Machine Learning Crash Course

Week 2

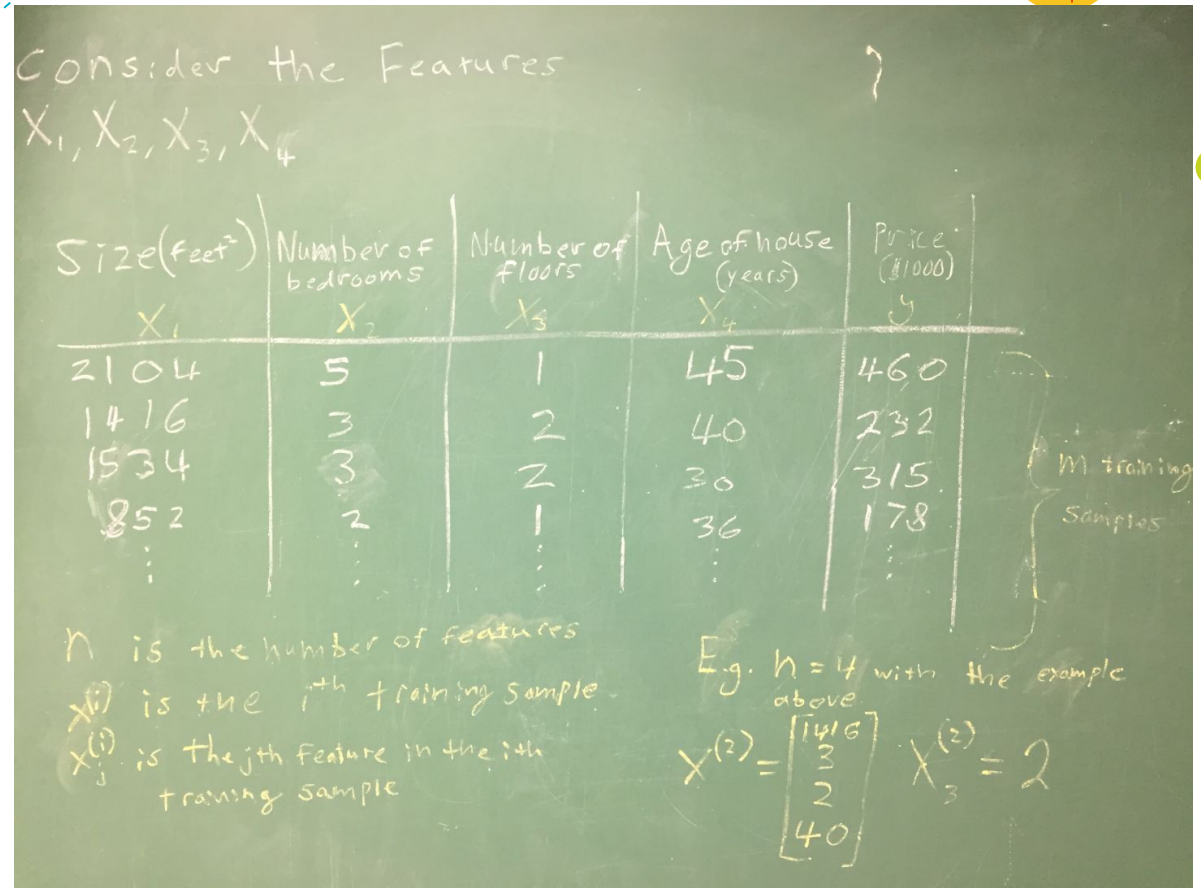
“The pace of progress in artificial intelligence (I’m not referring to narrow AI) is incredibly fast. Unless you have direct exposure to groups like Deepmind, you have no idea how fast—it is growing at a pace close to exponential. The risk of something seriously dangerous happening is in the five-year timeframe. 10 years at most.” —Elon Musk

Multivariate Linear Regression

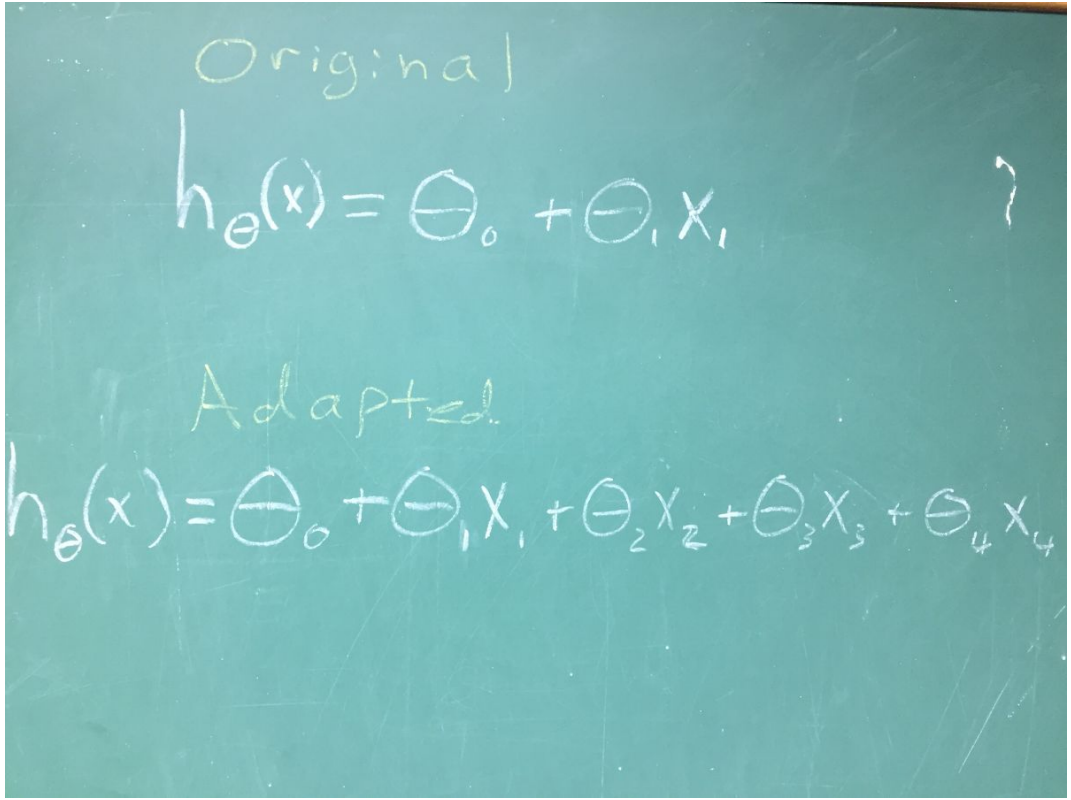
Today's Topics

- Introducing multiple features to the data
- Adapting the Gradient Descent function
- Polynomial Regression
- Normal Equations
- Normal Equations with Non-Invertibility

