CS102: Big Data

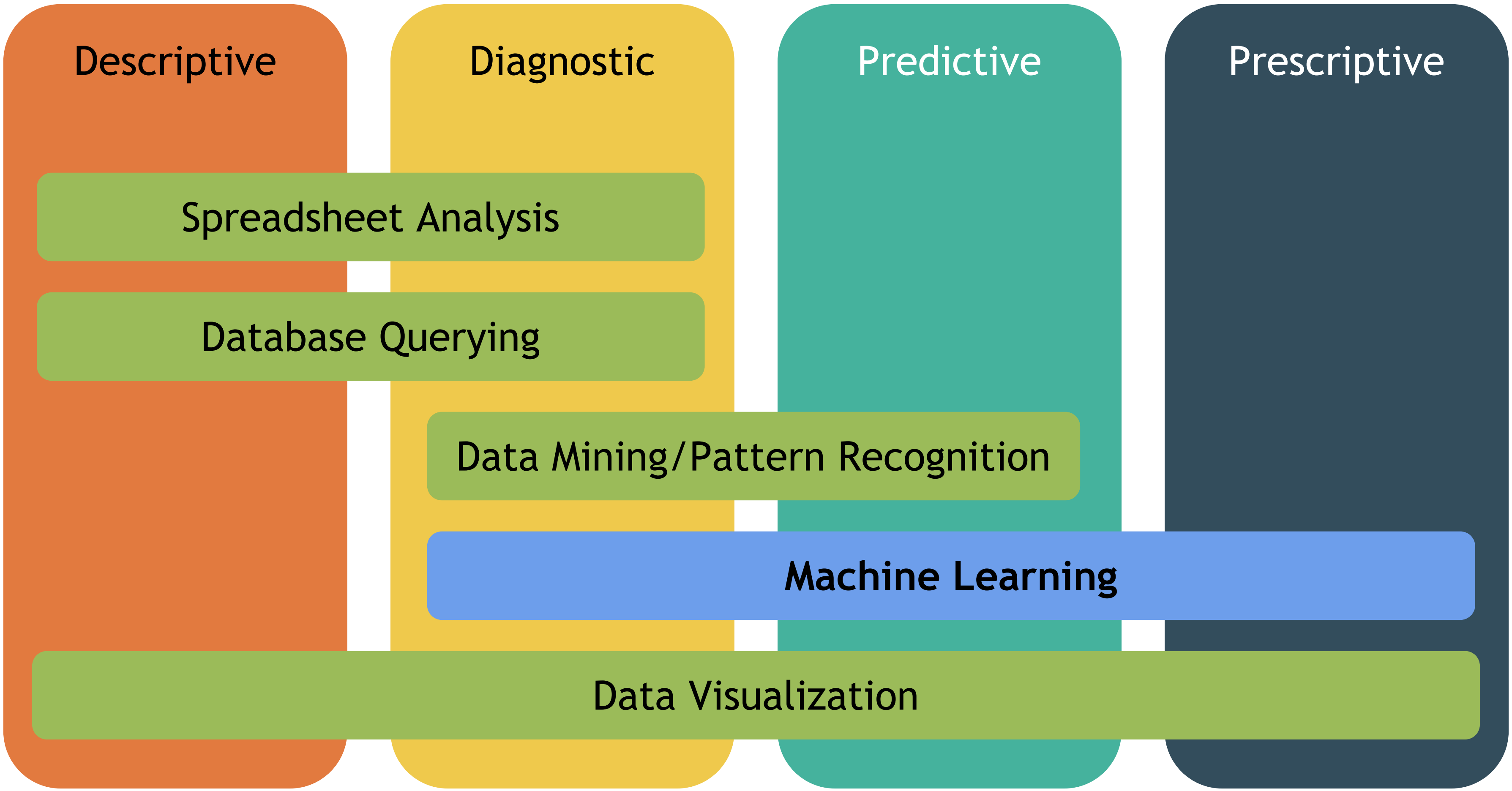
Tools and Techniques, Discoveries and Pitfalls

Spring 2017

Ethan Chan, Lisa Wang

Lecture 10 - Regression Part 2 / Classification Part 1

Tools & Techniques



Announcements

- Midterms graded, pick up after class
- Assignment 4 Part 1 will be out tonight

Last Week

- Introduction to Machine Learning
- ML Application Areas
- Supervised vs. unsupervised learning
- Simple Linear Regression

Plan for Today

- Regression Algorithms
  - Linear Regression Example in Python with Pandas
  - Polynomial Regression
- Classification Algorithms
  - K-Nearest Neighbors
  - Decision Trees
  - Logistic Regression
- Classification Metrics
  - Accuracy

