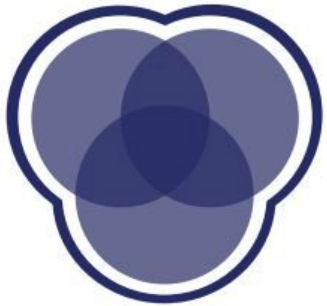


CDS

Cornell Data Science

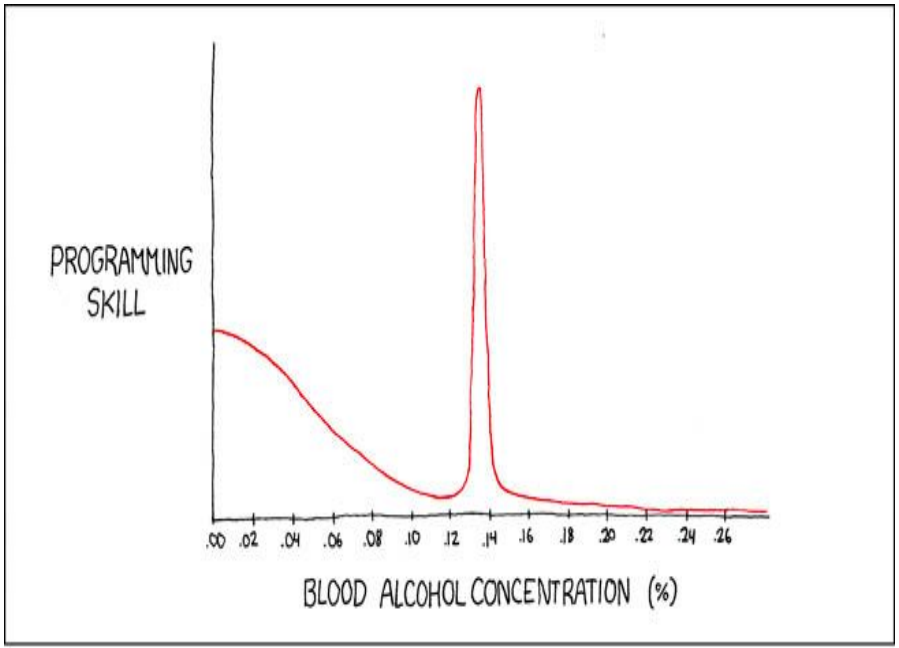
Classification

Recap

We have learned regression models to predict numeric continuous variables.

- Linear Regression
- Logistic Regression
- Decision Tree

Ex: Predicting stock value, monthly temperature, etc.



https://xkcd.com/323/

Intro to Classification

“What kind of species is this?”

“How would consumers rate this restaurant?”

“Which Hogwarts House do I belong to?”

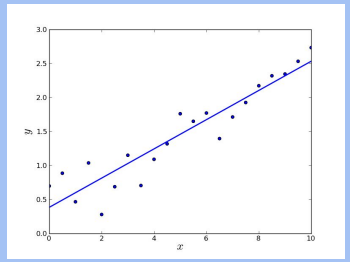
“Am I going to pass this class?”

https://www.bdcwire.com/wp-content/uploads/2015/04/Screen-Shot-2015-04-06-at-1.40.38-PM-e1428342443848-1000x531.png

Recall: Predictive Modeling Challenges

Regression

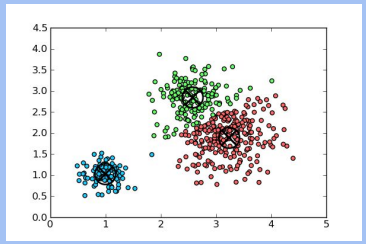
"How much?"
Used for *quantitative* predictions



