CS 559: Linear Regression & K-Nearest Neighbors

Lecture 2-2

In Jang
ijang@stevens.edu

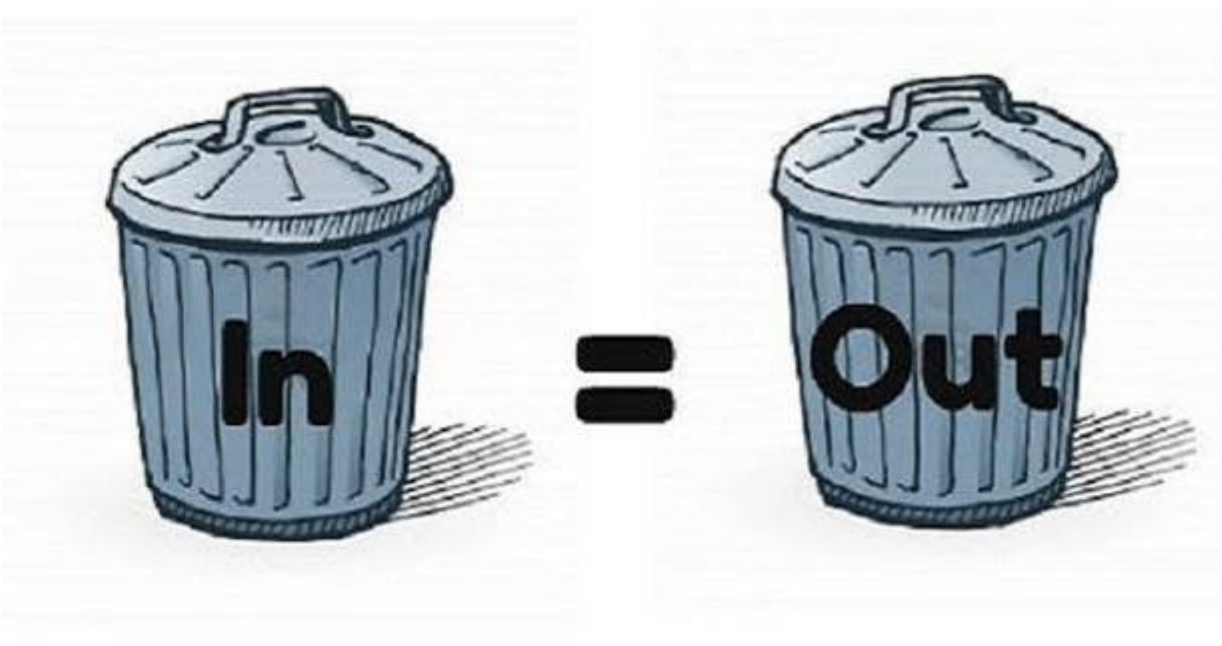
STEVENS
INSTITUTE of TECHNOLOGY
THE INNOVATION UNIVERSITY®
1870

Outline

- Importance of data preprocessing
  - Missing data, scaling data, encoding categorical data
- Supervised vs unsupervised ML
- General paradigm of machine learning
- Supervised learning – two examples
- Parametric Supervised learning
  - Linear regression & its nonlinear extension
- Nonparametric supervised learning
  - KNN for both regression and classification

Importance of data preprocessing

- Data preprocessing is to make sure we have sensible data for ML



[https://www.linkedin.com/pulse/supply-chain-planning-garbage-out-cheryl-wiebe]

Some data issues need to be addressed before applying ML algorithms

- Missing values
  - Observation we intended to collect but did not get them
    - Data entry issues, equipment errors, incorrect measurement etc
      - An individual may only have responded to certain questions in a survey, but not all
  - Problems of missing data
    - Reduce representativeness of the sample
    - Complicating data handling and analysis
    - Bias resulting from differences between missing and complete data
- Data in different scales
  - Weight of a person (Pounds) vs weight of an elephant (US ton)
    - 1 US ton = 2000 Pounds
  - For predicting weights for them, the error of elephant weights will significantly bias the prediction accuracy relative to the error for the persons weights

Missing data handling

- Reducing the data set
  - Elimination of samples with missing values
  - Elimination of features (columns) with missing values

- Imputing missing values
  - Replace the missing value with the mean/median (numerical) or most common (categorical) value of that feature

- Treating missing attribute values as a special value
  - Treat missing value itself as a new value and be part of the data analysis

"Applied Missing Data Analysis" C. Enders, 2010

Data in different scale

- Approaches to bring different values onto the same scale
  - Normalization: rescale the feature to a range of [0,1]
  - Standardization: re-center the feature to the mean and scaled by variance

$$x_{norm}^{(j)} = \frac{x^{(j)} - x_{min}}{x_{\max} - x_{min}}$$

x_{min} and x_{max} are the min/max values of feature column $x^{(j)}$

$$x_{std}^{(j)} = \frac{x^{(j)} - \mu_x}{\sigma_x}$$

μ_x and σ_x are the mean and standard deviation of feature column $x^{(j)}$

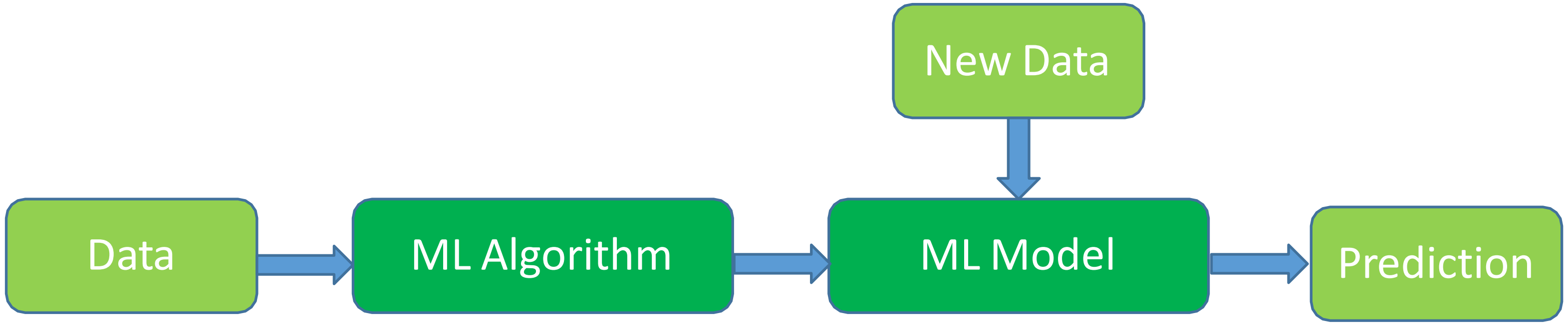
- Data scaling should be one of the first steps of data preprocessing for many machine learning algorithms
  - Some machine learning algorithms can handle data in different scales (e.g., decision trees and random forests)

Categorical data handling

- for ordinal data, convert the strings into comparable integer values
  - E.g., XL > L > M > S → 5 (XL) > 4 (L) > 3 (M) > 2 (S)
  - Note that the value of integer itself has no special meaning besides for ordering
  - Mapping needs to be unique: 1 to 1 mapping for going back and forth
- For nominal data, convert the strings into integers
  - E.g., Red (0), Blue (1), Green (2)
  - A common practice to avoid software glitches in handling strings
  - Note that the value of integer itself has no special meaning (non-comparable)
  - Mapping needs to be unique: 1 to 1 mapping for going back and forth
- To avoid mistakenly comparing encoded integers for nominal data, one-hot encoding can be used
  - Each unique value becomes a separate dummy feature