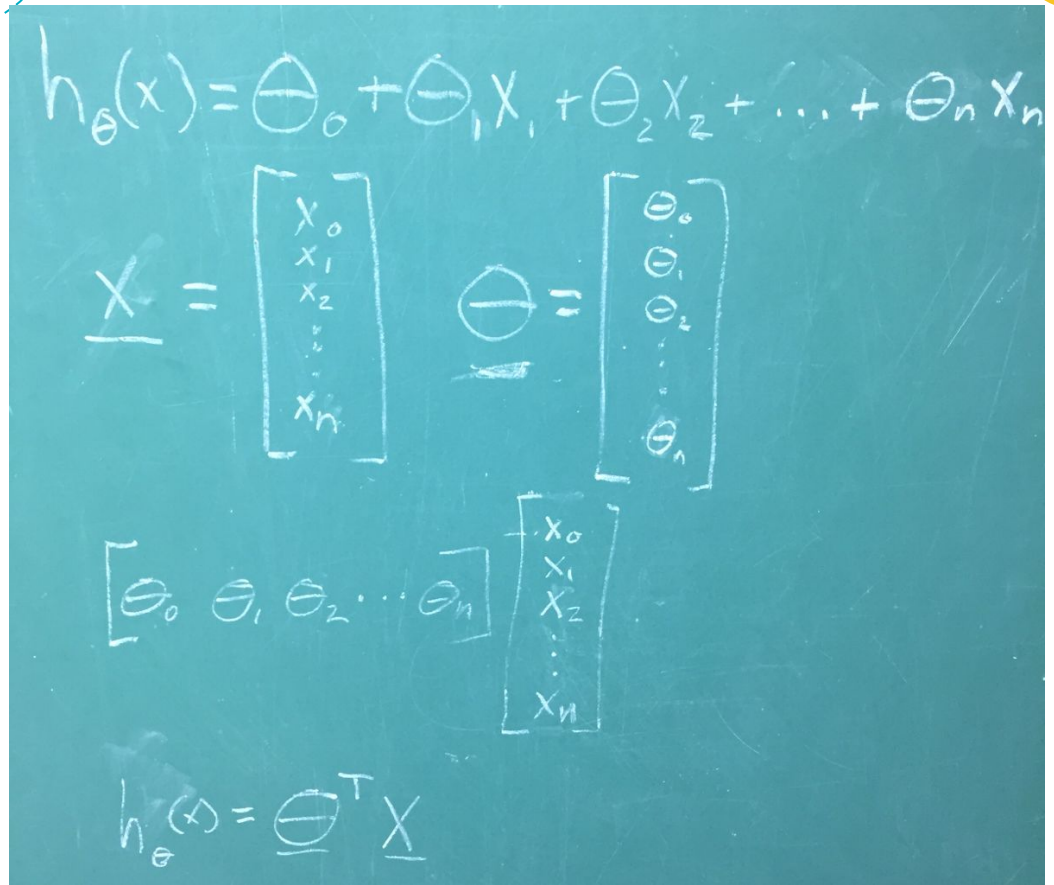
Housing Price Prediction



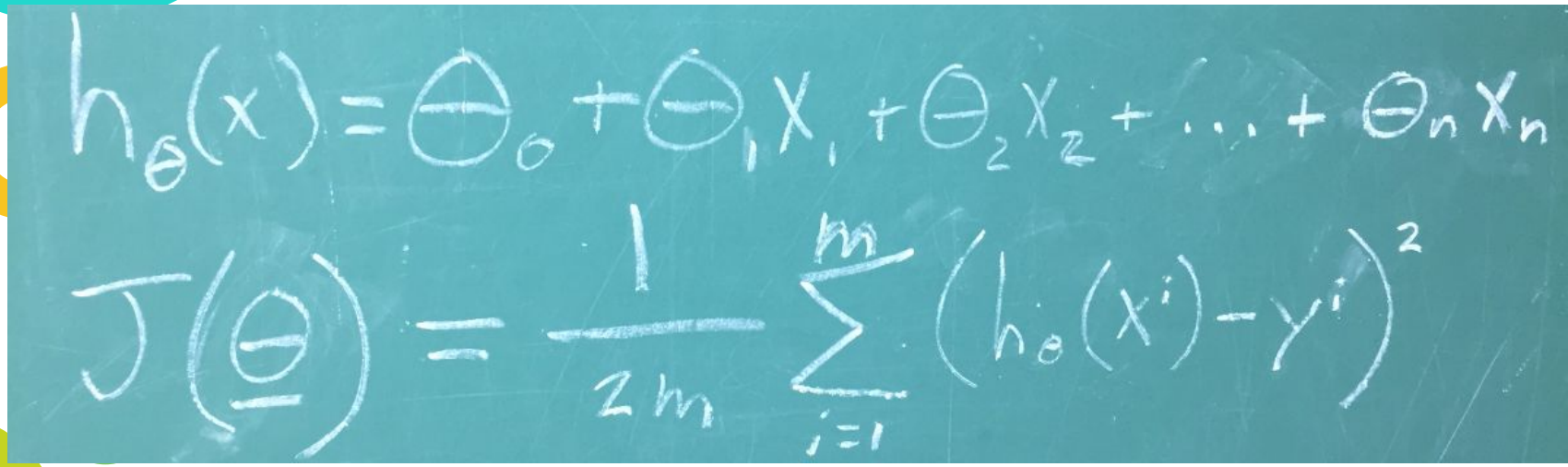
Hypothesis Function



