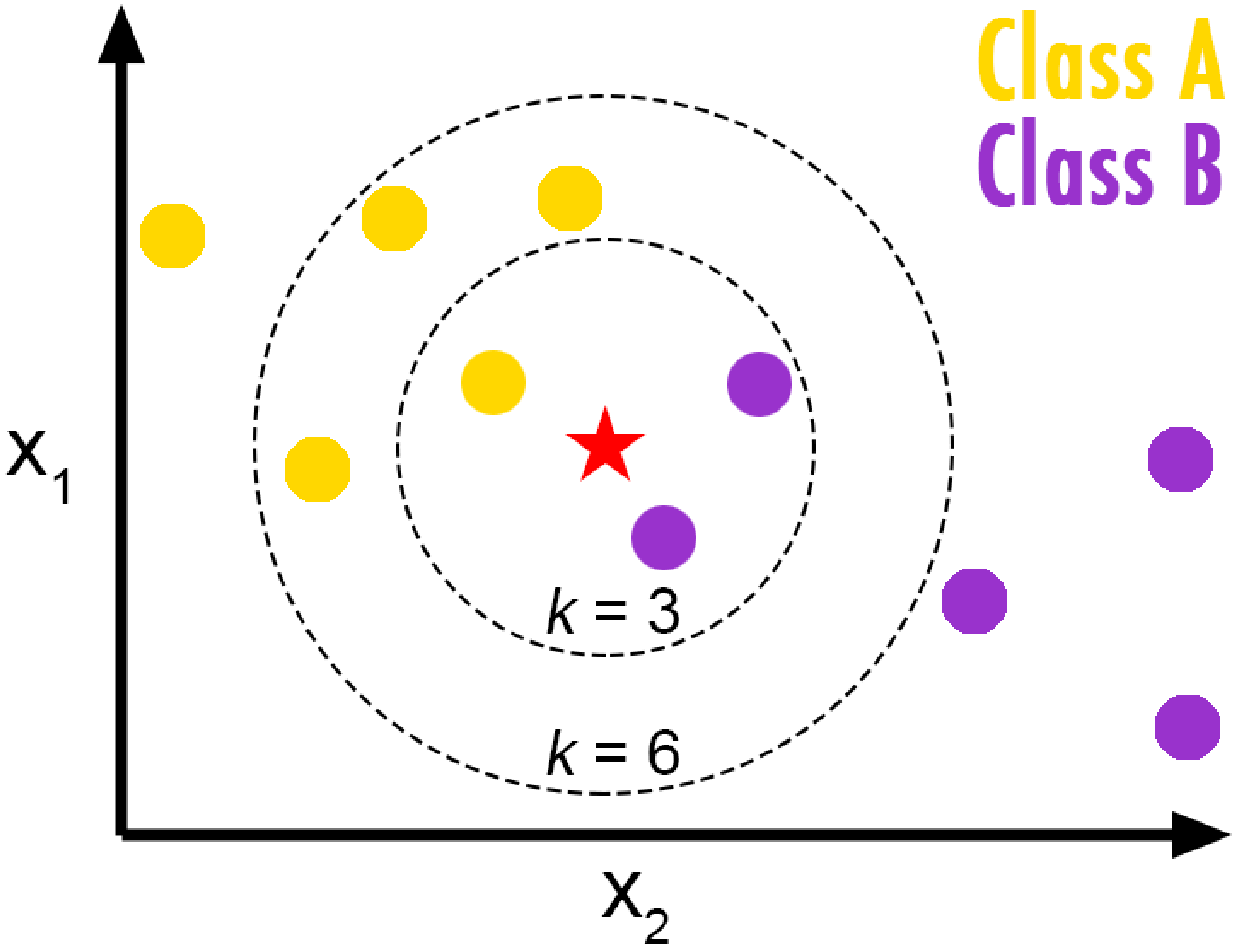
Week 2: K-nearest-neighbors



Week 1 review

- Good vs bad problems
  - Quantifiable/measurable
  - Can collect the data
- Preprocessing
  - Impute missing values (mean, median, mode, KNN)
  - EDA on data, throw out useless features (low variance, etc)
  - Transform (scale data, dummy variables, etc)