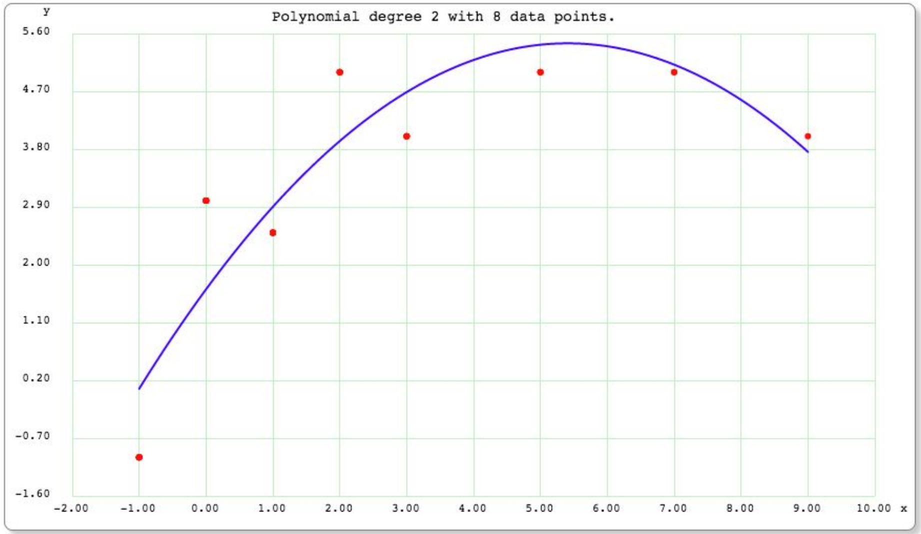
Linear Regression with numpy and pandas

Download lecture_10.zip from the course website

Open lecture_10_regression.ipynb

Polynomial Regression

- Instead of finding the best fit line y=ax + b (degree = 1), find best fit curve (generalize to higher degree polynomials)



Polynomial Regression

More formally:

- Given set of data points (x,y) in two-dimensional space, find n-th degree polynomial $f(x)=a_0+a_1x+a_2x^2+\dots a_nx^n$ that best "fits" the points.
- Degree = 1: line (linear regression)
- Degree = 2: parabola

Interactive Polynomial Regression

https://arachnoid.com/polysolve/

1. Change the degree of the polynomial and observe how the fit curve changes: Try degree = 1, degree =2, degree = 6. What do you observe? Which one do you think yields the best result?

2. Change degrees again, and this time pay attention to the coefficient of determination R^2. What happens to the correlation coefficient when you increase the degree?

3. Add three data points and observe how the best fit changes.

Regression

Using data to make inferences or predictions

- Supervised
- Training data, each example:
  - Set of predictor values - “independent variables”
  - Numerical output value - “dependent variable”
- Model is function from predictors to output
  - Use model to predict output value for new predictor values
- Example
  - Predictors: mother height, father height, current age
  - Output: height

Slide content adopted from Prof. Jennifer Widom’s course materials.

Classification

Using data to make inferences or predictions

- Supervised
- Training data, each example:
  - Set of feature values – numeric or categorical
  - Categorical output value - “label”
- Model is method from feature values to label
  - Use model to predict label for new feature values
- Example
  - Feature values: age, gender, income, profession
  - Label: buyer, non-buyer

Slide content adopted from Prof. Jennifer Widom’s course materials.

Classification: More Examples

Medical diagnosis

- **Feature values:** age, gender, history, symptom1-severity, symptom2-severity, test-result1, test-result2
- **Label:** disease

Email spam detection

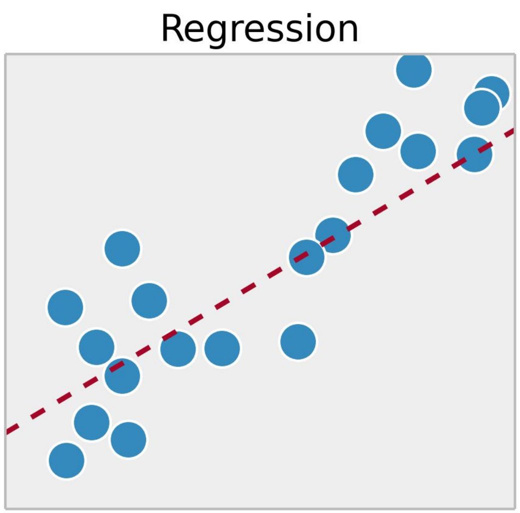
- **Feature values:** sender-domain, length, #images, $keyword_1$, $keyword_2$, …, $keyword_n$
- **Label:** spam or not-spam

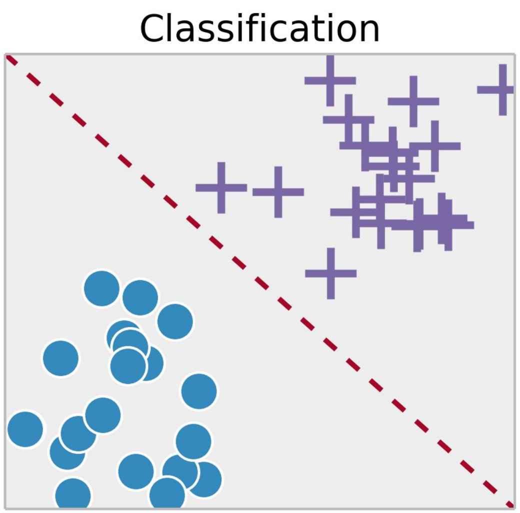
Credit card fraud detection

- **Feature values:** user, location, item, price
- **Label:** fraud or okay

Slide content adopted from Prof. Jennifer Widom's course materials.