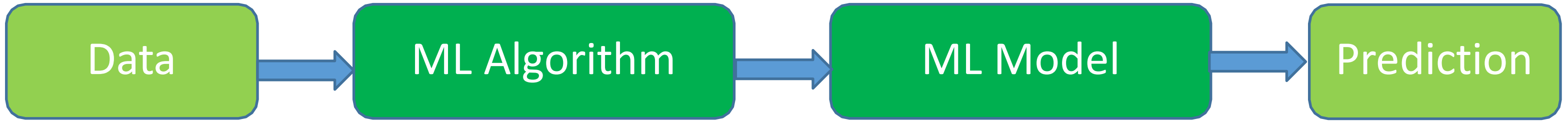
Machine learning, models and data

- Machine learning is an algorithm that learns a model from data (training), so that the model can be used to predict certain properties about new data (generalization)

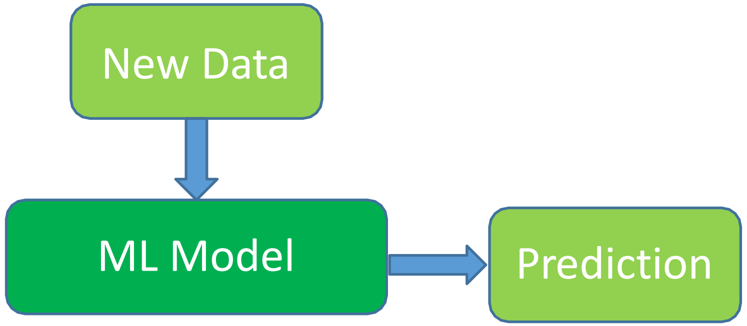


Training vs Inference

- Training is to build the ML model from data



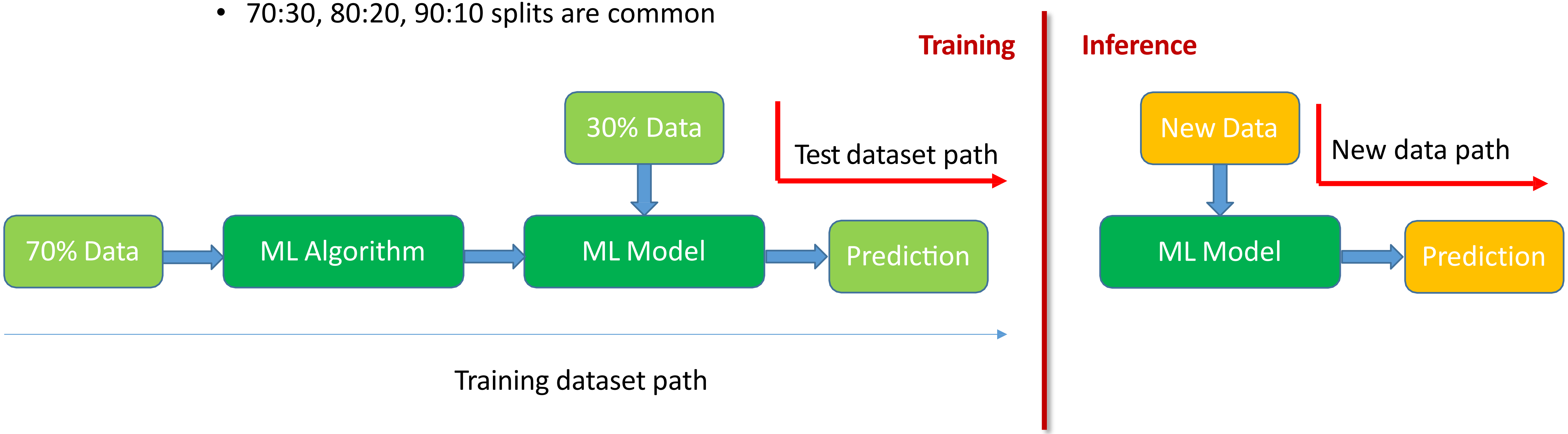
  - Typically, training is a one-time effort, but computationally intensive
  - Speed is a main concern
- Inference is to use the ML model to predict results for new data (generalization – most interesting for applications)



  - Typically, inference is fast but happens more frequently with a lot of more new data (unlabeled)
  - Scalability is a main concern

Split known data into training and test datasets

- Data known to ML model developers are split into two sets
  - Training dataset: data used to train the model
  - Test dataset: data used to give an indication on how well the trained model will generalize to new data (unknown at this point)
    - Test dataset is kept till the very end to evaluate the final model
    - Since test dataset withholds valuable information that the learning algorithm could benefit from, we don't want to put too much data into the test dataset either
      - 70:30, 80:20, 90:10 splits are common



What data to use for learning and what to learn? Supervised vs Unsupervised

- If data contains a set of features (factors) that are easy to obtain in practice, and at least one feature that is difficult to obtain
  - The goal of ML becomes clear: can we build a ML model that can help predict that one “difficult” feature based on a set of “easy” features?
  - If the feature of interest is numerical, ML = regression
  - If the feature of interest is categorical (mostly nominal), ML = classification
    - If two categories, ML = binary classification
    - If multiple categories, ML = multiclass classification
  - We’re typically given a set of data with features of interest clearly labeled, and we use these data to train a ML model. This is also called “supervised” learning
- If all features in the data are equally easy to obtain, which is a nice way to say that we can’t precisely define what to look for, ML becomes an exploration problem
  - Discover the hidden structures, such as clustering
  - This is also called “unsupervised” learning

Supervised learning – an example

- Let's look at the Advertising data set
  - http://www-bcf.usc.edu/~gareth/ISL/Advertising.csv
- In the data
  - Sales for each of 200 products
  - Advertising budgets for each product in TV, radio and newspaper

- If you have a new product, how should you spend your advertising money to generate the most sales? How much? In which media?

- How can we use data to answer these questions?

What question to ask is important for ML

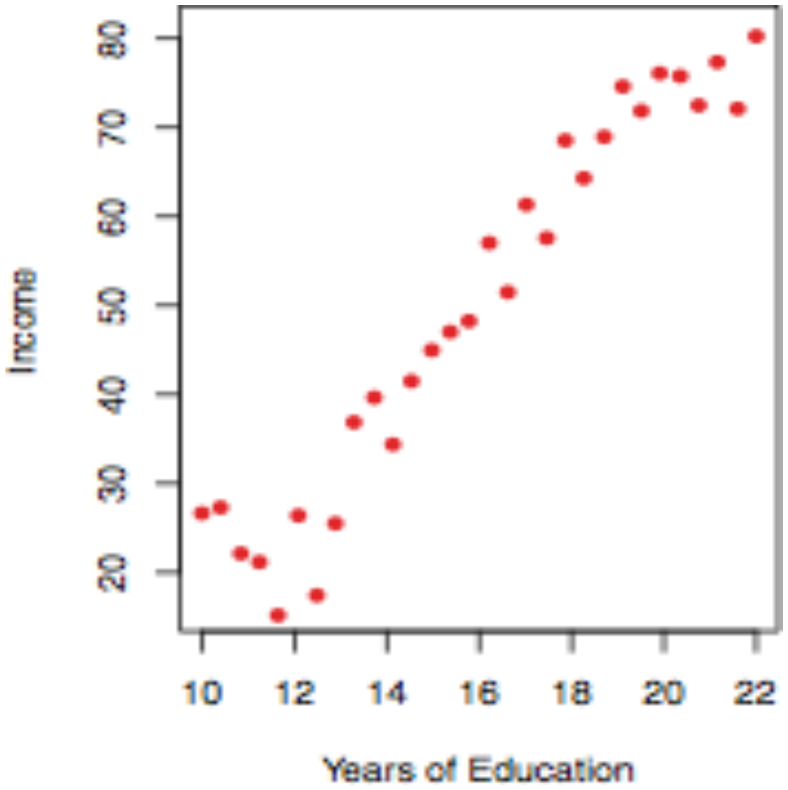
If you have a new product, how should you spend your advertising money to generate the most sales?