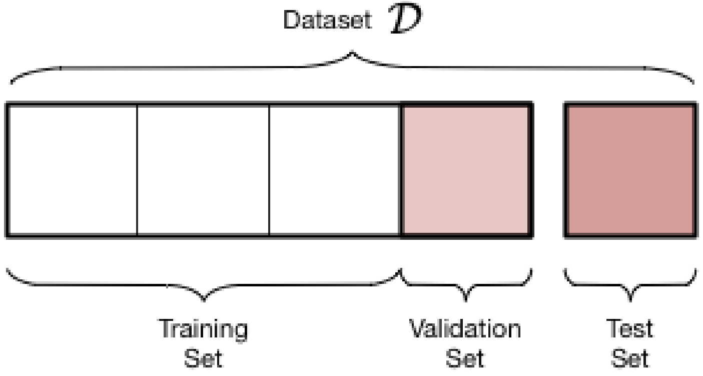
Week 1 quick quiz

- Use the AirQualityUCI.csv file under week 2, and the **NO2(GT)** column
  - Impute missing values twice, once with KNN and once with the overall mean for each column
    - Make 2 histograms of the column with KNN and mean-imputed
    - Make a scatter plot of the imputed values (x-axis should be KNN-imputed, y-axis mean-imputed)
- These are sensor data measure air quality in an Italian city from 2004-2005

http://archive.ics.uci.edu/ml/datasets/Air+Quality

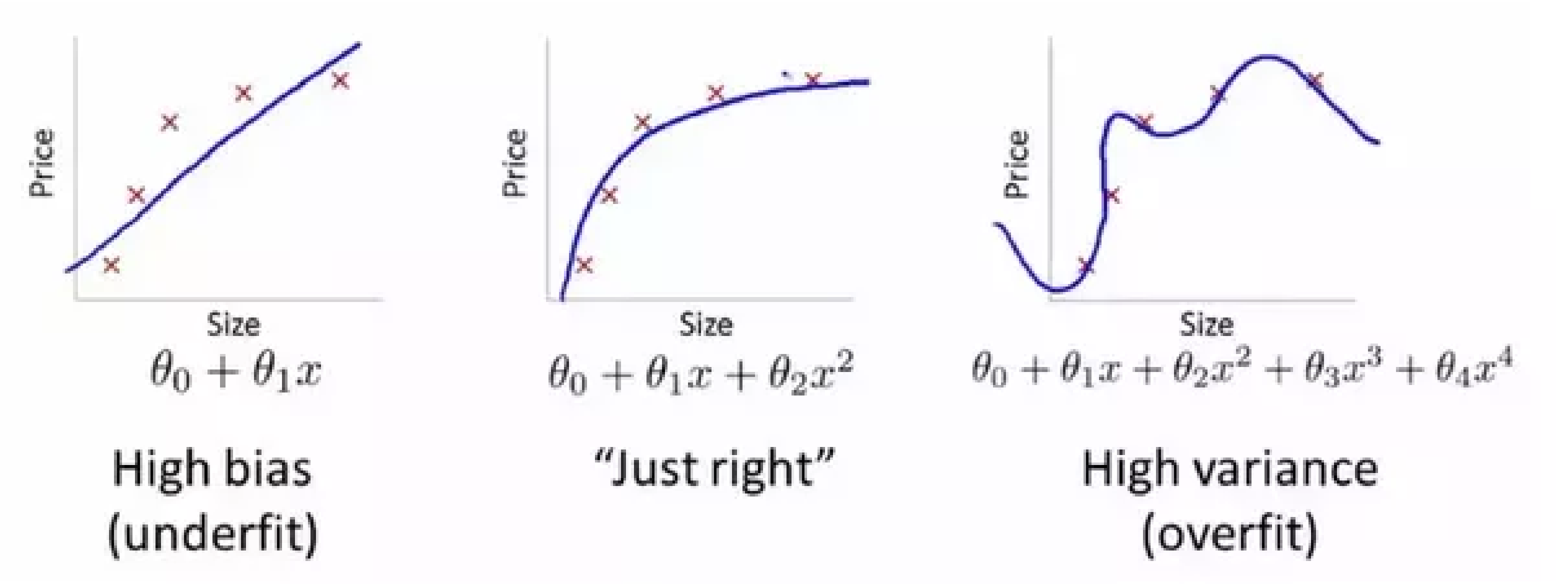
A little more data prep: train/validation/test

