Intro to Machine Learning - Week 4

Regression (Part 1)

Linear Regression
Loss Function
Gradient Descent

What will we focus on?

Concepts, Problems

1 hour

Google Colab Project

1 hour

Schedule

Supervised Learning
Classification & Regression, Hypothesis Testing

Classification 1
Conditional Probability, Naive Bayes, Bayesian Learning

Classification 2
Information Gain, Decision Trees, Random Forest, Ensembles

Regression 1
Linear Regression, Loss Function, Gradient Descent

Regression 2
Logistic Regression, Support Vector Machine, Model Tuning

Model Evaluation
Accuracy Metrics, Over- & Underfitting, Cross Validation

Regression

- Target responses are **continuous/ numerical**

Examples of Regression

- Stock prices
- Cost of airplane tickets
- Credit Scores
- Supply Chain Management
- Sports Projections

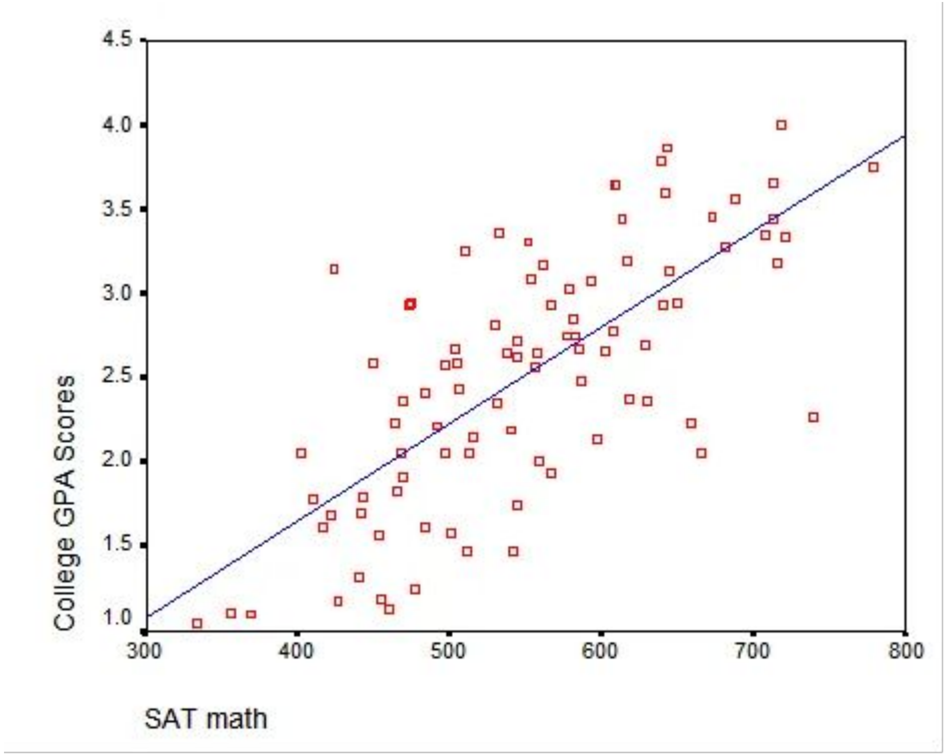
Linear Regression

Features (x_i) & Target response (y) are **continuous in nature**

- Linear regression is used for finding **linear relationship** between the target response and one or more features.
- This has a variety of applications like
  - Historical stock prices → Future stock price
  - Teams Past Performance → Future performance
  - Process, memory used → Power consumption