Regression vs. Classification





Output values are real numbers (continuous)

Regression fits a curve to the data, so you can use the curve to predict the real-valued output

Output values in two or more classes, e.g. cats and dogs (discrete)

Classification tries to predict the class based on features by learning *decision boundaries (the red line)*

K-Nearest Neighbors

Predict the “majority vote” of your neighbors

K-Nearest Neighbors (KNN)

For any pair of data items i_1 and i_2, from their feature values compute $distance(i_1, i_2)$

Example:

Features - gender, profession, age, income, postal-code

$person_1$ = (male, teacher, 47, $25K, 94305)
$person_2$ = (female, teacher, 43, $28K, 94309)

$distance(person_1, person_2)$

Intuitively, distance should measure similarity between two data items

Slide content adopted from Prof. Jennifer Widom's course materials.

K-Nearest Neighbors (KNN)

Features - gender, profession, age, income, postal-code

$person_1$ = (male, teacher, 47, $25K, 94305) buyer

$person_2$ = (female, teacher, 43, $28K, 94309) non-buyer

Remember training data has labels

To classify a new item *i* : In the labeled data find the K closest items to *i*, assign most frequent label

$person_3$ = (female, doctor, 40, $40K, 95123)

Slide content adopted from Prof. Jennifer Widom's course materials.

KNN Example: Predicting City Temperatures

- City temperatures – France and Germany
- Features: longitude, latitude
- Distance is Euclidean distance

  $distance([o_1,a_1],[o_2,a_2]) = sqrt((o_1-o_2)^2 + (a_1-a_2)^2)$

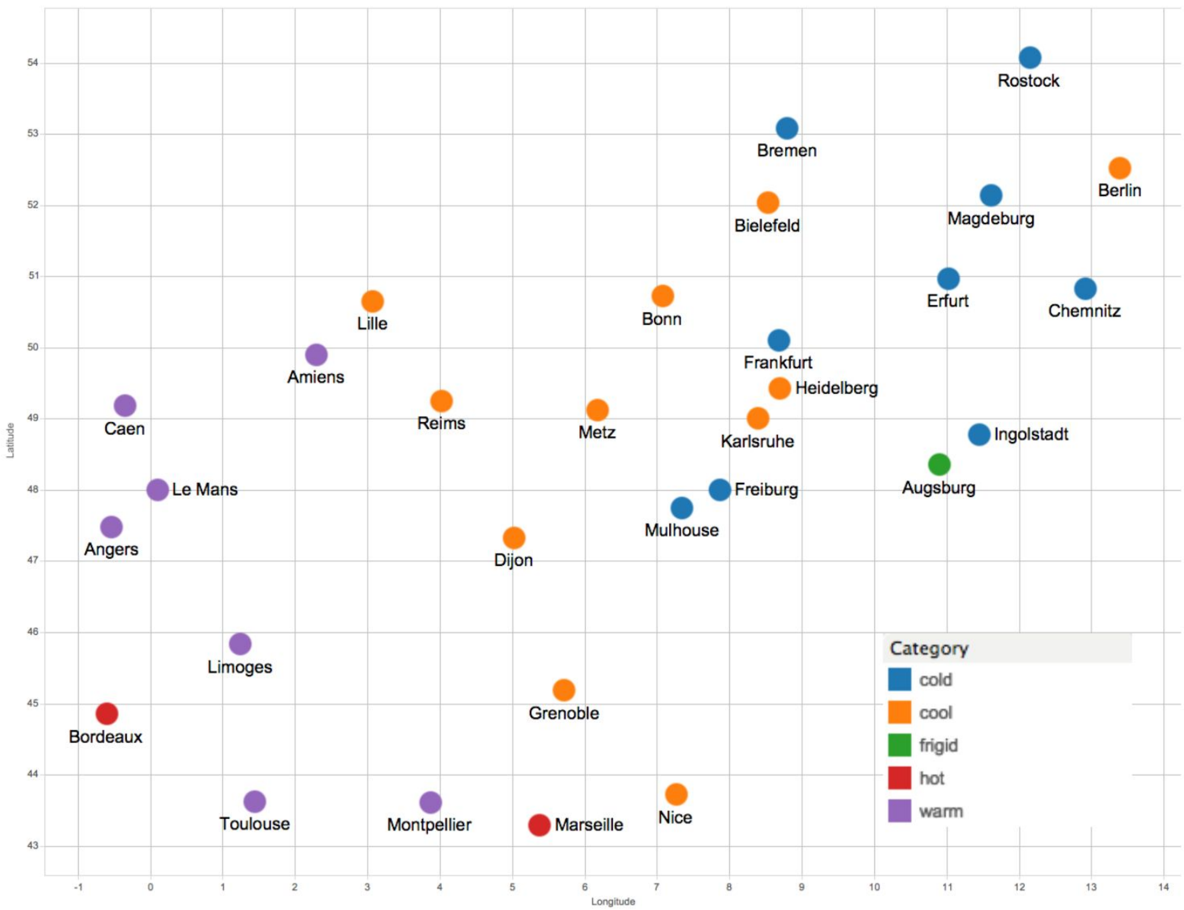
  = actual distance in x-y plane
- Labels: frigid, cold, cool, warm, hot