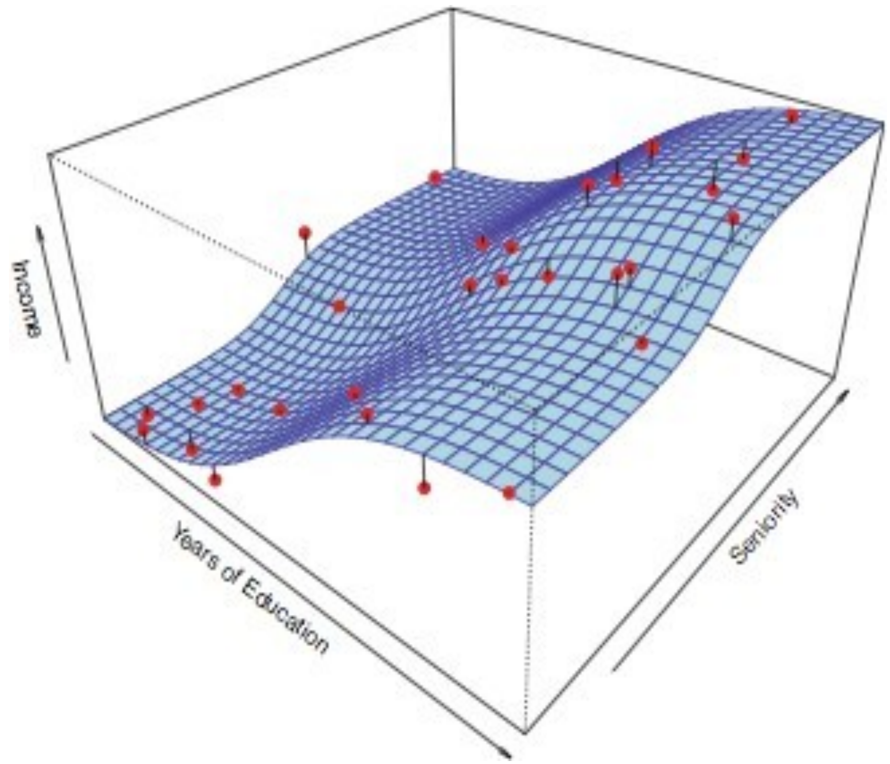
- Idea: predict the level of sales from TV, radio and newspaper's advertising budgets, respectively

- Input variables (covariates, independent variables, predictors, features):
  - TV
  - Radio
  - Newpaper
- Output variable (response, dependent variable):
  - Sales

Supervised learning – another example

- We want to predict the annual income (income) of an individual based on years of education (years of education)
  - Income data from ISL
- You can also consider seniority as another feature as well





Supervised learning – function fitting

- Suppose we have n observations, $(x_1, y_1), \ldots, (x_n, y_n)$, where each x_i is a vector of features, and y_i is the corresponding feature of interest
  - We know what question of interest to ask

- To answer the question, we need to estimate the relationship

$$Y = f(X) + \epsilon.$$

- How to find an estimate of function f(x)?

Parametric vs non-parametric models

- Parametric models
  - The function can be described by a finite number of parameters
  - The interactions between inputs, parameters and outputs are well described (when is this useful?)
  - But sometimes, it comes with limited flexibility (desirable or undesirable?)
  - Example: linear regression
- Non parametric models
  - The function can NOT be described by a finite number of parameters
  - Example: KNN

- Question: for the MLE model we discussed last time, which one does it belong it?

Regression and simple linear regression

- Regression: a ML technique to summarize relationships between continuous variables
  - One is regarded as response, outcome, dependent, or target variable
  - The rest is regarded as predictor, explanatory or independent variable(s)
- Simple linear regression
  - Simple: study only one predictor variable
  - Linear: the relationship is linear
- The ML model to predict the response

$$y \approx H(w, x) = w_0 + w_1 x$$

where y is the observed outcome, e.g., target variable, x is the explanatory variable, $w = [w_0, w_1]^T$ is the model parameters to be determined from ML training, and *H(w,x)* is the linear regression model that would give the predicted target variable.

About the housing dataset

- https://archive.ics.uci.edu/ml/machine-learning-databases/housing/
  - 1. CRIM: per capita crime rate by town
  - 2. ZN: proportion of residential land zoned for lots over 25,000 sq.ft.
  - 3. INDUS: proportion of non-retail business acres per town
  - 4. CHAS: Charles River dummy variable (= 1 if tract bounds river; 0 otherwise)
  - 5. NOX: nitric oxides concentration (parts per 10 million)
  - 6. RM: average number of rooms per dwelling
  - 7. AGE: proportion of owner-occupied units built prior to 1940
  - 8. DIS: weighted distances to five Boston employment centres
  - 9. RAD: index of accessibility to radial highways
  - 10. TAX: full-value property-tax rate per $10,000
  - 11. PTRATIO: pupil-teacher ratio by town
  - 12. B: 1000(Bk - 0.63)^2 where Bk is the proportion of blacks by town
  - 13. LSTAT: % lower status of the population
  - 14. MEDV: Median value of owner-occupied homes in $1000's

Two simple but often ignored questions before applying any ML algorithms

- First, we need to identify what data feature to predict from the model
  - Assume we're interested in predicting the housing prices

0.00632	18.00	2.310	0	0.5380	6.5750	65.20	4.0900	1	296.0	15.30	396.90	4.98	24.00
0.02731	0.00	7.070	0	0.4690	6.4210	78.90	4.9671	2	242.0	17.80	396.90	9.14	21.60
0.02729	0.00	7.070	0	0.4690	7.1850	61.10	4.9671	2	242.0	17.80	392.83	4.03	34.70
0.03237	0.00	2.180	0	0.4580	6.9980	45.80	6.0622	3	222.0	18.70	394.63	2.94	33.40
0.06905	0.00	2.180	0	0.4580	7.1470	54.20	6.0622	3	222.0	18.70	396.90	5.33	36.20
0.02985	0.00	2.180	0	0.4580	6.4300	58.70	6.0622	3	222.0	18.70	394.12	5.21	28.70
0.08829	12.50	7.870	0	0.5240	6.0120	66.60	5.5605	5	311.0	15.20	395.60	12.43	22.90
0.14455	12.50	7.870	0	0.5240	6.1720	96.10	5.9505	5	311.0	15.20	396.90	19.15	27.10
0.21124	12.50	7.870	0	0.5240	5.6310	100.00	6.0821	5	311.0	15.20	386.63	29.93	16.50