Example

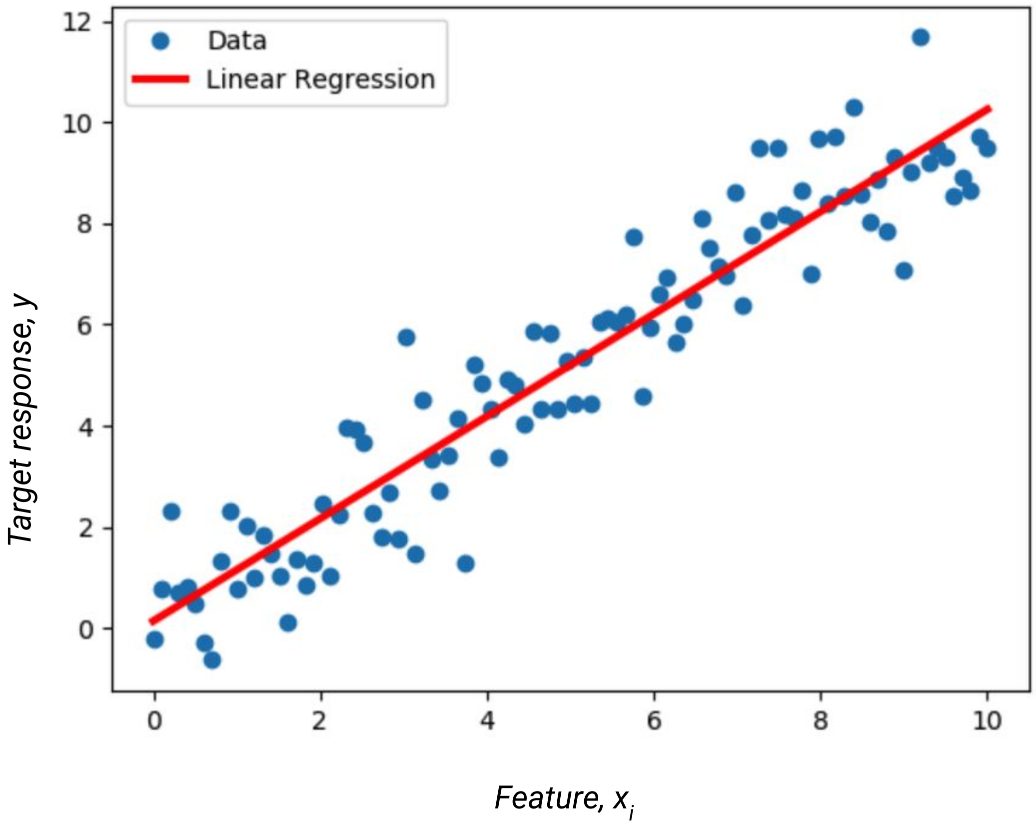
Students SAT math scores and college GPA when they graduate is collected.



https://www.theanalysisfactor.com/linear-regression-outcome-boundaries/

Linear Regression

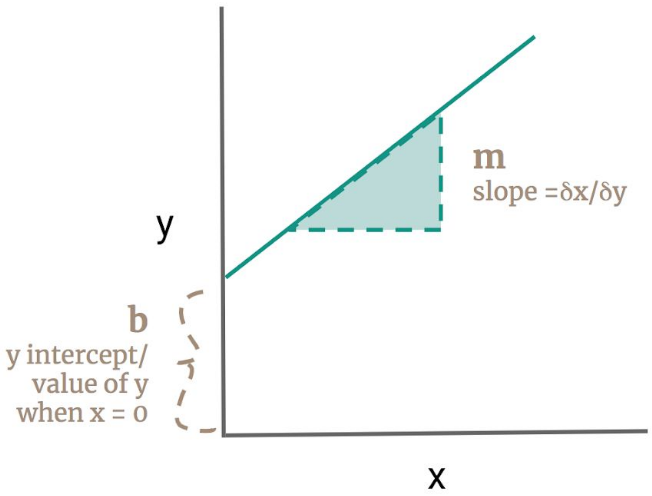
The core idea is to obtain a line that **best fits the data**



https://www.researchgate.net/figure/Linear-Regression-model-sample-illustration_fig3_333457161