Nice (7.27, 43.72) cool
Toulouse (1.45, 43.62) warm
Frankfurt (8.68, 50.1) cold
......

Predict temperature category from longitude and latitude

Slide content adopted from Prof. Jennifer Widom's course materials.

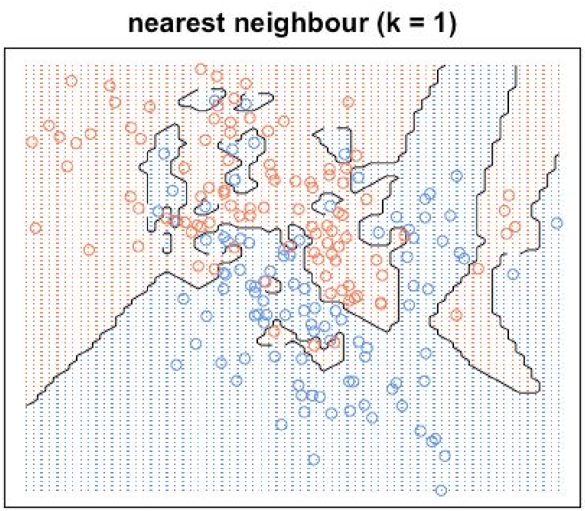
KNN Example: Predicting City Temperatures

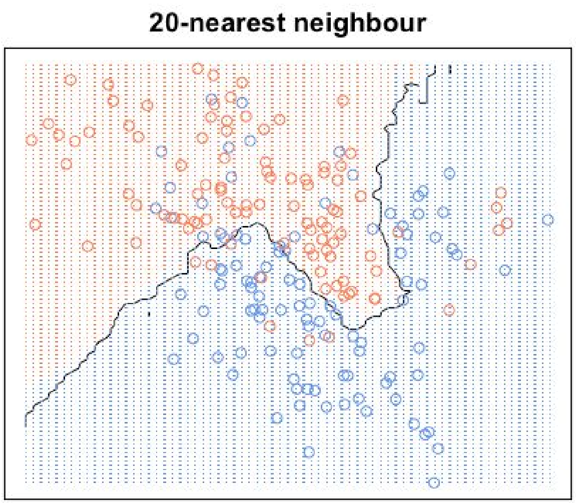


Slide content adopted from Prof. Jennifer Widom's course materials.

KNN : What's the K?

- k is the number of nearest neighbors (data items) we take into account to make our prediction
- Odd k preferred for breaking ties, but not necessary
- Choose k to balance overfitting (k too small) / underfitting (k too large)





Source: https://kevinzakka.github.io/2016/07/13/k-nearest-neighbor/

KNN Summary

To classify a new item *i* : find K closest items to *i* in the labeled data, assign most frequent label

Pros:

- Simple and intuitive algorithm, no hidden math
- Training data itself forms the model, so "training" is instantaneous
- Once distance function is defined, rest is easy

Slide content adopted from Prof. Jennifer Widom's course materials.

KNN Summary

To classify a new item *i*: find K closest items to *i* in the labeled data, assign most frequent label

Cons:

- Finding nearest neighbors in high dimensions is computationally hard
- Not efficient for data with lots of features
  - Medical Diagnosis: Symptoms as features,
  - Email spam detection: Words as features
- Does not perform well if classes are imbalanced

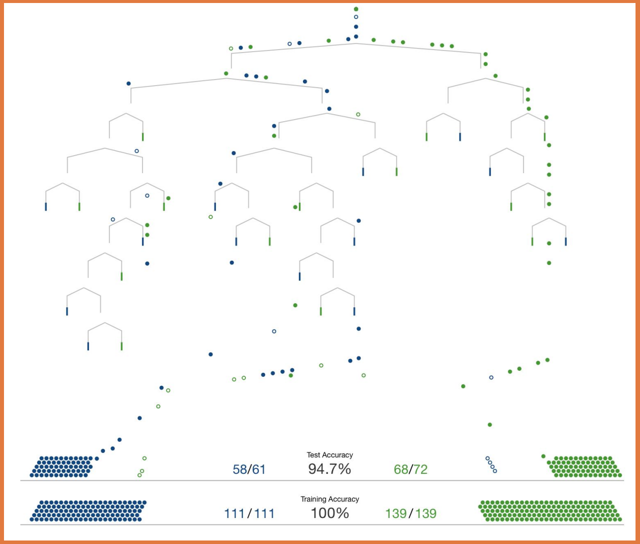
Slide content adopted from Prof. Jennifer Widom's course materials.