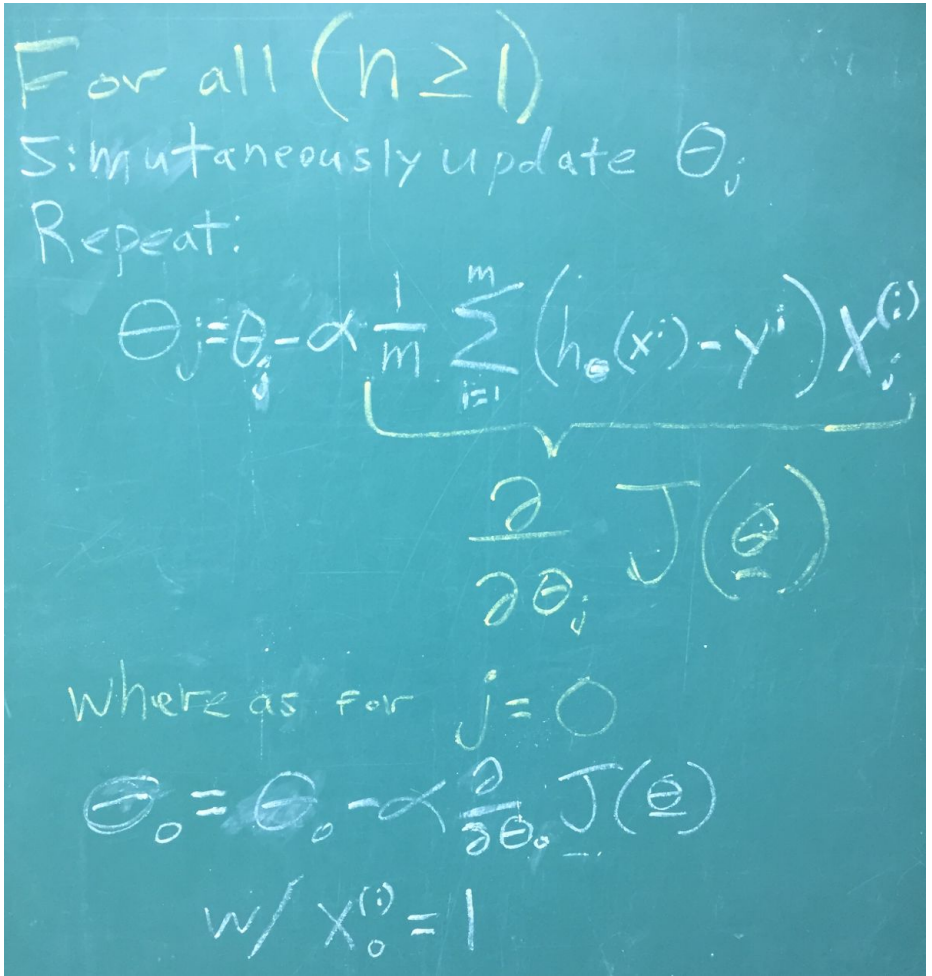
Linear Regression



Gradient Descent



Algorithm Adaptation

