CDS

Cornell Data Science

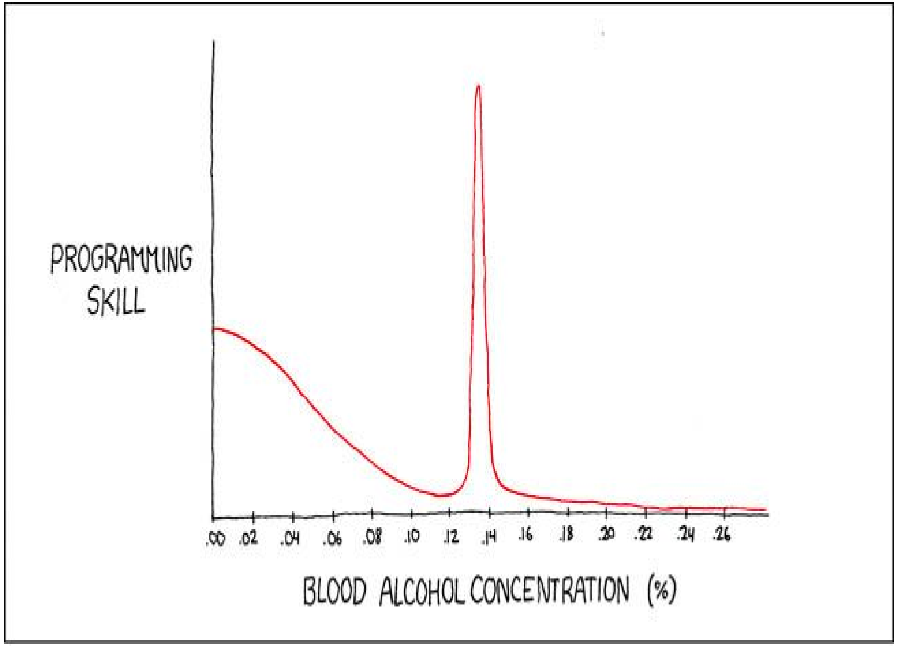
Classification

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?”

Source

Conditional Probability

Conditional Probability - Probability of an event A *given* an event B. Normalize the probability of A and B by the probability of A. Written $P(A|B)$.

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

Independence - Completely unrelated (very different from **uncorrelated**)

In terms of conditional probability, A and B are independent iff $P(A|B) = P(A)$.

Source

The Bayesian Classifier

- The ideal classifier: a theoretical classifier with the highest accuracy
- Picks the class with the highest conditional probability for each point
- Assumes conditional distribution is known
- Exists only in theory and does not exist in reality!
- A conceptual **Golden Standard**