Decision Trees

Identifying boundaries, one branch at a time

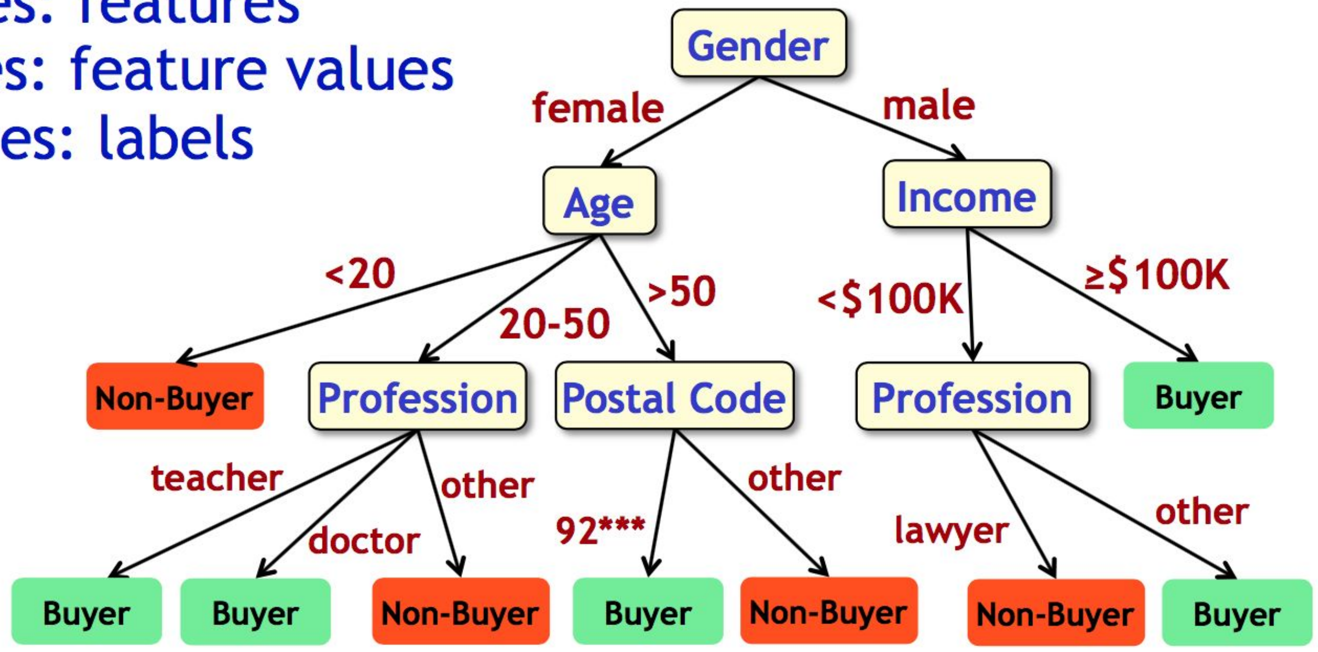
A visual introduction to Machine learning (Decision Trees)

http://www.r2d3.us/visual-intro-to-machine-learning-part-1/



Decision Trees

Nodes: features
Edges: feature values
Leaves: labels



New data item to classify:
Navigate tree based on feature values

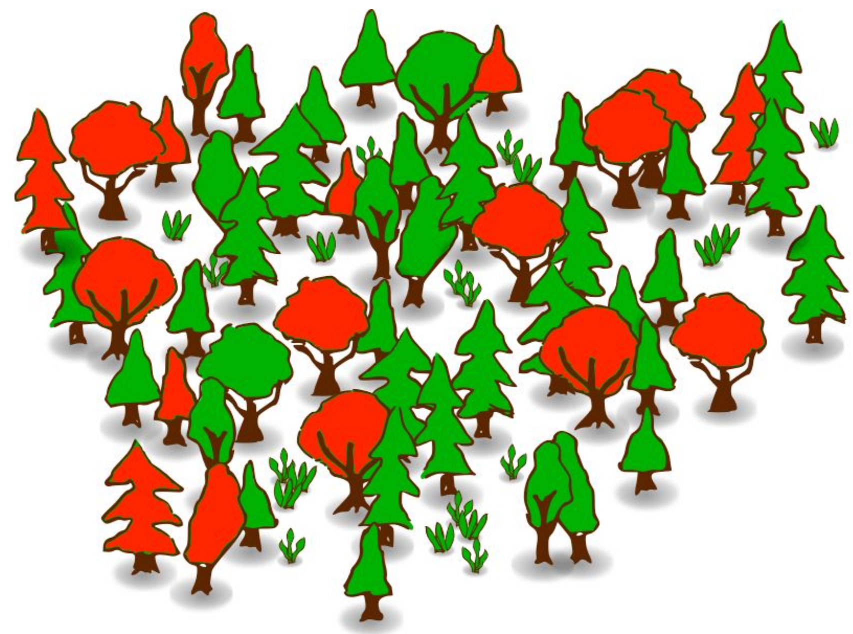
Slide content adopted from Prof. Jennifer Widom's course materials.