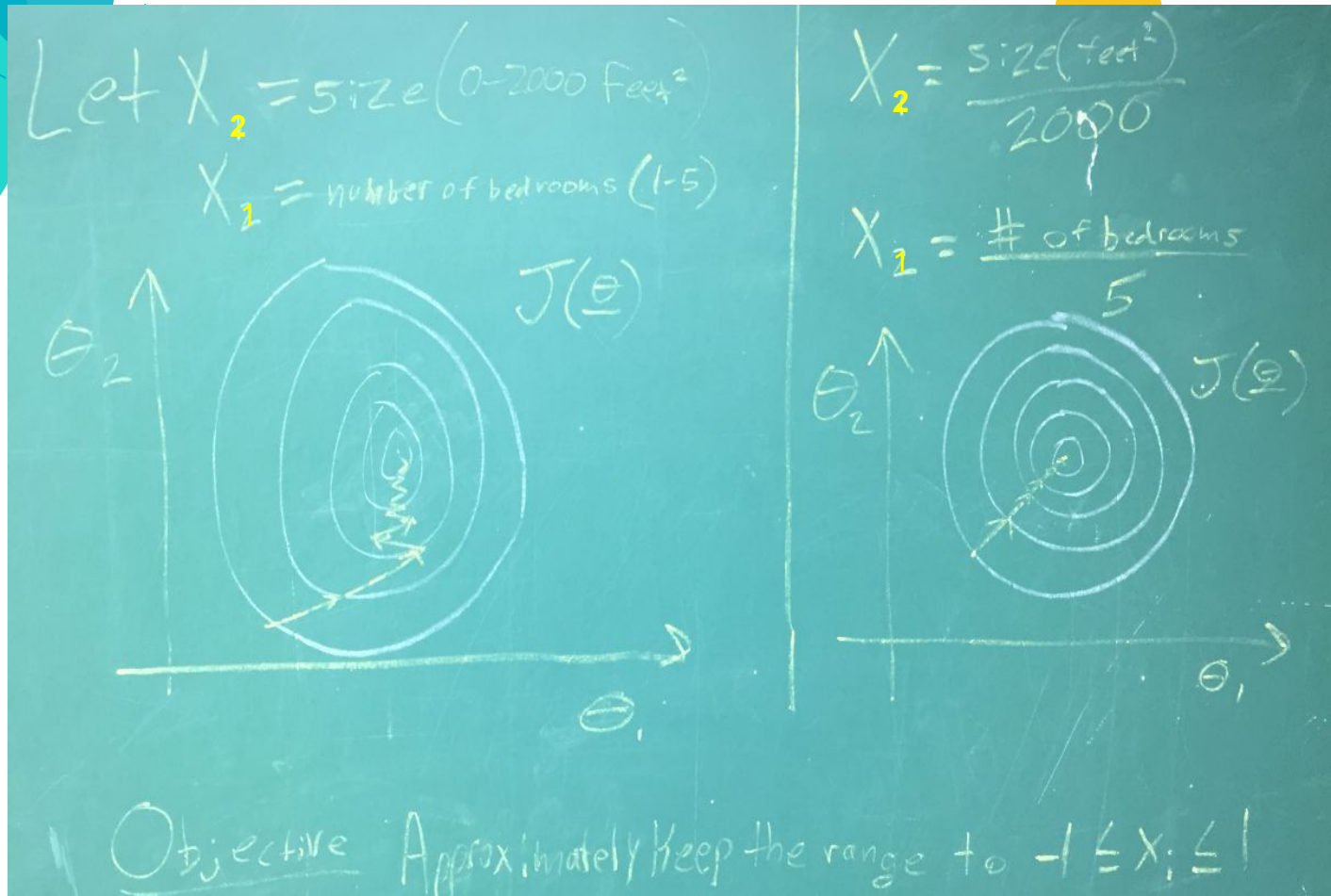


Feature Scaling

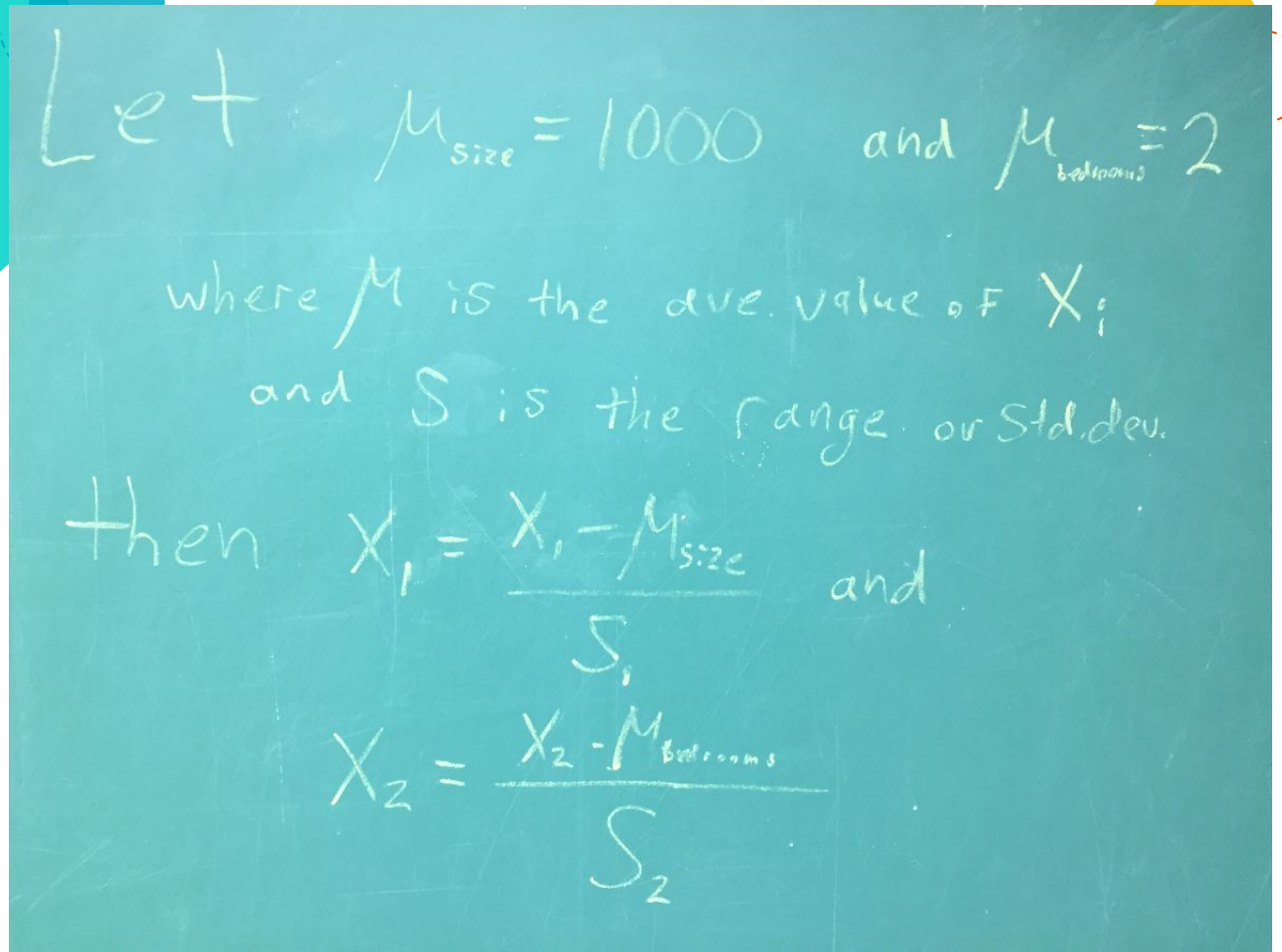
Benefits of proper feature scaling:

- Faster time reaching the global minimum

Feature Scaling

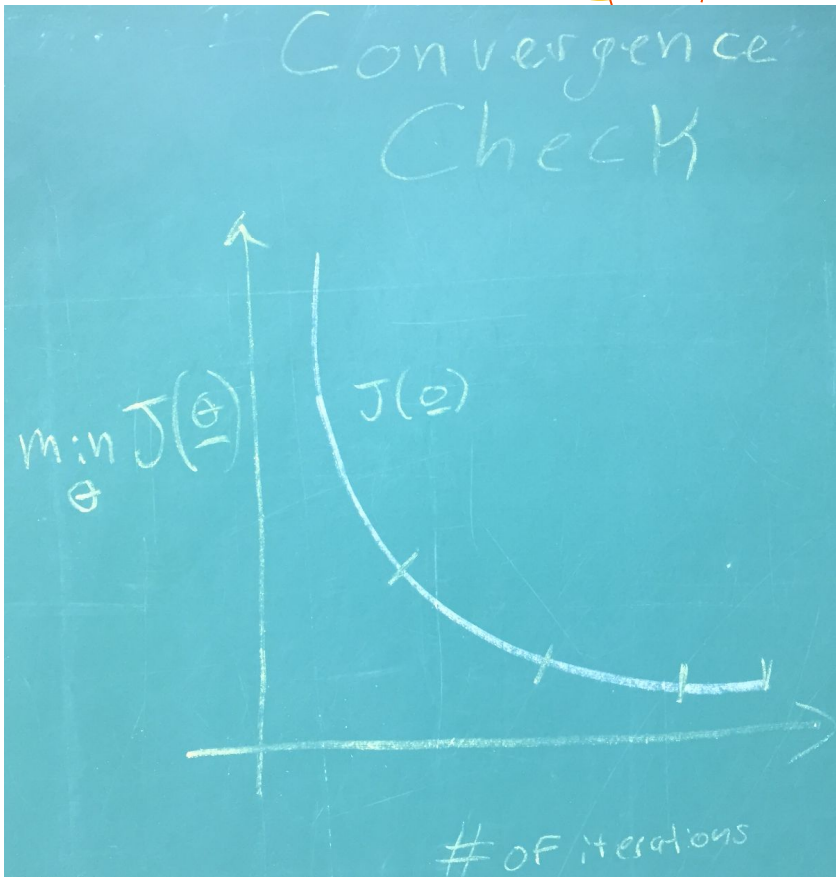


Mean Normalization

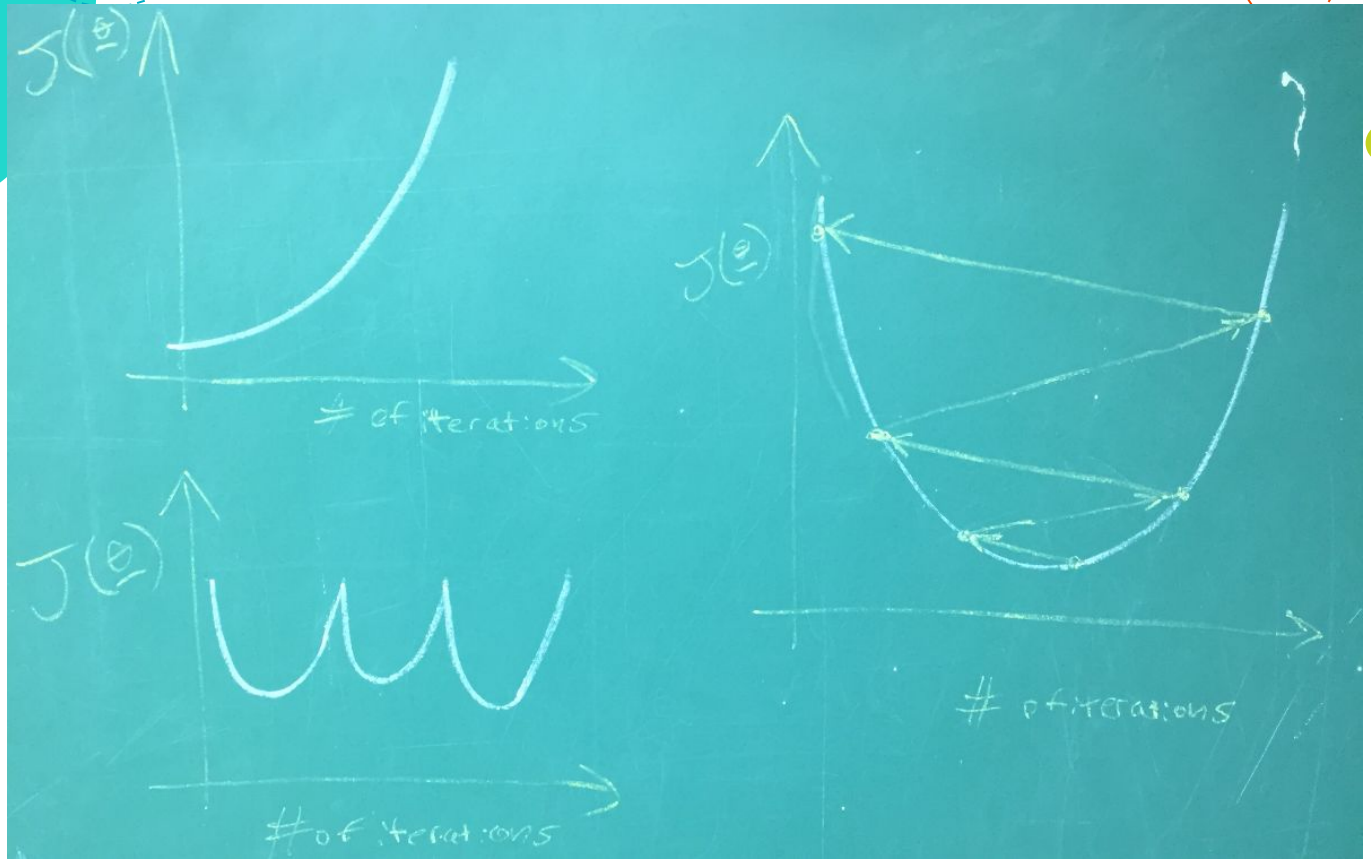


Gradient Descent Checking

- Converges if the cost decreases by less than 10^-3 between 2 iterations