  - So the last column MEDV becomes our target variable (and it's numerical)
- Second, what explanatory variables do we use for ML
  - These could be determined/given by the application in hand as well
    - What explanatory variables are easy to get in practice?
  - We can also use whatever data features we have (all other features)
    - Will computation become an issue? Are all of them good quality data?
  - We can explore the data to draw a sensible set of features (e.g., feature selection)

Exploratory data analysis

- Let’s visualize pair-wise scatterplots correlations between features

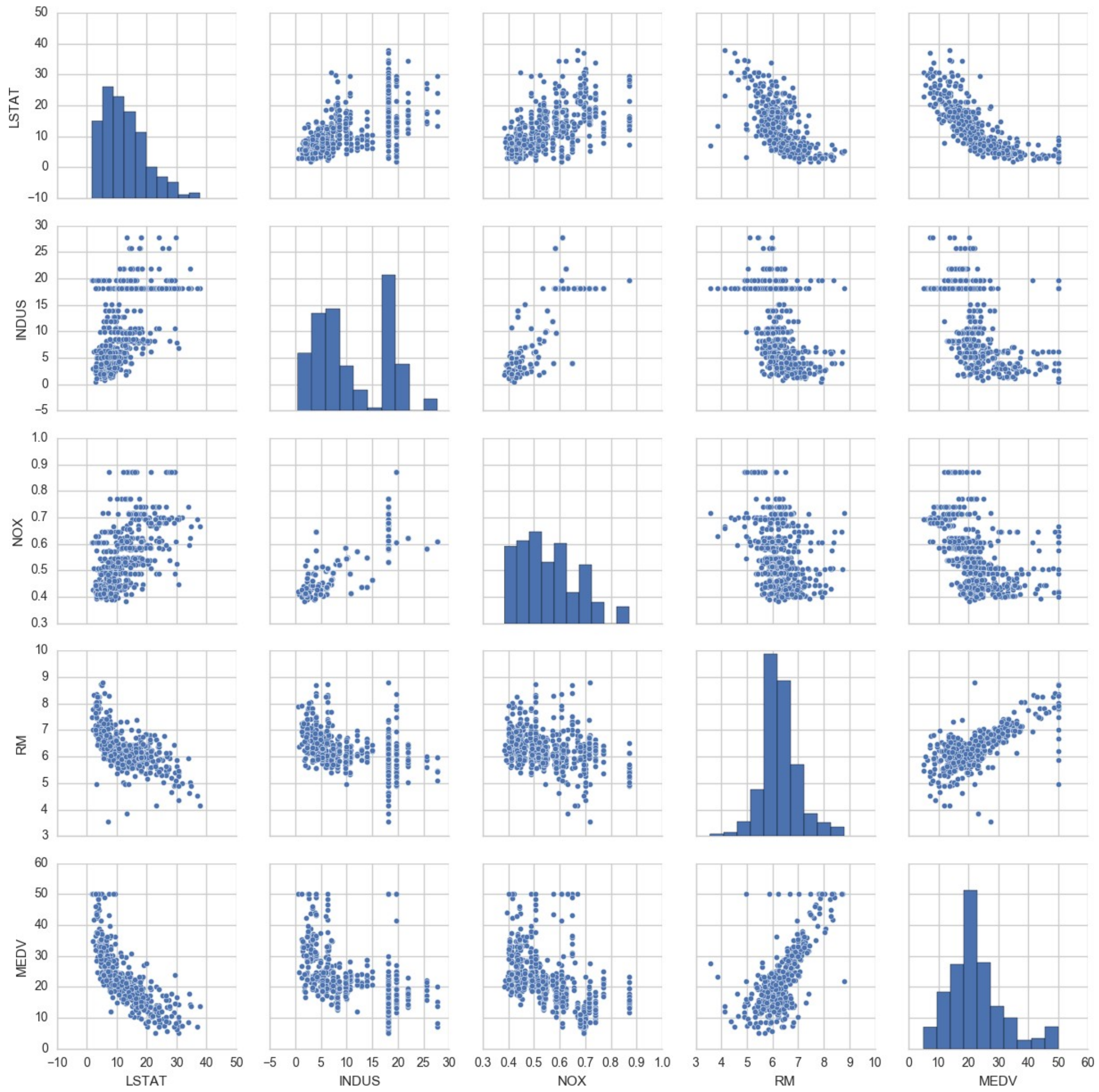
LSTAT: % lower status of the population

INDUS: proportion of non-retail business acres per town

NOX: nitric oxides concentration (parts per 10 million)

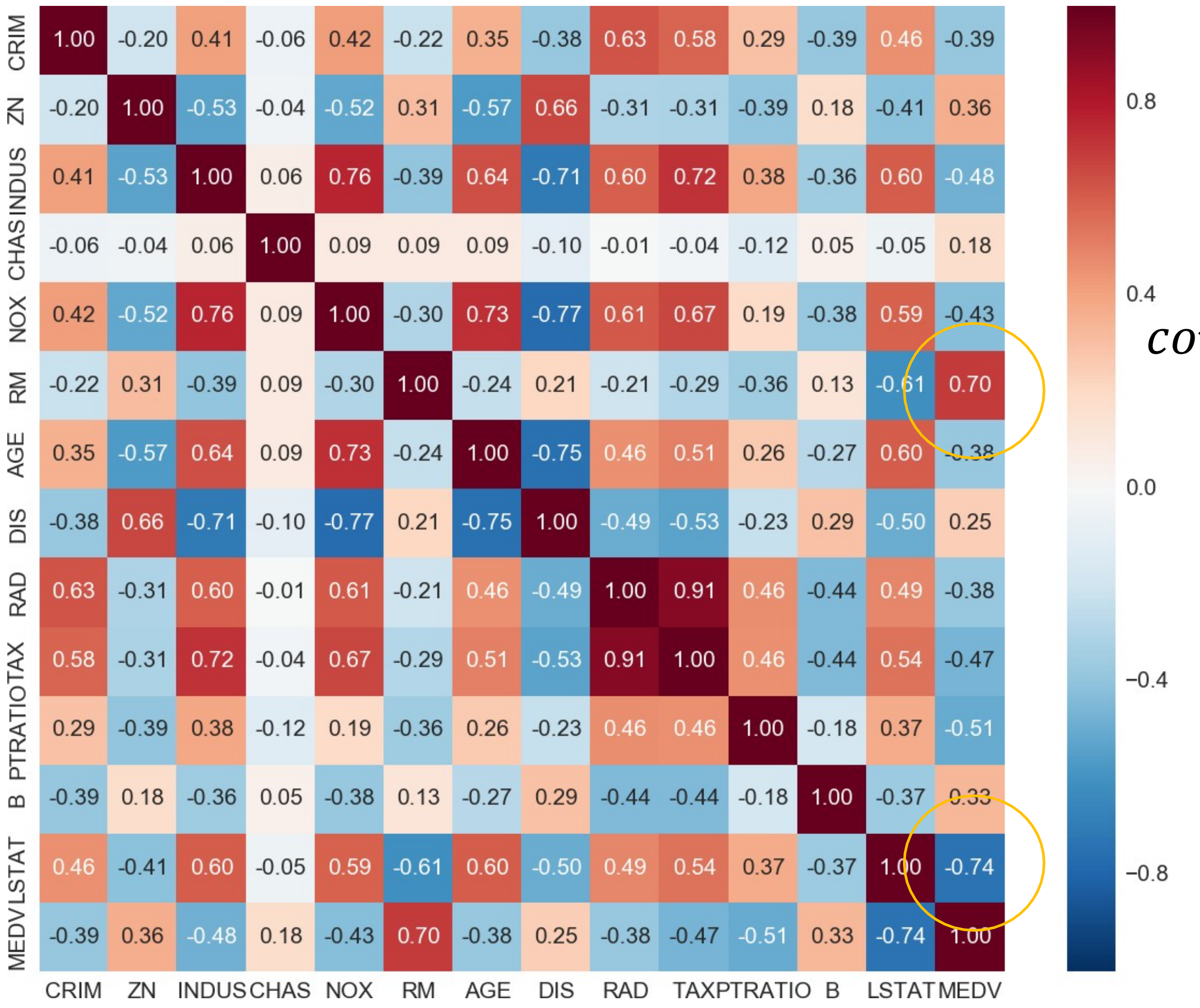
RM: average number of rooms per dwelling