- Should split data into training and test, at least
  - Training data is for fitting model
  - Testing is for scoring model
  - Test/train data should not overlap
  - Sometimes test is called ‘holdout’
- Validation is another section we could create, but this is more often used for neural nets



Bias-variance tradeoff and overfitting

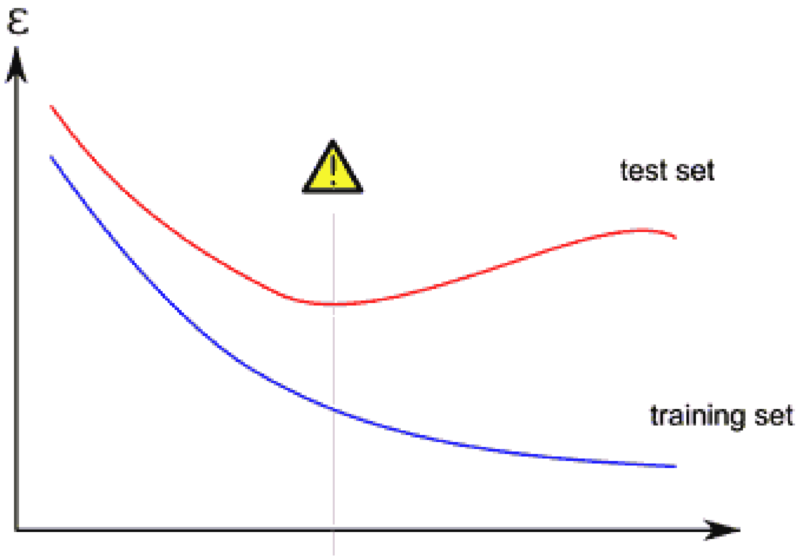
- We can use the validation and/or test sets to:
  - check for overfitting
  - Compare different models (KNN, linear model, etc)



https://www.quora.com/What-is-the-best-way-to-explain-the-bias-variance-trade-off-in-layman%E2%80%99s-terms

Bias-variance tradeoff and overfitting

- We can use the validation and/or test sets to:
  - check for overfitting
  - Compare different models (KNN, linear model, etc)



http://mlwiki.org/index.php/Overfitting

Cross-validation

- We go through a dataset and create train/test splits n times
  - Each time we score our model on the test data, and average the scores
  - This way we get to train on each part of the data, and test on each part separately, so it’s a bit more robust than just
- This is often used to tune hyperparameters
  - Difference between parameters and hyperparameters?

Cross-validation

- This is often used to tune hyperparameters
  - Difference between parameters and hyperparameters?
  - Hyperparameters are set by us, parameters are set by the algorithm
    - e.g. number of neighbors for KNN is a hyperparameter
    - Coefficients from a linear fit are parameters, because the algorithm set those numbers by itself

KNN algorithm

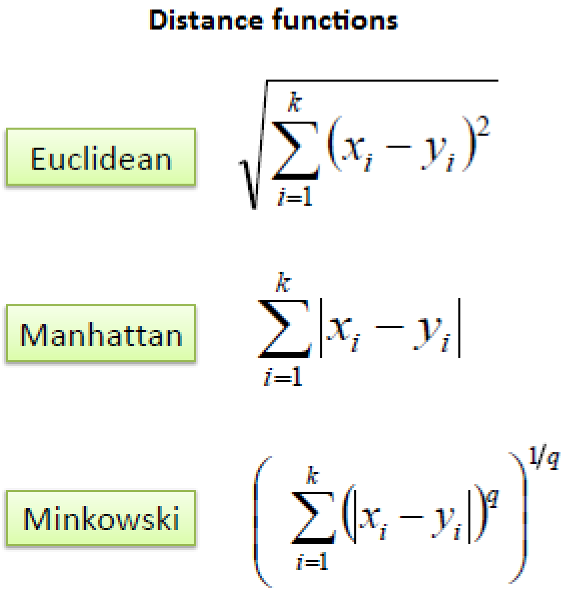
- Give the algorithm training data, and testing data
- For each testing point:
  - Calculate distance from each training point to the testing point
  - Find the *N* closest points based on distance
  - For classification, take the majority vote for most similar class
  - For regression, take the average of the nearest *k* points