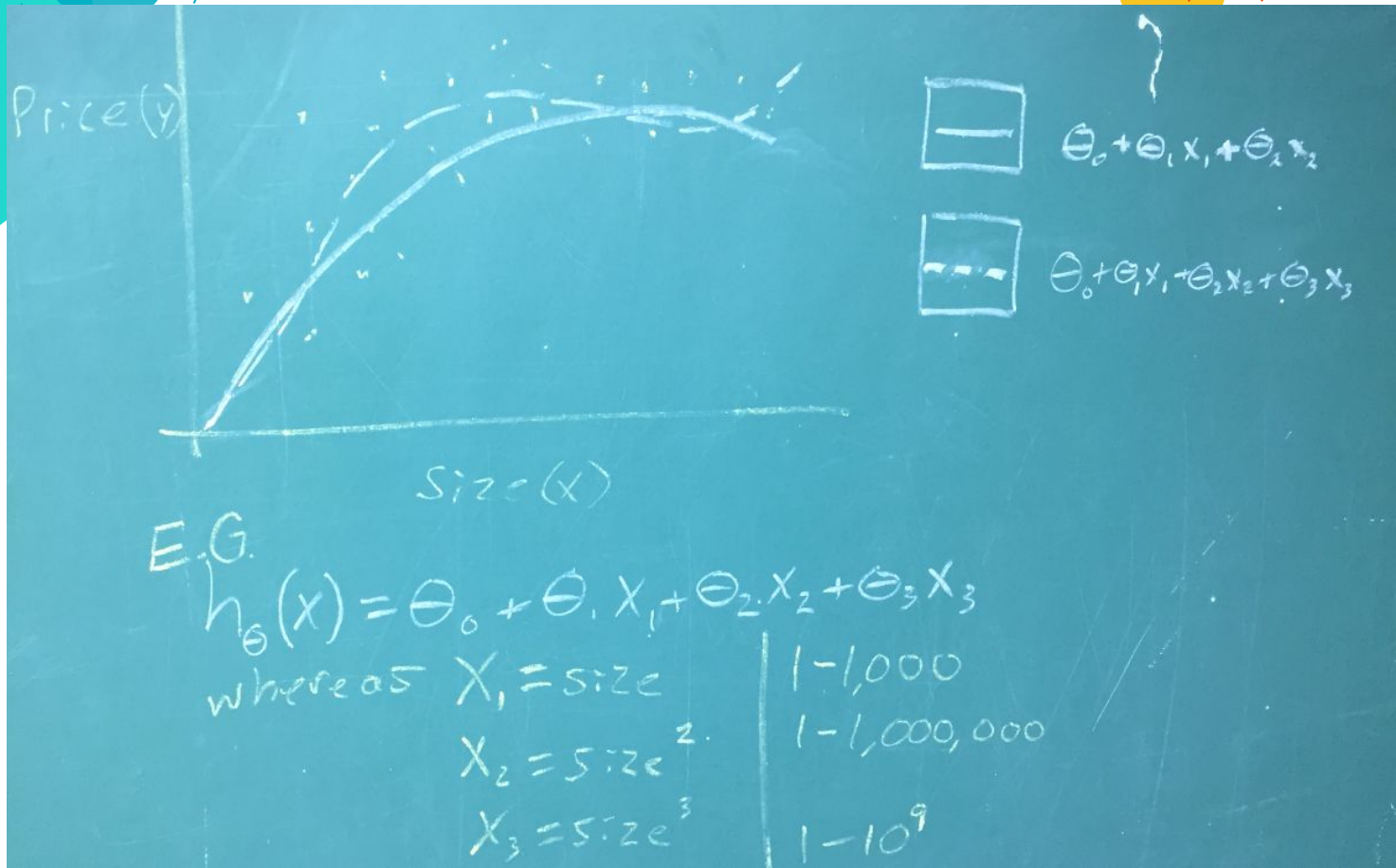
Learning Rate



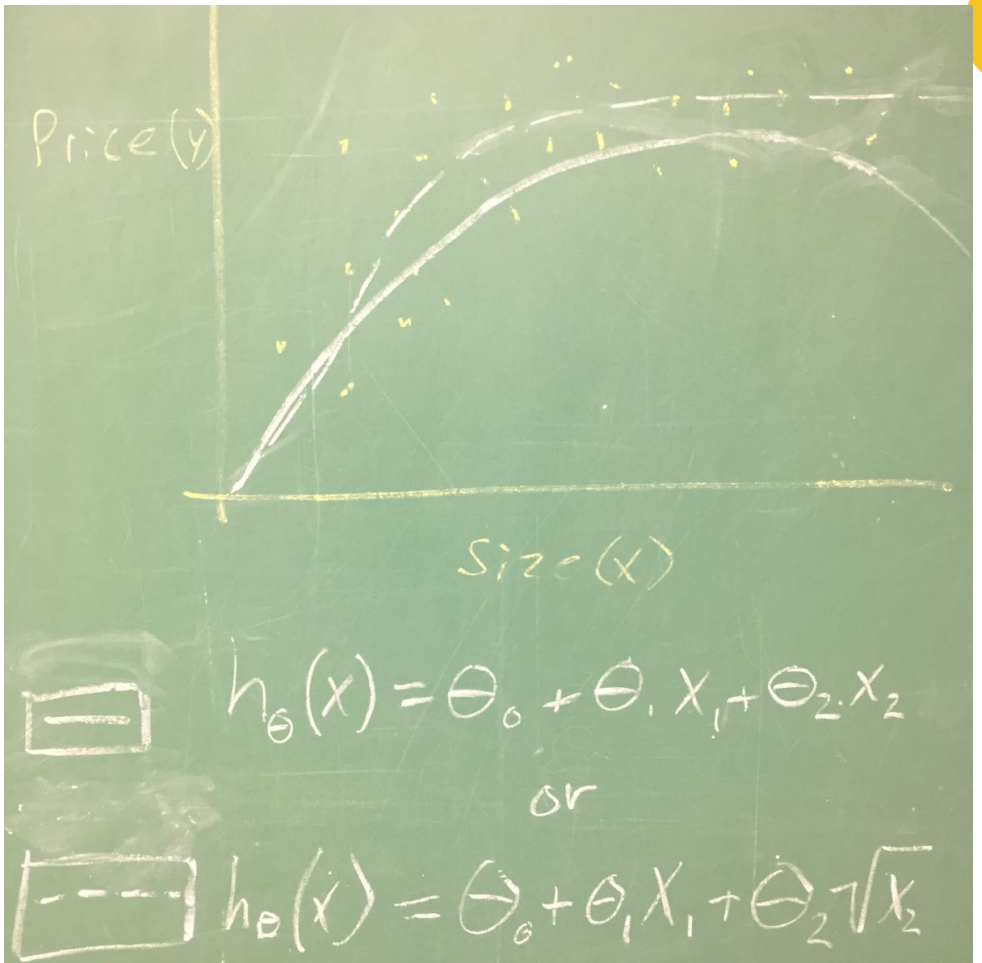
Polynomial Regression

- Pick a function to closely characterize the feature pattern within the data

Polynomial Regression



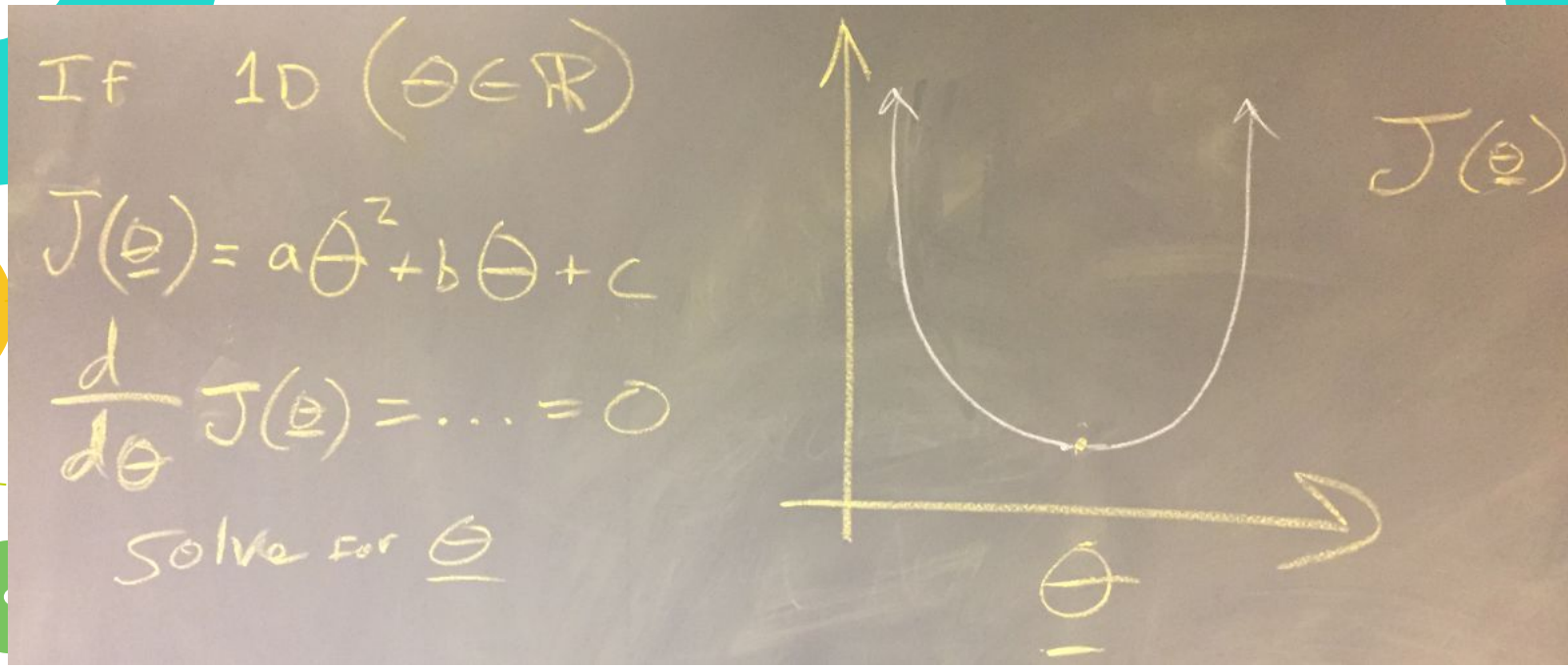