MEDV: Median value of owner-occupied homes in $1000's



Correlation matrix for the data

- Two features (RM and LSAT) have higher correlation with MEDV, implying high chances of being explanatory variables



$$correlation = \rho = \frac{Cov(x,y)}{\sigma_x \sigma_y}$$

Simple linear regression example

- The explanatory variable is chosen as RM (average number of rooms per dwelling)
- The target variable is chosen as MEDV (Median value of owner-occupied homes in $1000's)
- The ML model to predict the response

$$H(w, x) = w_0 + w_1 x$$

where y is the observed outcome, e.g., target variable, x is the explanatory variable, $w = [w_0, w_1]^T$ is the model parameters to be determined from ML training

What is the best fit?

- Prediction Error

$$e^i = y^i - H(w, x^i)$$

where (x^i, y^i) represents the i-th observed data (RM, MEDV)

- Problem formulation: minimize Sum of Squared Errors (SSE aka Least Square Problem)

$$Minimize_w \ f(w) = \frac{1}{2}\sum_{i=1}^{n}\left(y^i - H\left(w, x^i\right)\right)^2$$

where n is the number of total observations used for training

- This is a typical unconstrained optimization problem, more specifically, quadratic programming problem

$$Minimize_w \ f(w) = \frac{1}{2}\sum_{i=1}^{n}\left(y^i - w_0 - w_1 x^i\right)^2$$

Analytic solution

- For this simple formulation, we can apply calculus to find the optimal solution by setting the gradient to be zero

$$\nabla f(w) = \begin{bmatrix} \frac{\partial f(w)}{\partial w_0} \\ \frac{\partial f(w)}{\partial w_1} \end{bmatrix} = 0$$

- This will lead to the solution as

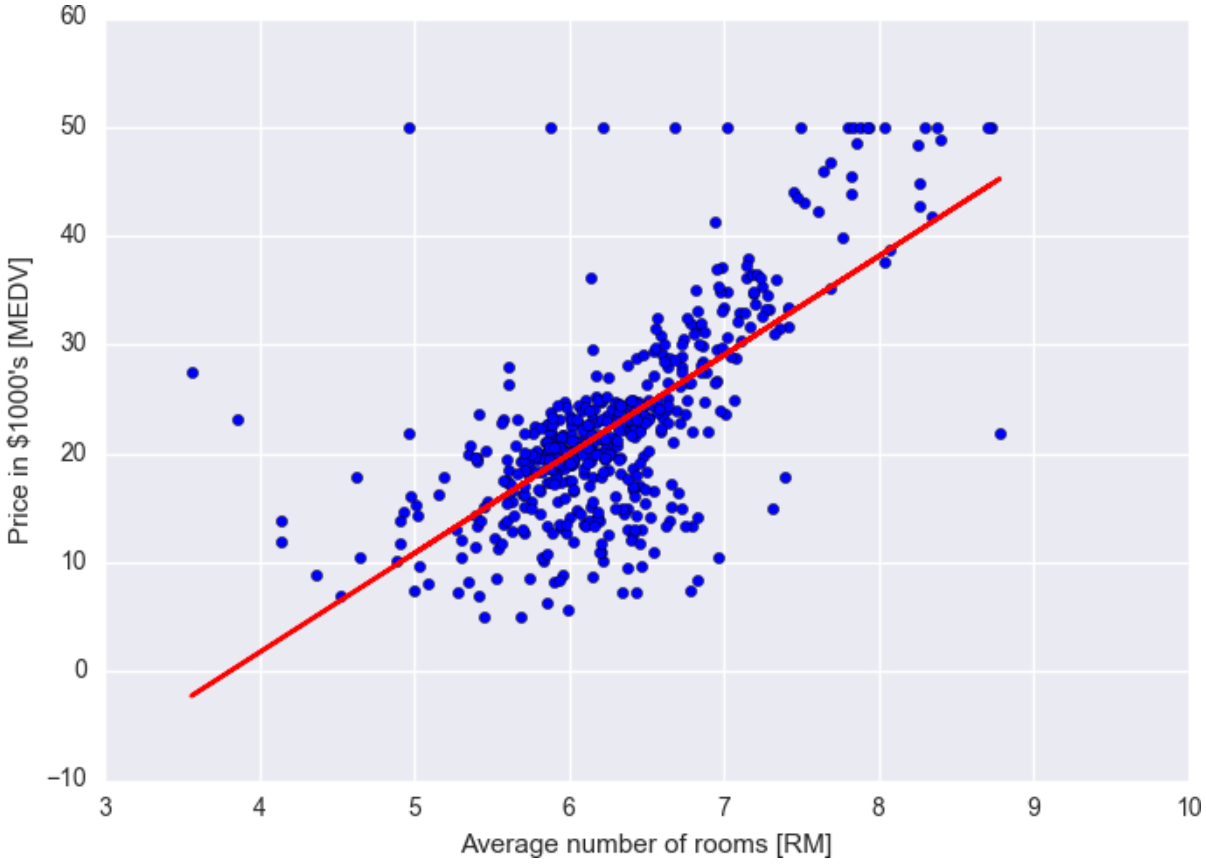
$$w_1 = \frac{\sum_{i=1}^{n} (x^i - \bar{x})(y^i - \bar{y})}{\sum_{i=1}^{n} (x^i - \bar{x})^2}$$

$$w_0 = \bar{y} - w_1 \bar{x}$$

- The so-obtained solution is also called the least square (regression) line

Solution to the simple linear regression model

- Is the model good enough?