Distance metrics

- What are some distance metrics?
  - –
  - –
  - –

Distance metrics

- What are some distance metrics? (at least 3)
  - Euclidean
  - Manhattan
  - Minowski (general formula)



Distance functions

Euclidean $\sqrt{\sum_{i=1}^{k}(x_i - y_i)^2}$

Manhattan $\sum_{i=1}^{k}|x_i - y_i|$

Minkowski $\left(\sum_{i=1}^{k}(|x_i - y_i|)^q\right)^{1/q}$

http://www.saedsayad.com/k_nearest_neighbors.htm

  - Manhattan is best for large number of features, like TFIDF

    https://bib.dbvis.de/uploadedFiles/155.pdf

KNN runtime

- Time complexity (computational complexity), also called big-O runtime
- *n* training samples, *d* features, *k* neighbors
- Either O(ndk) or O(nd + kn) depending on algorithm
- Useful to understand how to reduce runtime if taking too long – either reduce number of samples, number of neighbors, and/or number of features
- What are the runtimes for training and testing?

https://stats.stackexchange.com/a/219664/120921

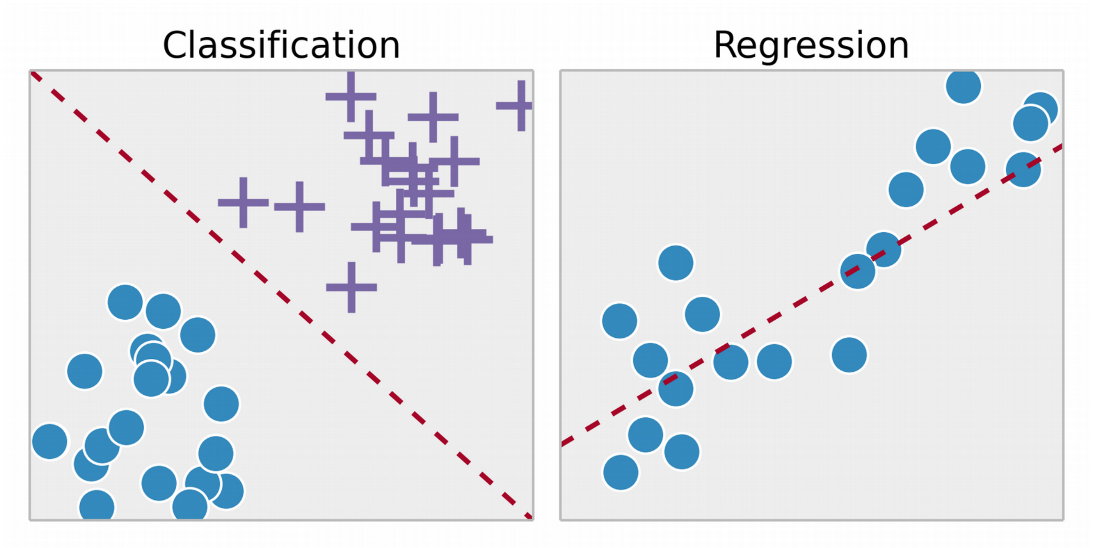
Normalizing features

- What happens if one feature is much larger than another? E.g. weight (1000s) and horsepower (100s)
- How might we normalize the features?