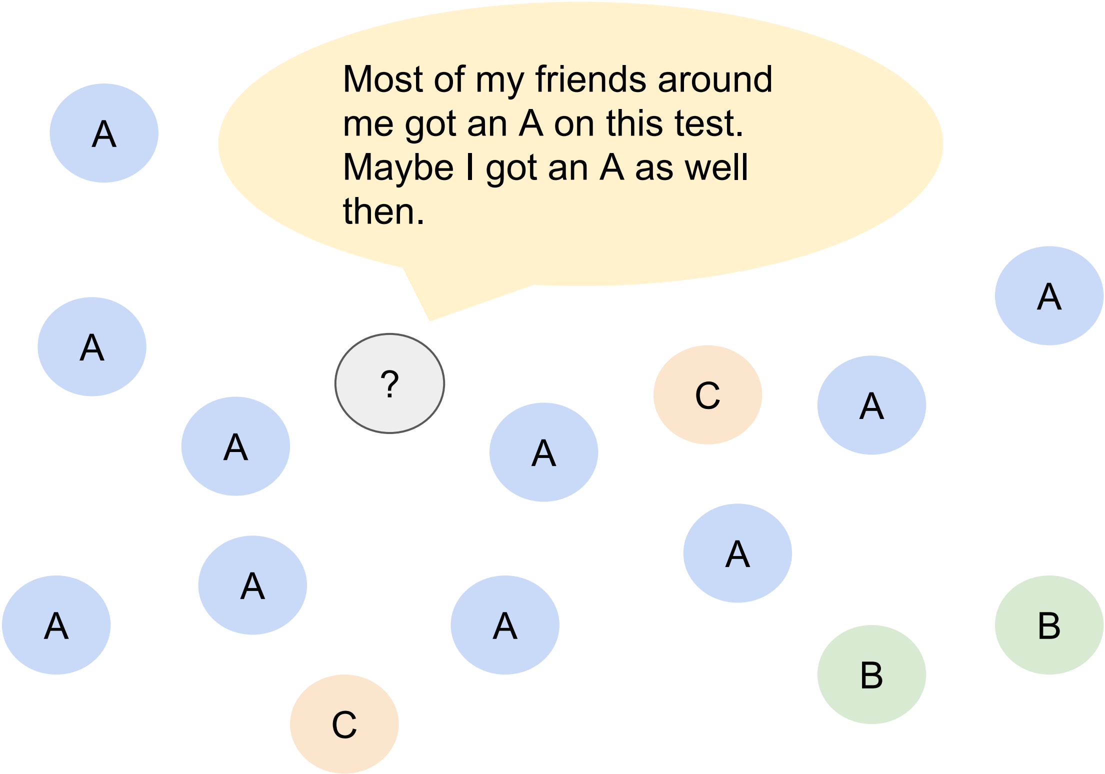
Classifier 1: k-Nearest Neighbors (KNN)

Easy to Interpret

Fast calculation

No prior assumptions

Good for coarse analysis



KNN

How does it work?

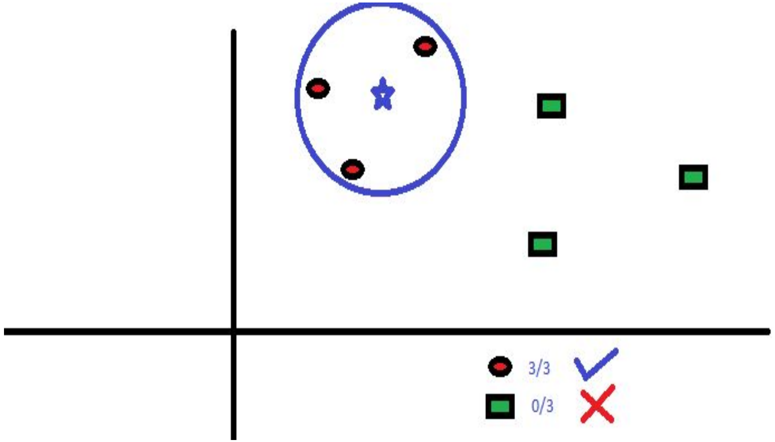
Define a *k* value (in this case k = 3)

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!



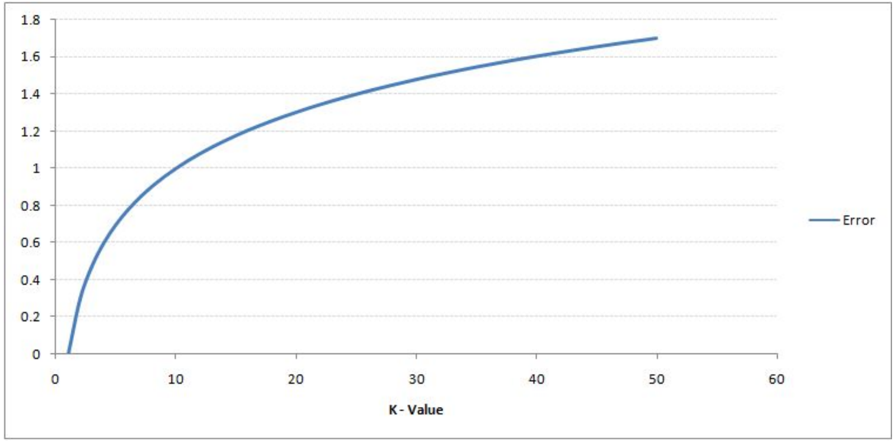
https://www.analyticsvidhya.com/blog/2014/10/introduction-k-neighbours-algorithm-clustering/