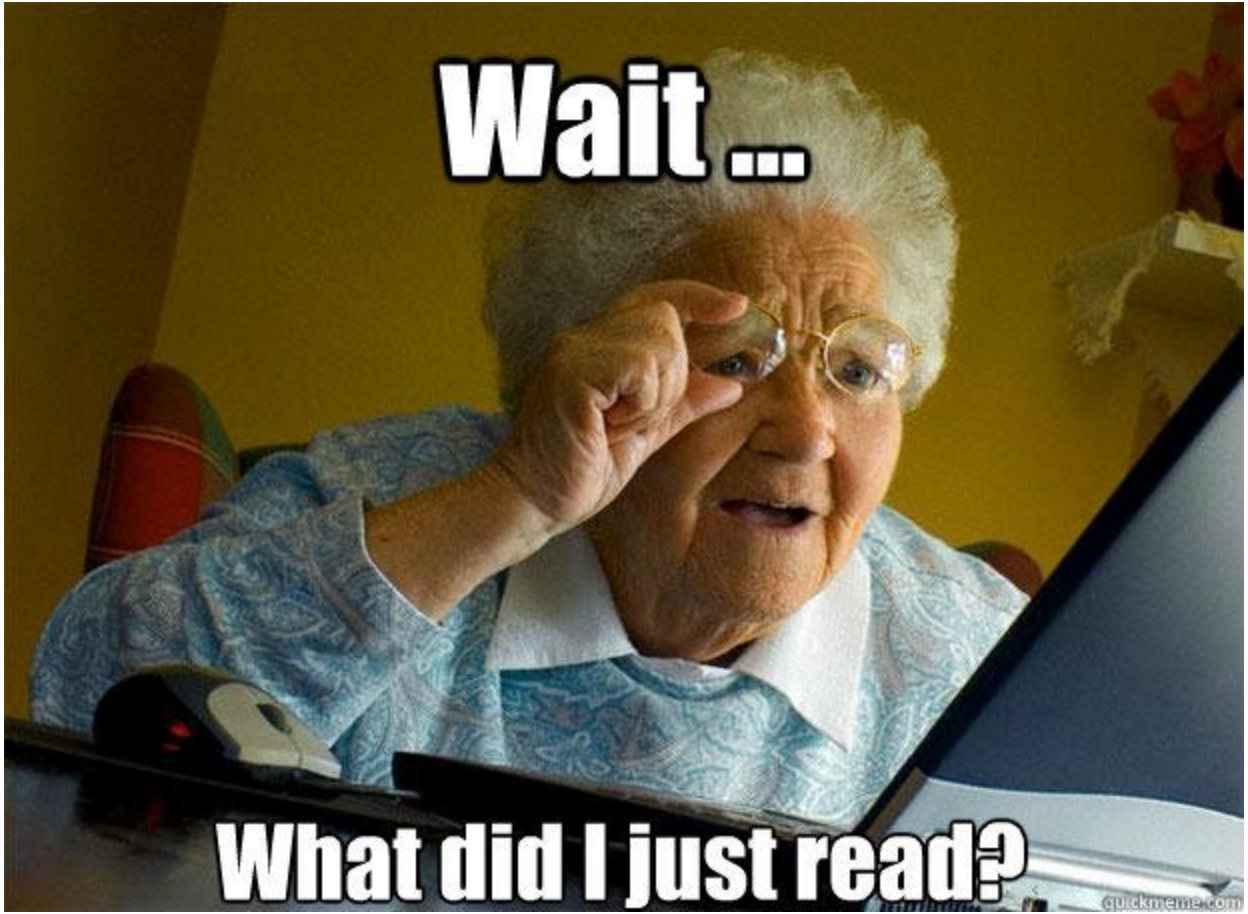
https://lh3.ggpht.com/-d16xbB_2m94/Uml0Gi76XtI/AAAAAAAAI94/5ezC-ZqZWz8/s1600/linearRegression.png

Classification

"What kind?"
Used for *qualitative* predictions



http://blog.mpacula.com/wp-content/uploads/2011/04/kmeans1.png


Wait ...
What did I just read?
quickmeme.com

Classifier 1: k-Nearest Neighbors (KNN)

Simple yet powerful tool

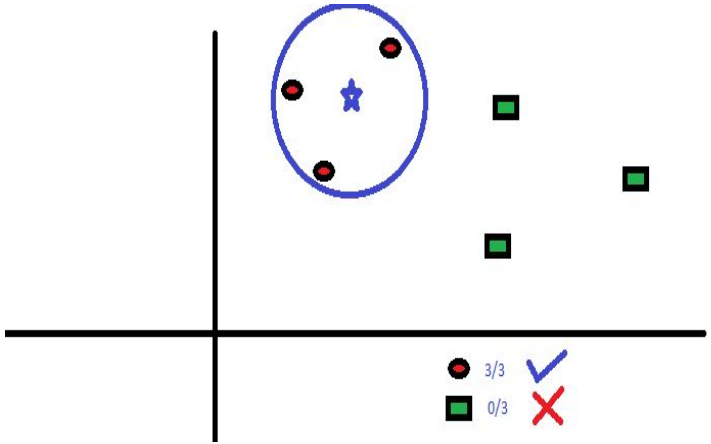
- Very easy to interpret output
- Relatively fast calculation time
- Makes no prior assumptions

Too simple and only good for coarse analysis

KNN

How does it work?