Decision Trees: Challenges

- Primary challenge is building good decision trees from training data
  - Which features and feature values to use at each choice point
  - HUGE number of possible trees even with small number of features and values
- Common approach: Create a “forest” of many decision trees, and combine results

Random Forest

- A random forest is a group ("ensemble") of decision trees
- To make a prediction, we first predict using each decision tree, and then choose the class with the "tree votes"
- Generalizes better than single decision tree

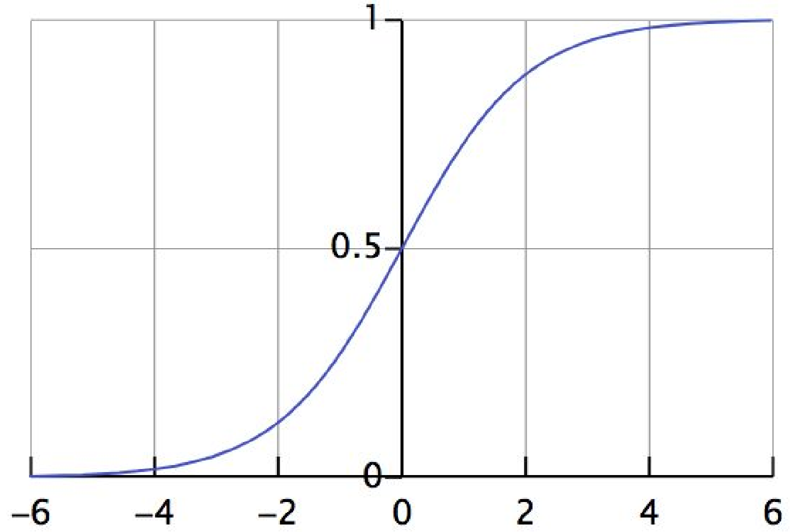


Source:
http://www.kdnuggets.com/2016/12/random-forests-python.html

Logistic Regression

Predicting the probability of class

What's a Logistic Function?



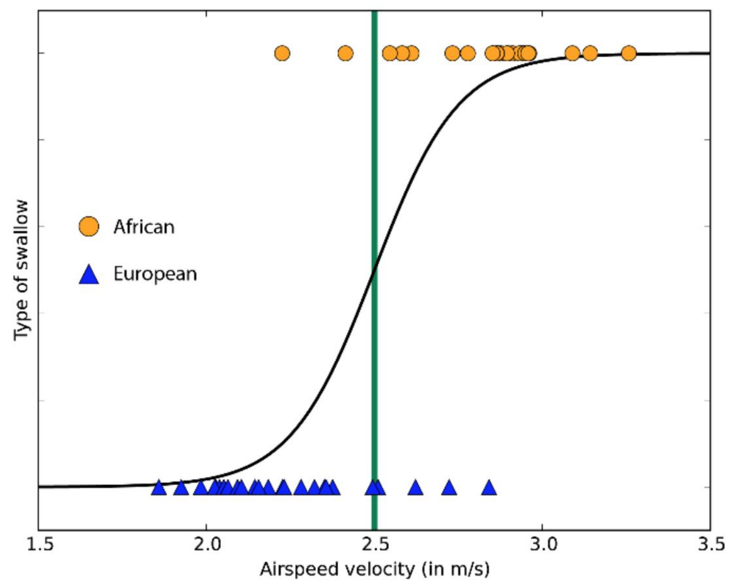
Standard logistic function: L = 1, k = 1, x_0 = 0
Source: https://en.wikipedia.org/wiki/Logistic_function

$$f(x) = \frac{L}{1 + \mathrm{e}^{-k(x - x_0)}}$$

- e = natural logarithm base (Euler's number)
- x_0 = x-value of sigmoid's midpoint
- L = curve's maximal value
- k = steepness of the curve