Polynomial Regression



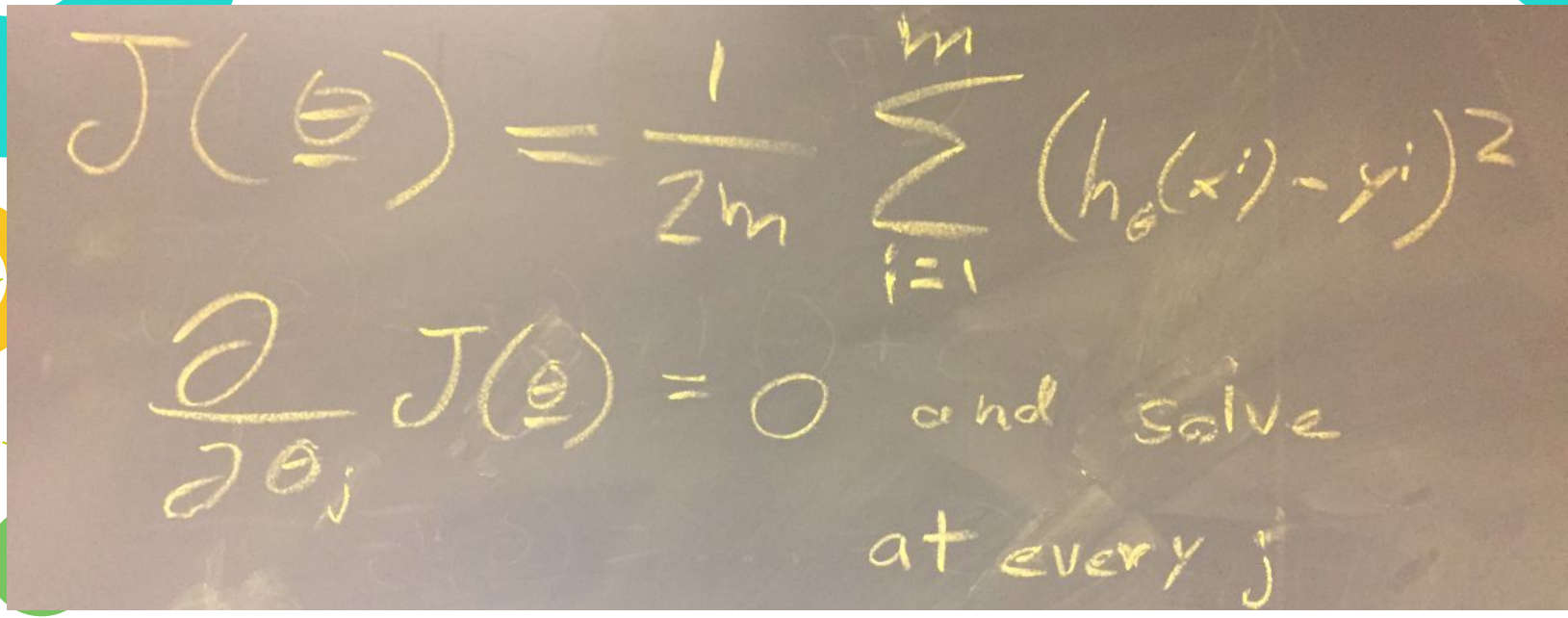
Normal Equation

- Analytically solve for theta using matrices
  - If theta is a real number
    - Set the derivative to 0 and solve
  - Else set the partial derivative to 0 and solve