Normalizing features

- What happens if one feature is much larger than another? E.g. weight (1000s) and horsepower (100s)
- How might we normalize the features?
  - Min/max (0-1, -1 to 1, etc)
    - Could be a problem with large outliers
  - Z-scaling (set mean = 0 and stddev = 1)
    - Outliers can still cause problems, but not as bad as min/max
  - Normalize each vector (set total length to 1) https://www.khanacademy.org/computing/computer-programming/programming-natural-simulations/programming-vectors/a/vector-magnitude-normalization
  - Normalize (norm) is not a great idea because the relative magnitudes of features won't change

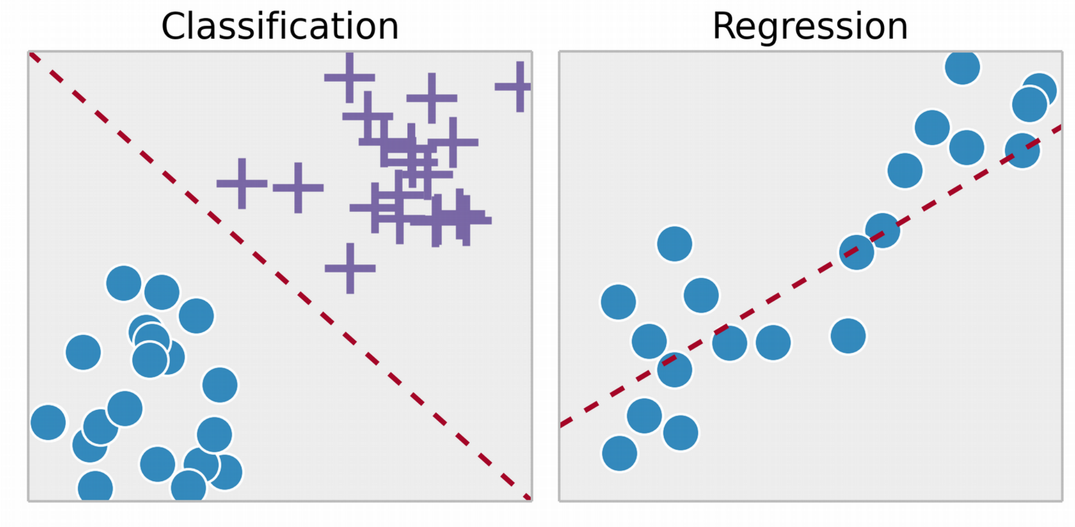
KNN difference for classification vs regression?



http://ipython-books.github.io/featured-04/

KNN difference for classification vs regression?

- Classification takes majority vote of *k* nearest
  - If a tie, could return NA or a random guess of the possibly classes
- Regression takes average of *k* nearest points
  - Could also weight by distance in different ways



http://ipython-books.github.io/featured-04/

Demo: auto mpg

- We are going to predict miles per gallon from other car characteristics.
- Is this classification or regression?
- What are the parameters and hyperparameters in KNN?

Demo: auto mpg

- We are going to predict miles per gallon from other car characteristics.
- Is this classification or regression?
- What are the parameters and hyperparameters in KNN?
  - Hyperparameters are k (number neighbors), distance metric, weighting for predictions
  - Parameters might be considered the nearest neighbors to a point we are trying to predict (debatable, usually parameters are things like coefficients in a linear fit)

Demo: auto mpg

- We are going to predict miles per gallon from other car characteristics.
- Is this classification or regression?
  - Regression, because we are predicting a continuous variable, not a categorical variable
- Use the auto.dt.nona.csv file (which has NAs replaced with KNN imputation)
- Use the knn_demo.R and knn_demo.Rmd files
- Also knn_demo.ipynb file for Python (jupyter notebook)

Project: heart disease prediction

- Task – either predict the value of ‘num’ (how much blood vessels have narrowed due to plaque, 0 is not much and 4 is a lot). 4 is the worst case of heart disease (most plaque buildup in vessels)
- Or predict if the ‘num’ column is 0 or >= 1. If >= 1, the person has heart disease.
- Before you do KNN, you will need to answer the question and decide: are you doing classification or regression?
- Tune the k hyperparameter to the optimal value and support it with data (elbow plot)
- Report accuracy and/or other scoring metrics (confusion matrix, R^2, SSE, etc) on train and test data sets
- Make at least 2 plots. Options:
  - plot PCA dimensions of data (e.g. 1st and 2nd) and color points by classes
  - Plot some of the variables on x and y and color by classes
  - Color by errors/misclassifications
  - More EDA

Ideas to try if you want more

- Try classification instead of regression
  - Any of the UCI classification datasets are good
- Look more into PCA
- Explore other R packages for KNN – there are many
- Try using other distance and weighting settings for the KNN hyperparameters