Extensions of linear regression formulations

- Multiple linear regression model

$$H(w, x) = w_0 + \sum w_i x_i$$

- Polynomial regression

$$H(x, w) = w_0 + w_1 x + w_2 x^2 + \cdots + w_d x^d$$

- Nonlinear transformation of some data features
  - For example, log transform the LSAT data

$$H(w, x) = w_0 + w_1 x_1 + w_2 \log x_2$$

- Since the unknown model parameters are linear, they all end up with the same quadratic-like formulation

Multi-linear regression formulation

- The analytic solution is still possible, but not so obvious

$$\nabla f(x) = \begin{bmatrix} \frac{\partial f(w)}{\partial w_0} \\ \vdots \\ \frac{\partial f(w)}{\partial w_n} \end{bmatrix} = 0$$

- The solution is given by

$$w = (X^T X)^{-1} X^T Y$$

  - Where the definition of matrix X depends on the exact form used

$$H(w,x) = w_0 + w_1 x_1 + \cdots + w_m x_m$$

$$H(w,x) = w_0 + w_1 x_1 + w_2\, log(x_2) + \cdots + w_m\, log(x_m)$$

$$X = \begin{bmatrix} 1 & x_1^{(1)} & x_2^{(1)} & \cdots & x_m^{(1)} \\ 1 & x_1^{(2)} & x_2^{(2)} & \cdots & x_m^{(2)} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & x_1^{(n)} & x_2^{(n)} & \dots & x_m^{(n)} \end{bmatrix}$$

$$X = \begin{bmatrix} 1 & x_1^{(1)} & \log(x_2^{(1)}) & \cdots & \log(x_m^{(1)}) \\ 1 & x_1^{(2)} & \log(x_2^{(2)}) & \cdots & \log(x_m^{(2)}) \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & x_1^{(n)} & \log(x_2^{(n)}) & \dots & \log(x_m^{(n)}) \end{bmatrix}$$

Nonlinear regression

- So far our regression models are all linear w.r.t. model parameters

$$H(w,x) = w_0 + w_1 x_1 + \cdots + w_m x_m$$
$$H(w,x) = w_0 + w_1 x + w_2 x^2 \ldots + w_d x^d$$
$$H(w,x) = w_0 + w_1 x_1 + w_2\, log(x_2) + w_3\, log^2(x_2)$$

- Nonlinear regression is simply a replacement of the model with a nonlinear function w.r.t. parameters

$$H(w,x) = e^{w_0 + w_1 x_1} + w_2 x_2$$

$$H(w,x) = \frac{w_0 + w_1 x_1}{1 + w_2 x_2}$$

- Note some nonlinear form can be transformed to be a linear problem

$$y = H(w,x) = e^{w_0 + w_1 x_1} \rightarrow \ln(y) = w_0 + w_1 x_1$$

Nonlinear regression formulation

- It is almost identical to linear regression, except the ML model

$$\text{Minimize}_w \; f(w) = \frac{1}{2}\sum_{i=1}^{N}\left(y^{(i)} - H\left(w, x^{(i)}\right)\right)^2$$

- The difficulty level depends on the computation of model gradients
  - More on this later

KNN - K nearest neighbors

- Idea: average the values of the k closest observations

$$\hat{Y} = \frac{1}{k} \sum_{x_i \in N_k(x)} y_i$$

- where $N_k (x)$ is the set of observations with the k smallest distances to the query point x
    - Classification: pick the majority label
    - Regression: average the values

- Why is this a nonparametric model? Does it have parameters?

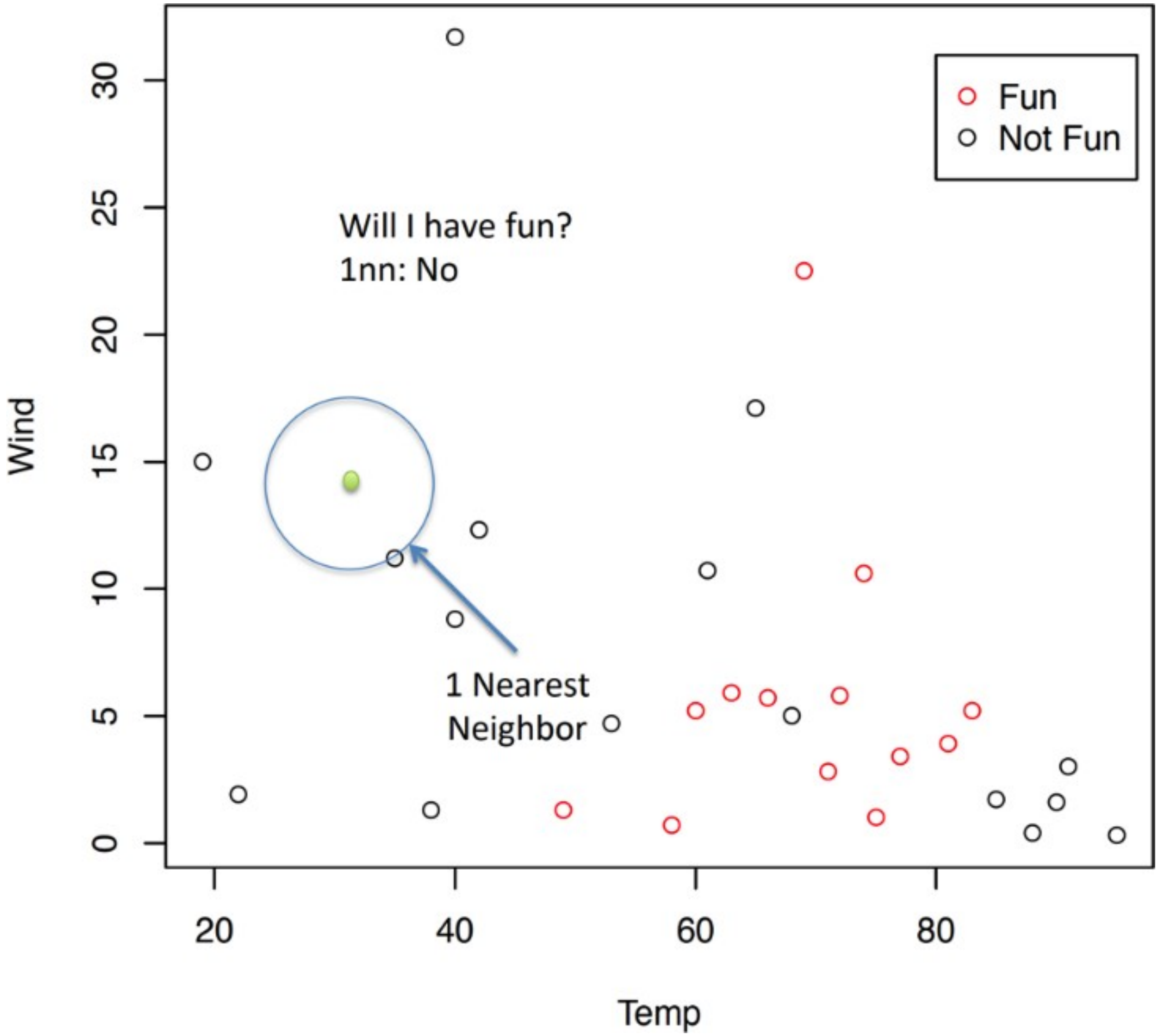
KNN – an example

- Is it a good day to go for a run?

