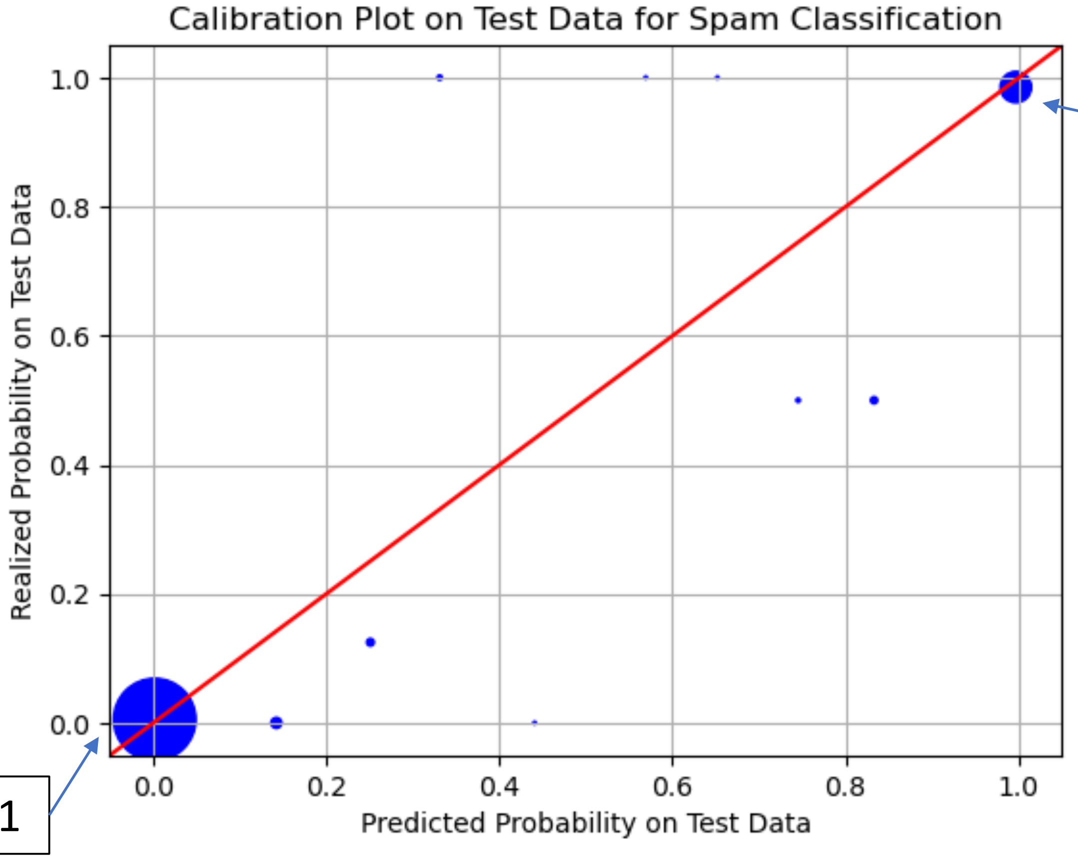
- Then we use:

$$P(w_i|\mathrm{Spam}) = \frac{N_{w_i|\mathrm{Spam}} + \alpha}{N_{\mathrm{Spam}} + \alpha \cdot N_{\mathrm{Vocabulary}}}$$

α is a "smoothing parameter"

$$P(w_i|\mathrm{Ham}) = \frac{N_{w_i|\mathrm{Ham}} + \alpha}{N_{\mathrm{Ham}} + \alpha \cdot N_{\mathrm{Vocabulary}}}$$

Prediction quality



134 of 1114 predictions > 0.99

944 of 1114 predictions < 0.01

Incoming message!

- Should we block or allow this message?

```
test_set['SMS'][917]
```

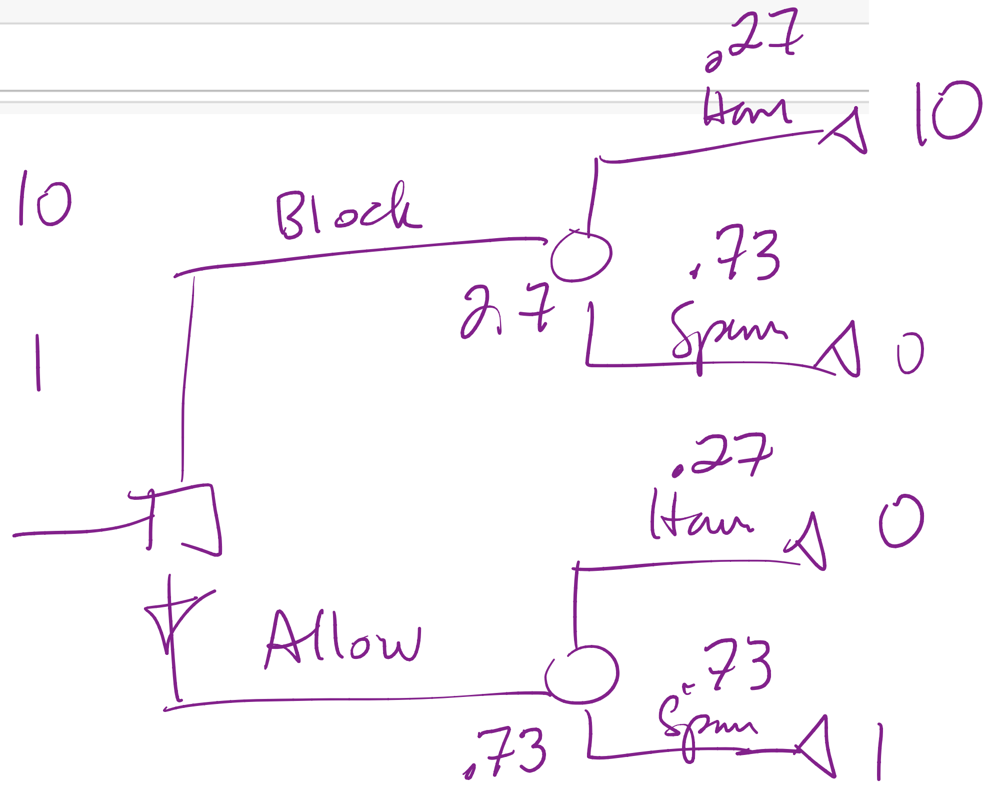
```
'Dont forget you can place as many FREE Requests with 1stchoice.co.uk as you wish. For more Information call 08707808226.'
```

```
classify_message(test_set['SMS'][917])
```

```
['spam', 0.7323405126933885]
```

Cost of Blocking Ham = 10

Cost of Allowing Spam = 1



Looking ahead to next time (Class 4)

- Homework 3 due at 11:59pm on **Monday, February 19**
  - Main goals: practice your understanding of classification methods
  - TA support available over the long weekend

- Class 4:
  - Classification trees
  - Ensemble methods:
    - o Bagging and “random forests”
    - o Boosting