Logistic Regression Example

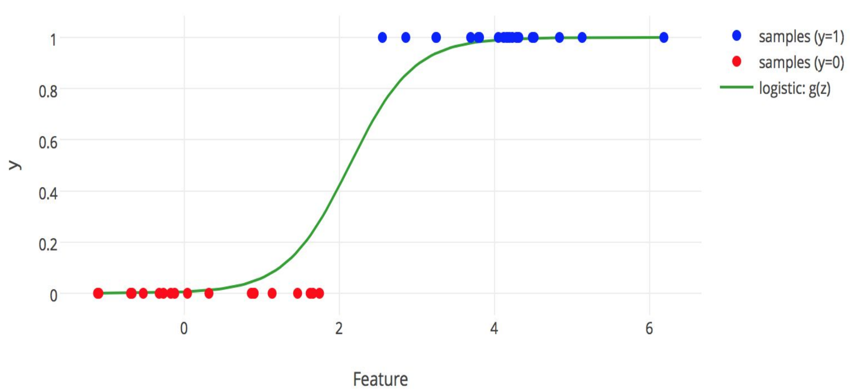
- Input feature x: Airspeed velocity (single feature)
- Output class y: Type of Swallow (two classes)
- Decision boundary: find x where logistic fit g(x) = threshold T, usually T = 0.5. (Here at x = 2.5)



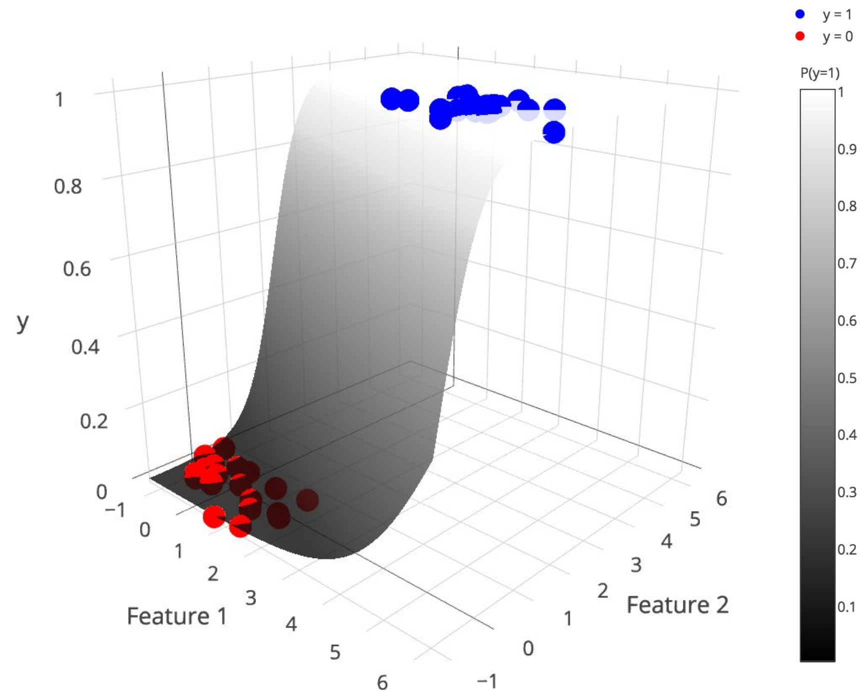
Slide content adopted from Prof. Jennifer Widom's course materials.