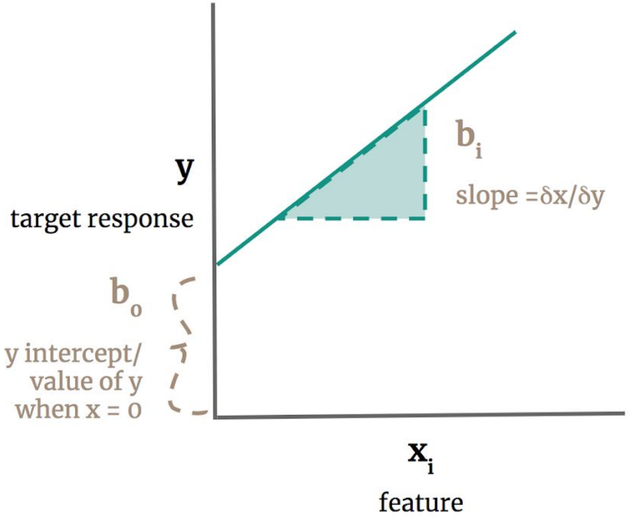
Linear Algebra

WOMEN WHO CODE SILICON VALLEY x WOMEN WHO CODE DATA SCIENCE



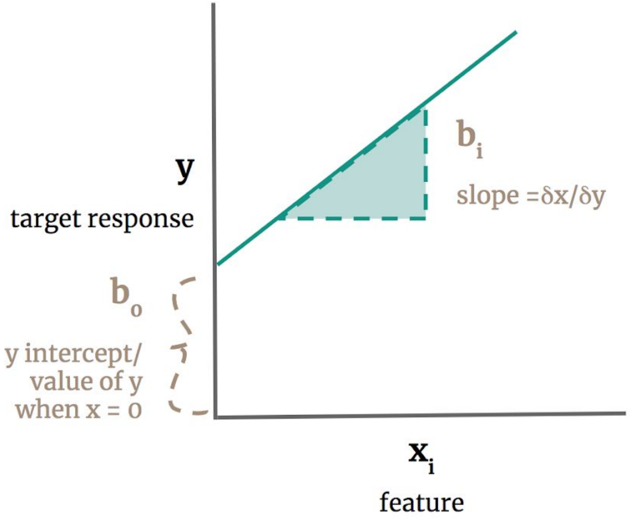
$$\underbrace{y}_{\text{Predictor variable}} = (\underbrace{m}_{\text{Slope of x}} \times \underbrace{x}_{\text{Dependent Variable}}) + \underbrace{b}_{\text{Y intercept}}$$

If x_i is a feature and y is the target



$$\underbrace{y}_{\text{Target response, y}} = (\underbrace{b_i}_{\text{Slope of } x_i} \times \underbrace{x_i}_{\text{Feature } x_i}) + \underbrace{b_o}_{\text{y intercept}}$$

Formula



$$\underbrace{y}_{\text{Target response, y}} = (\underbrace{b_i}_{\text{Slope of } x_i} \times \underbrace{x_i}_{\text{Feature } x_i}) + \underbrace{b_o}_{\text{y intercept}}$$

WOMEN WHO CODE SILICON VALLEY X WOMEN WHO CODE DATA SCIENCE

1-feature Linear Regression

$$\underbrace{y}_{\text{Target response, } y} = (\underbrace{b_i}_{\text{Slope of } x_i} \times \underbrace{x_i}_{\text{Feature } x_i}) + \underbrace{b_0}_{\text{y intercept}}$$

We have to estimate

$$b_i = \frac{\sum[(x - \bar{x}) \times (y - \bar{y})]}{\sum(x - \bar{x})^2}$$

$$b_0 = y - (b_i \times x_i)$$

Solve for y, when we have samples of x_i and y

What if we have > 1 feature?

$$\underbrace{y}_{\text{Target response, } y} = \underbrace{b_1 x_1}_{\text{Feature } x_1} + \underbrace{b_2 x_2}_{\text{Feature } x_2} + \ldots + \underbrace{b_n x_n}_{\text{Feature } x_n} + \underbrace{b_0}_{\text{y intercept}}$$

Linear Regression

Types

Features (x_i) & Target response (y) are continuous in nature

WOMEN WHO CODE SILICON VALLEY x WOMEN WHO CODE DATA SCIENCE

Simple Linear Regression

- There is one feature x_1 and a target response y

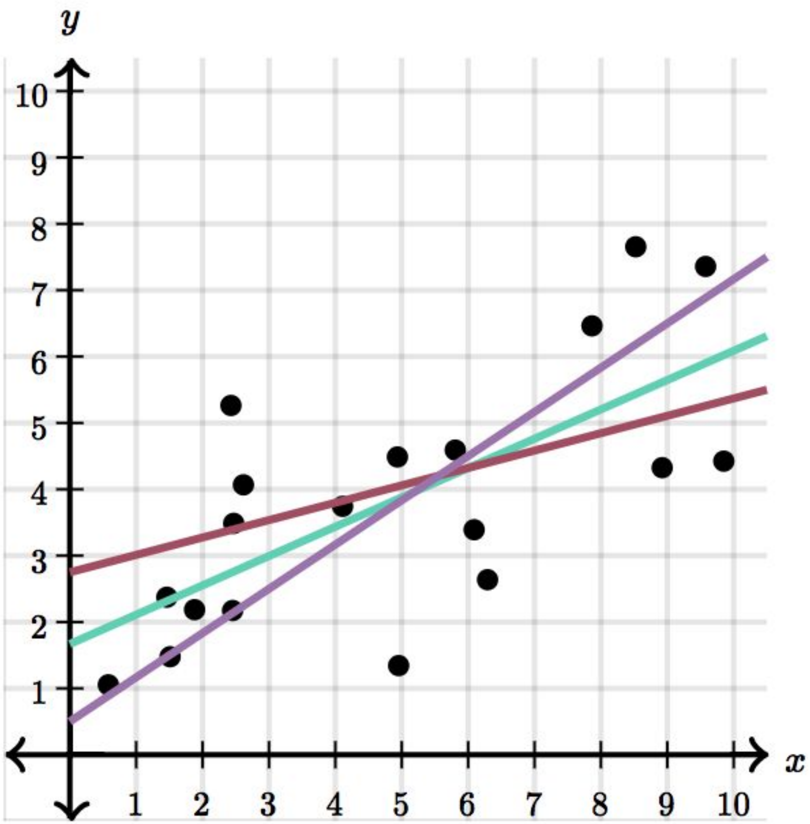
$$\underbrace{y}_{\text{Target response, y}} = \underbrace{b_1 x_1}_{\text{Feature } x_1} + \underbrace{b_0}_{\text{y intercept}}$$