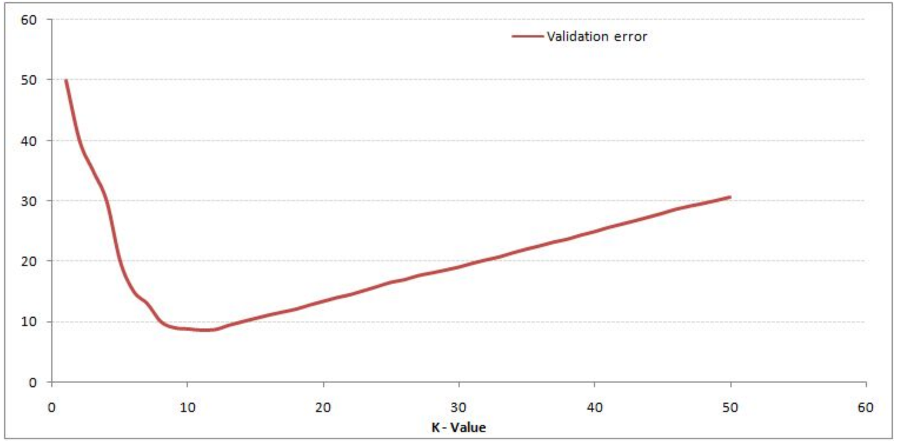
Question:

What defines a good *k* value?

KNN

The *k* value you use has a relationship to the fit of the model.





Source

Classifier 2: Naive Bayes Classifier

Problem. We have k classes and want to predict which class the point x belongs to. x is a vector with n features.

- Calculate the probability of x being in each k
- Predict x to be the class with the maximum probability

Assuming all n features are independent, the probability of x being in each k is

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

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

Probability Distribution Used

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

Gaussian Naive Bayes: likelihood of features assumed to be normally distributed

Bernoulli Naive Bayes: The features follow a "coin-flip" model. Two outcomes, one with probability p and one with probability $1 - p$.

Other lesser-known distributions include Beta and Gamma.

Classifier 3: Support Vector Machine

Powerful tool with a cool name.

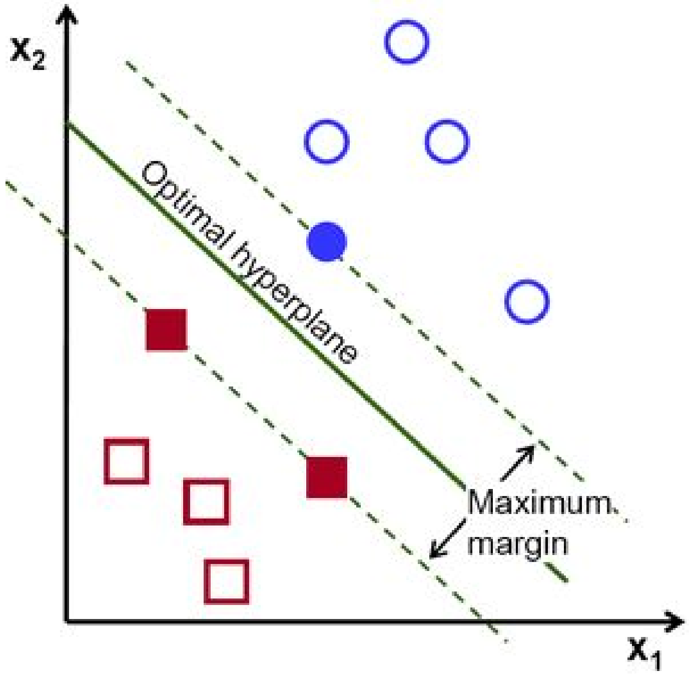
Great at classifying data in high dimensional spaces. Only uses subset of data, hence memory efficient.