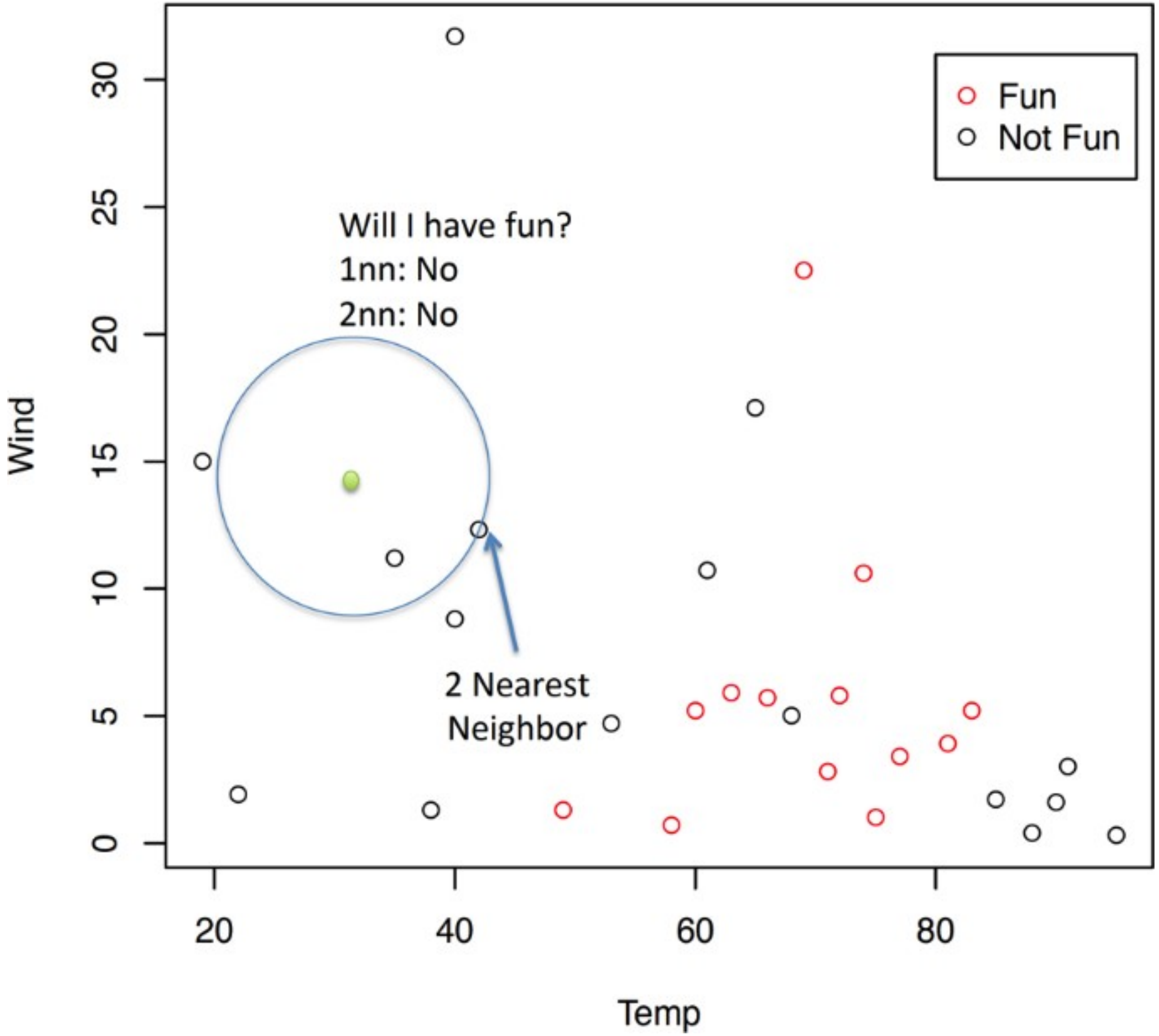
- A runner´s data on past running. With recorded temperature, wind and whether the run was fun:
  - Temperature (degrees F)
  - Wind Speed (mph)
  - Fun (yes, no)

- It is now 65 degrees and the wind is 9 mph. Will a run be fun?