- *Define* k value (in this case k = 3)
- *Pick* a point to predict (blue star)
- *Count* the number of closest types
- *Increase* the radius until the nearest type adds up to 3
- *Predict* the blue star to be a red circle!



https://www.analyticsvidhya.com/blog/2014/10/introduction-k-neighbours-algorithm-clustering/

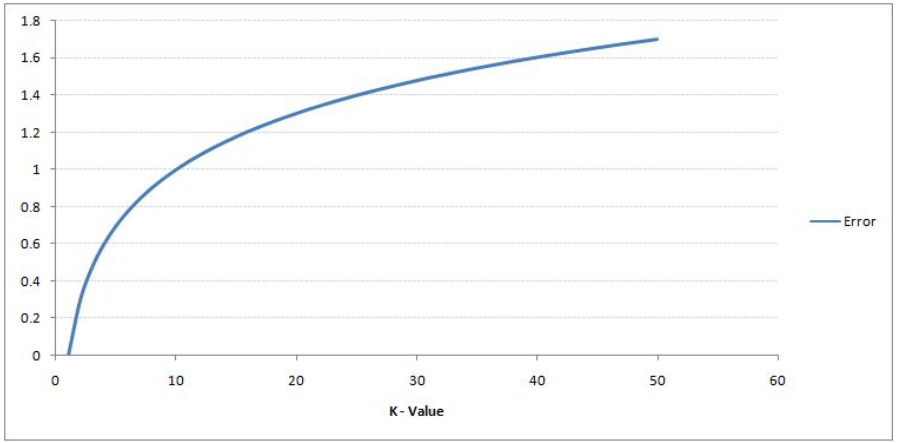
Question:

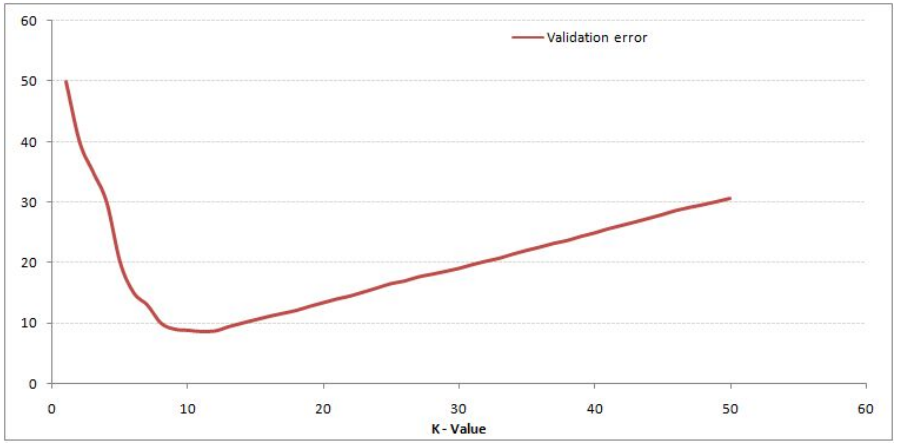
What defines a good *k* value?

Why is *k* odd?

KNN

k-value has a relationship to fit!





https://www.analyticsvidhya.com/blog/2014/10/introduction-k-neighbours-algorithm-clustering/

KNN in R - Demo Time!

We'll use the `knn` function from the `class` library.

Classifier 2: Naive Bayes

- Use Bayes Theorem
- Easy model to understand
  - Very fast!
  - Robust to outliers
  - Strong prediction even with small datasets
- Assuming every feature is **independent**

Conditional Probability and Independence

Conditional Probability:

Probability of an event A *given* an event B

Still confusing? Draw diagrams!

Independence:

When 2 events do not affect the probability of the other event occurring.
Assuming independence simplifies the calculation

Bayes' Theorem

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

https://www.quora.com/How-is-Bayes-theorem-used-in-robotics

Naive Bayes Classifier

Problem: We have k classes and want to predict which class the point x belongs to.

- Represent x with a vector of n features and calculate the probability of x being in each k given the the vector.
- Predict x to be the class with the maximum probability

By using Bayes' Theorem and assuming all n features are independent, this probability can be expressed as

$$p(C_k) \prod_{i=1}^{n} p(x_i \mid C_k)$$

http://scikit-learn.org/stable/modules/naive_bayes.html

Naive Bayes (Continued)

Naive Bayes classifiers differ by how they assume the distribution of $P(x_i|C_k)$.