Note: requires large calculation time and doesn’t handle noise well.

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 normal distance from borderline points.

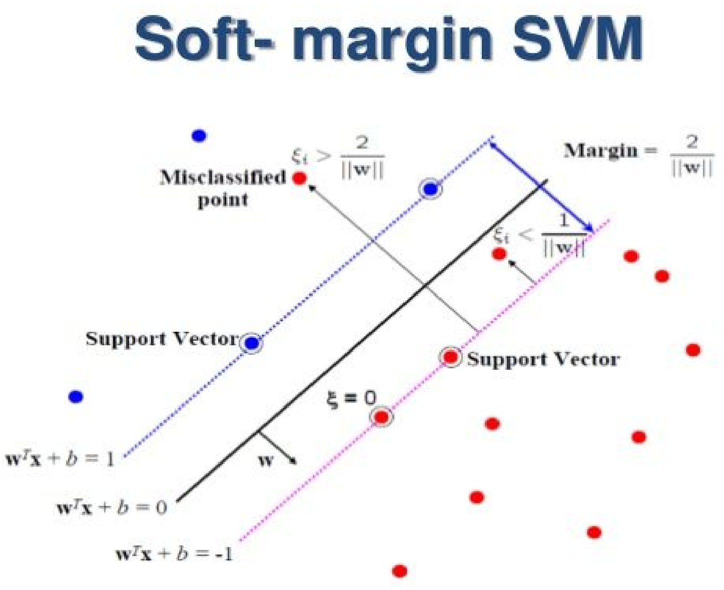


Source

Hard/Soft Margins

What if the two regions are not **linearly separable**?

- Soft margin allows misclassification
- Can account for "dirty" boundaries



$y_i(w^T x_i + b) \geq 1 - \xi_i \;\; for \;\; i = 1, \dots\dots, M \;\dots\dots(7)$

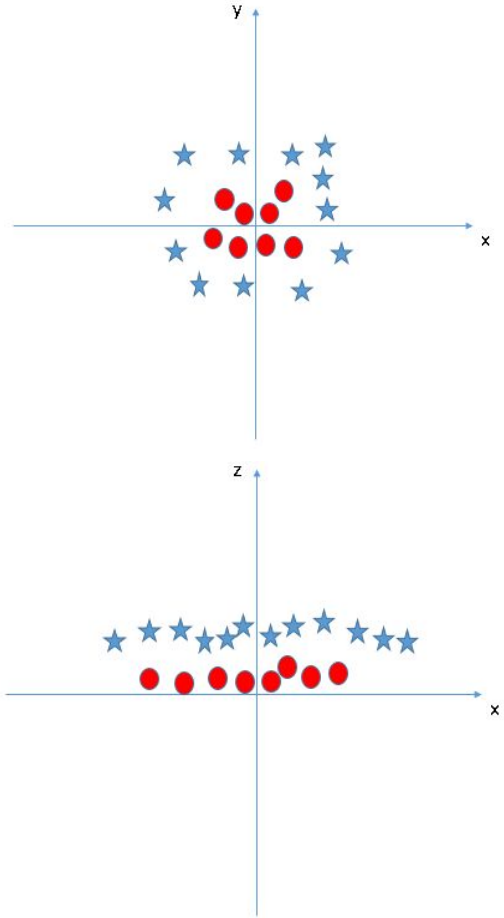
https://www.google.com/url?sa=i&rct=j&q=&esrc=s&source=images&cd=&cad=rja&uact=8&ved=0ahUKEwjHi8n3uPzSAhWr5YMKHeSVClgQjRwIBw&url=https%3A%2F%2Fwww.slideshare.net%2Fkambliruta%2Fevent-classification-prediction-using-support-vector-machine&psig=AFQjCNEs1KIzzSUG_2oEF31WYMTuPJUh4Q&ust=1490901995665839

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!)



Source

Coming Up

Your problem set: Continue project 1

Next week: Clustering and unsupervised learning