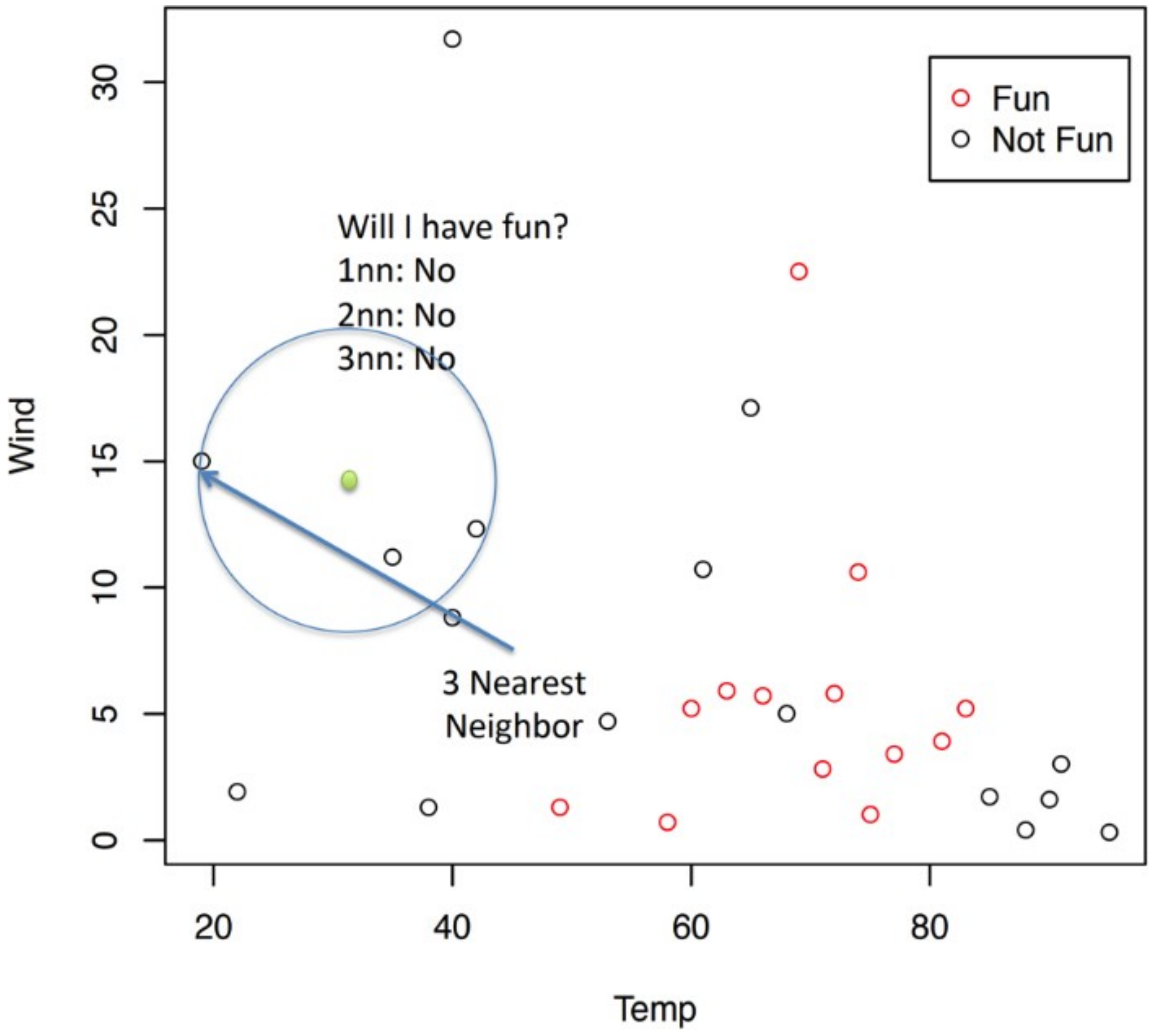
KNN classification: k=1



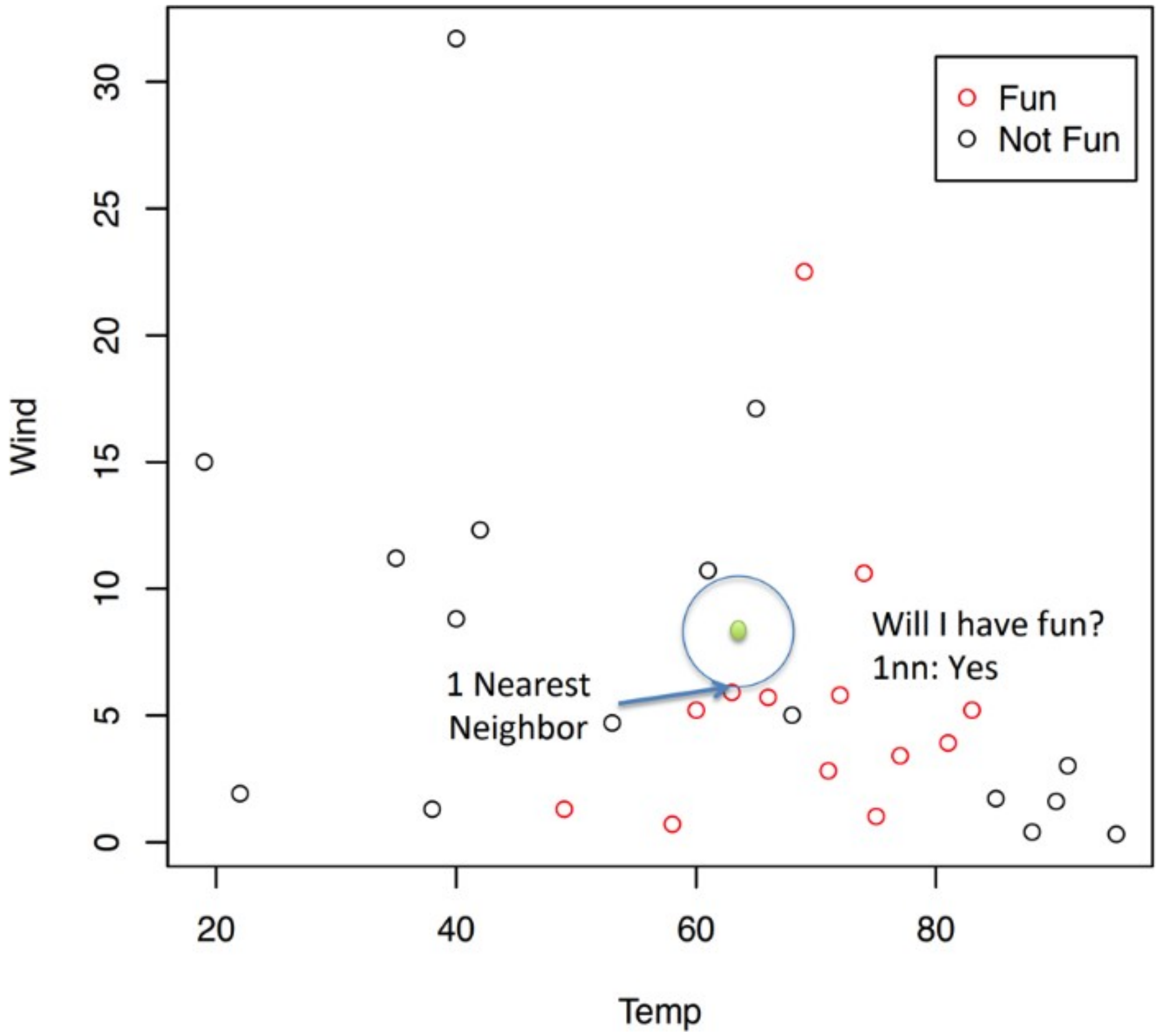
KNN classification:
k=2



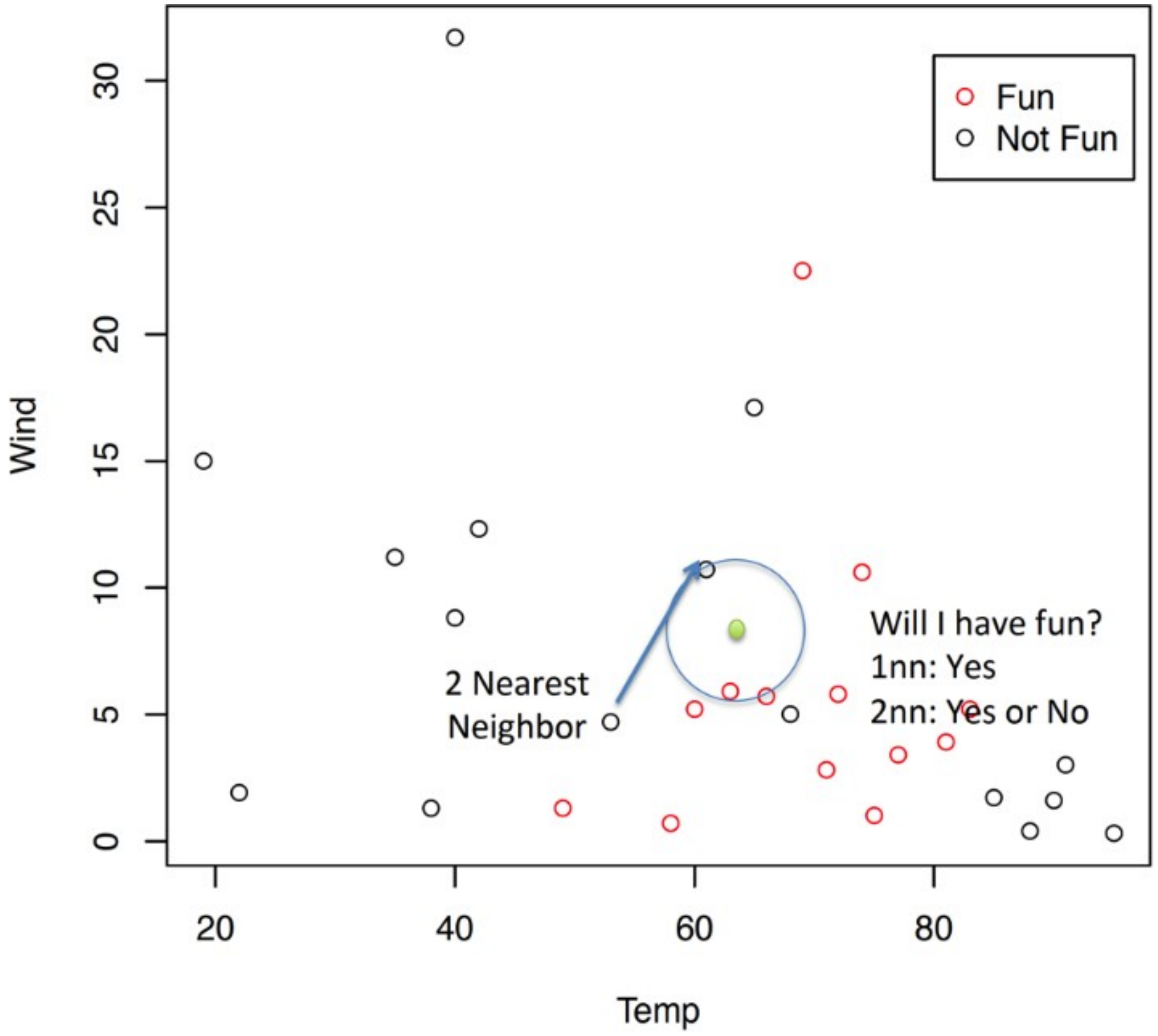
KNN classification: k=3



KNN classification: k=1



KNN classification: k=2



KNN classification: k=3