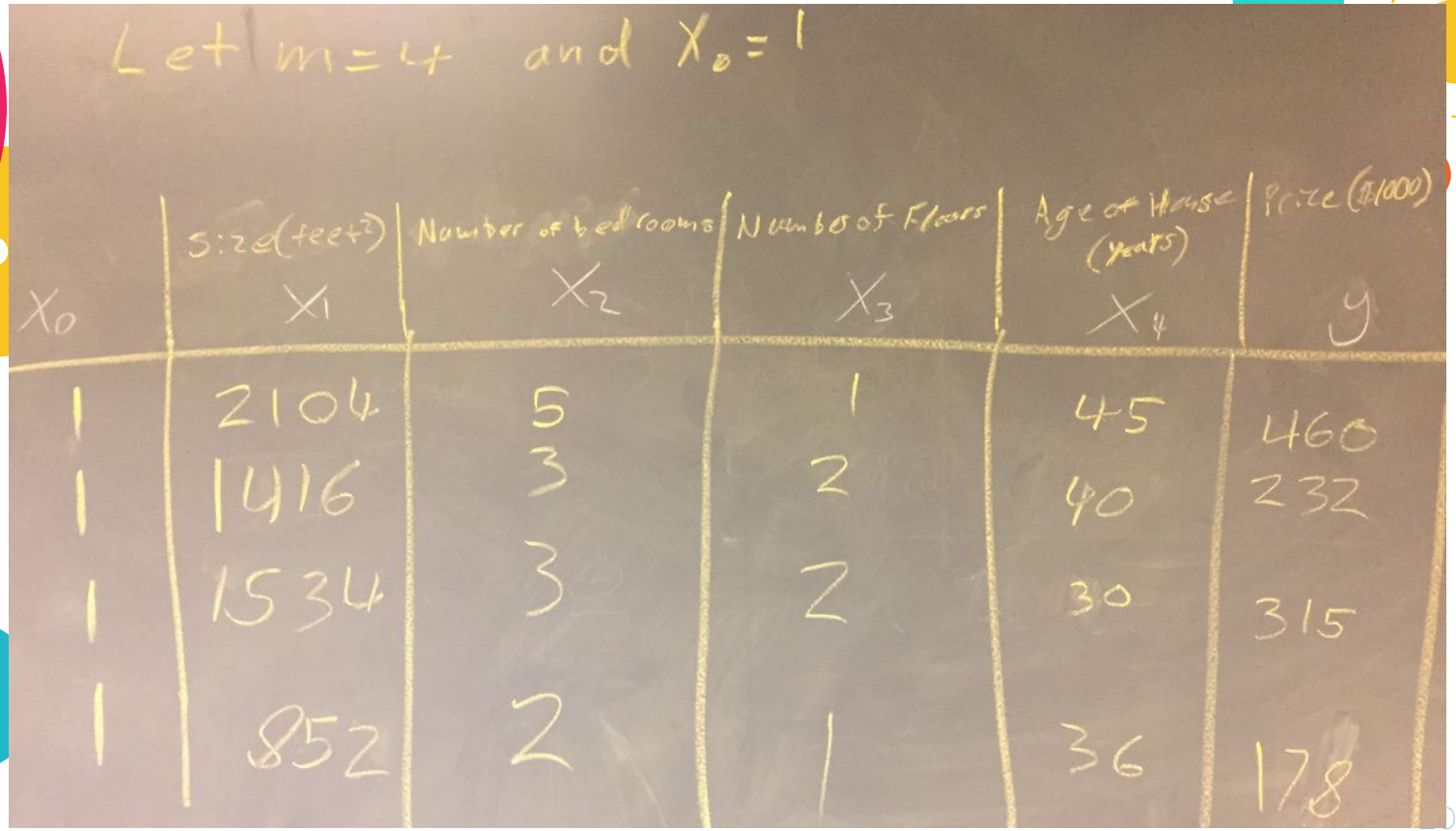
Normal Equation



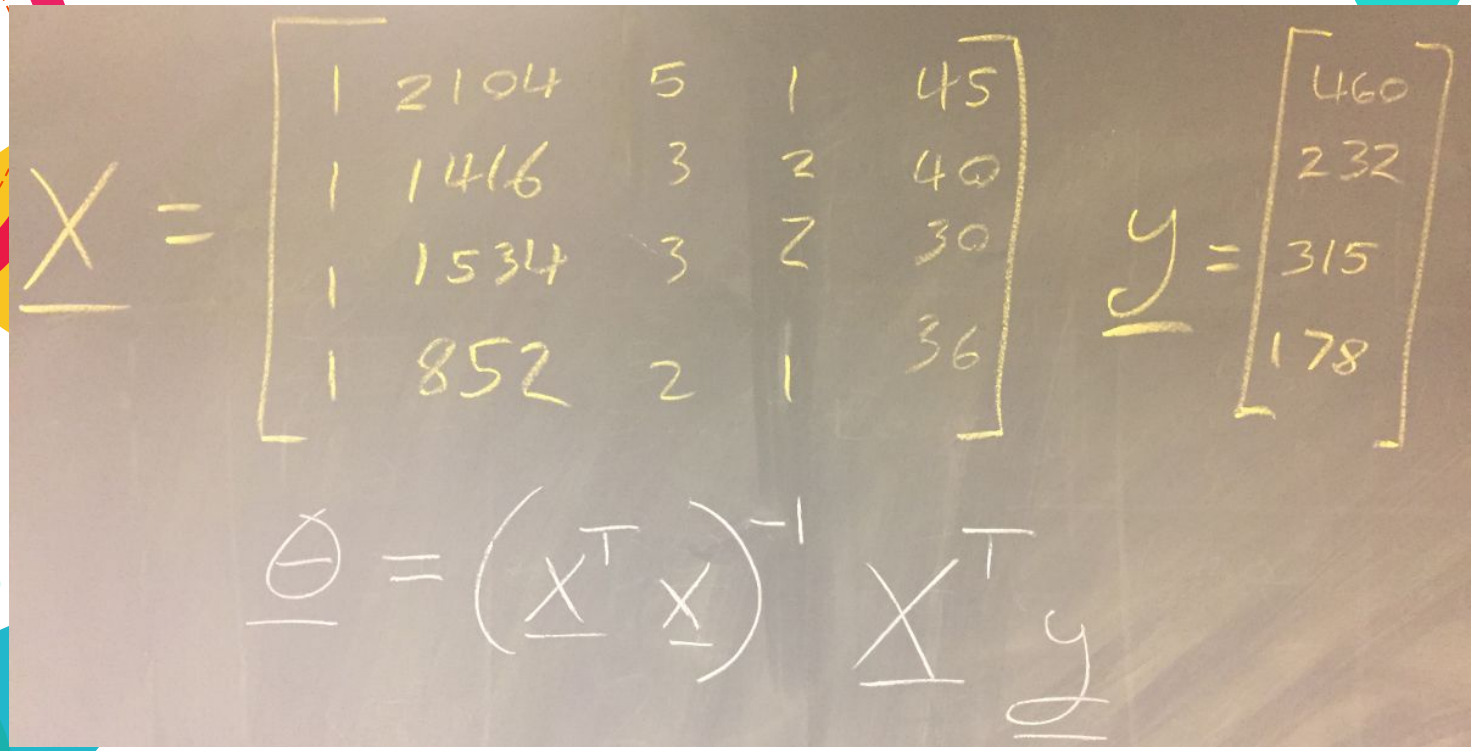
Normal Equation



Example


Let m = 4 and $X_0 = 1$
X_0
Size (feet²) X_1
Number of bedrooms X_2
Number of Floors X_3
Age of House (years) X_4
Price ($1000) y
1 2104 5 1 45 460
1 1416 3 2 40 232
1 1534 3 2 30 315
1 852 2 1 36 178

Example (continued)



Comparisons

Gradient Descent

- Chosen alpha component
- Many Iterations
- Works well with large n (10,000)
- Works well with data of features > 1000

Normal Equations

- No chosen alpha component
- No iterations
- Slow when n is large 1,000
- Works well with the number of features < 1000

Non-Invertibility

If some matrix X times its transpose is invertible the strategy is:

- Delete some features (especially ones that are redundant)
- Use regularization
- Generalized Inverse

Sources

- https://www.ritchieng.com/
- https://towardsdatascience.com/super-simple-machine-learning-by-me-multiple-linear-regression-part-1-447800e8b624
- http://www.dmi.unict.it/farinella/SMM/Lectures/25_Nov2015_2.pdf