Gaussian Naive Bayes: The likelihood of features is assumed to be Gaussian.

Bernoulli Naive Bayes: The likelihood of features is assumed to follow Bernoulli distribution

Naive Bayes in R - Demo Time!

Classifier 3 - SVM

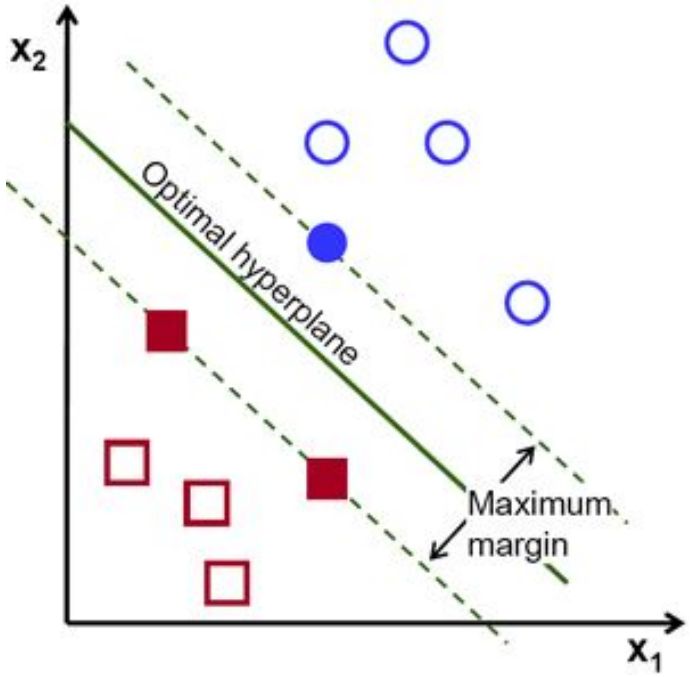
- Support Vector Machine
- Powerful tool with a badass name
  - Great at classifying data in a high dimensional space
  - Only uses subset of data, hence memory efficient
- Requires large calculation time and not great at noisy datasets

Maximal Margin Classifier

We want to find a **separating hyperplane**.

Once we find some candidates for the hyperplane, we try to maximize the **margin**, the distance from borderline points.

SVM is a maximal margin classifier on a higher dimension using **kernels**.



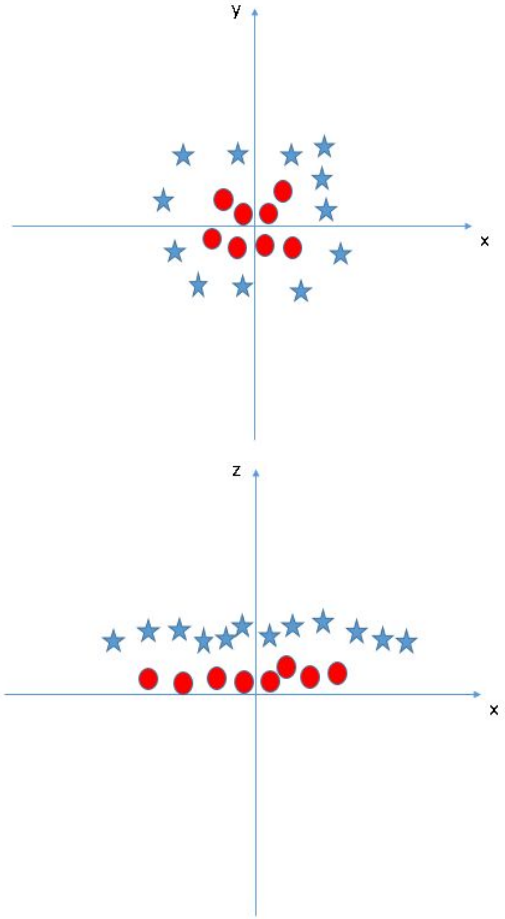
http://docs.opencv.org/2.4/doc/tutorials/ml/introduction_to_svm/introduction_to_svm.html

Kernels in Action

You cannot linearly divide the 2 classes on the *xy* plane at right.

Introduce new feature, $z = x^2 + y^2$ (**radial kernel**)

Map 2 dimensional data onto 3 dimensional data. Now a hyperplane is easy to find. (Imagine slicing a cone!)



https://www.analyticsvidhya.com/blog/2015/10/understaing-support-vector-machine-example-code/

Demo Time!

We'll use the `ksvm` function from `kernlab` library.

Coming Up

Your problem set: Continue project 1

Next week: Clustering and unsupervised learning!