Generalizing to More Features

1 Feature



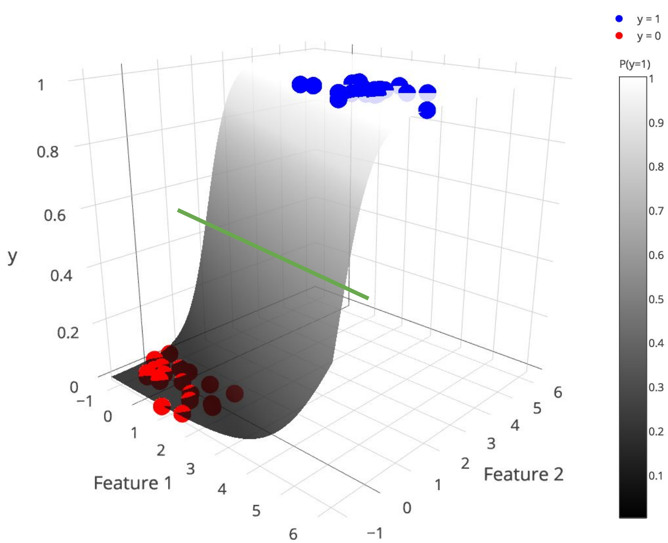
2 Features

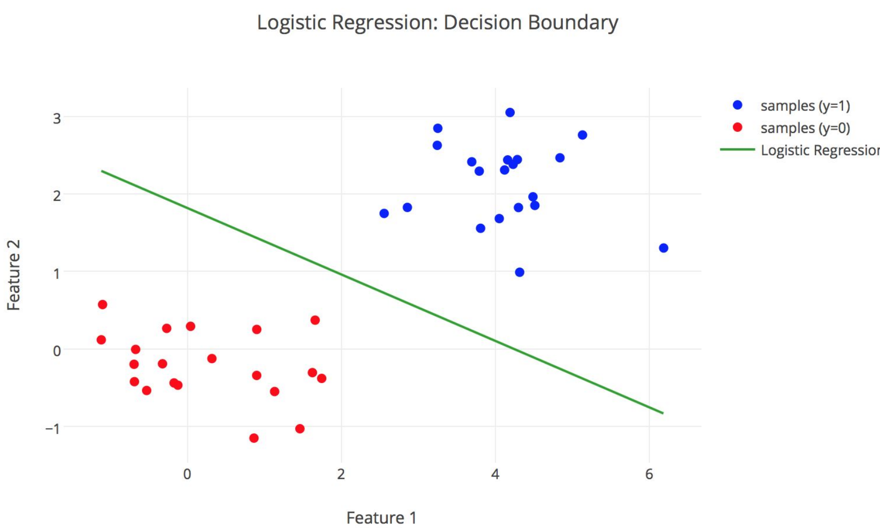


Source: https://florianhartl.com/logistic-regression-geometric-intuition.html

Logistic Regression: Decision Boundary

From logistic function to decision boundary





Why is Logistic Regression a Classification algorithm?

- Regression has multiple meanings :/
- The term regression in logistic regression refers to the underlying mathematical technique to find the best fit function
- Regression vs. classification in ML refers to a specific task (predicting real-valued outputs vs. discrete classes)

Training and Test

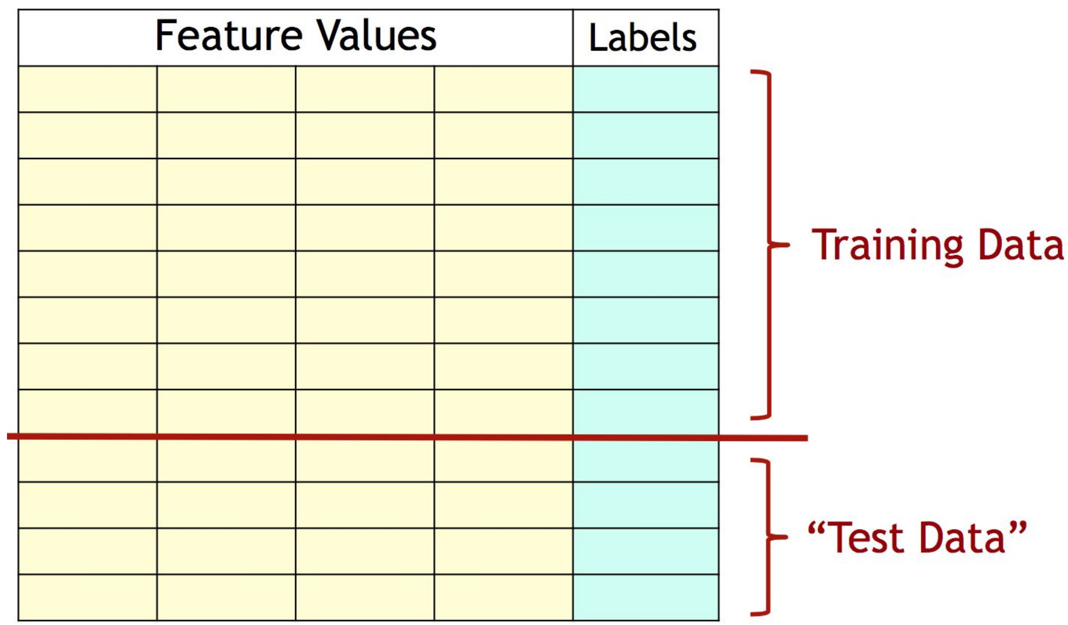
Evaluating our Models

Training and Test

- Create ML model from training data
- How do we know whether it is a good model?
- How can we figure out whether it underfit/overfit?

Training and Test

- Solution: Hold out some of your known data as test data



Slide content adopted from Prof. Jennifer Widom’s course materials.

Training and Test

- Solution: Hold out some of your known data as test data
- Evaluate your model (e.g .prediction accuracy) on both the training data and the test data
- If training accuracy much higher than test accuracy, then model likely overfitted

Let's code it up in Python!

Open lecture_10_classification_part_1_starter.ipynb

Up Next

- Classification Part 2
  - Naive Bayes
  - Support Vector Machines (SVM)
- Unsupervised Learning - Clustering
  - K Means Algorithm
- More metrics
  - False positives, false negatives
  - F1 Score

More Readings / Resources

- Complete Guide to KNN in R and Python by Kevin Zakka: https://kevinzakka.github.io/2016/07/13/k-nearest-neighbor/
- Logistic Regression - Geometric Intuition by Florian Hartl: https://florianhartl.com/logistic-regression-geometric-intuition.html