Multiple Linear Regression

- There are n features x_i and a target response y

$$\underbrace{y}_{\text{Target response, y}} = \underbrace{b_1 x_1}_{\text{Feature } x_1} + \underbrace{b_2 x_2}_{\text{Feature } x_2} + \dots + \underbrace{b_n x_n}_{\text{Feature } x_n} + \underbrace{b_0}_{\text{y intercept}}$$

Solve for y



https://www.khanacademy.org/math/statistics-probability/describing-relationships-quantitative-data/regression-library/a/introduction-to-residuals

Loss Function

Features (x_i) & Target response (y) are **continuous in nature**

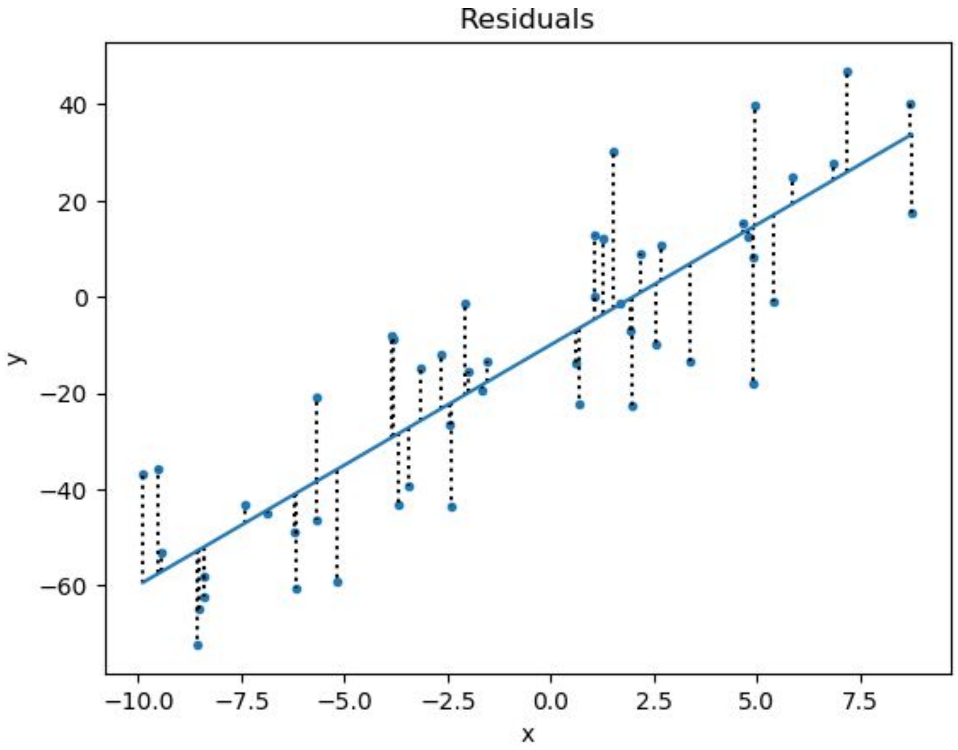
- Measure of how good your prediction model (y) is able to predict the expected outcome.
  - **Deviation** of prediction model from expected outcome
- We convert the learning problem into an **optimization problem**
  - define a **loss function**
  - optimize the algorithm to minimize the loss function.

https://towardsdatascience.com/importance-of-loss-function-in-machine-learning-eddaaec69519

Loss Function

Residuals

- Measure of deviation from the expected value i.e., $f(x_i) - y$



https://blog.bigml.com/2019/03/27/linear-regression-technical-overview/

Loss Function

- **Mean Absolute Error**
- *Mean Squared Error*
- *Root Mean Squared Error*

- MAE measures the average magnitude of the errors in a set of predictions, without considering their direction.

$$\underbrace{\text{MAE}}_{\text{Mean Absolute Error}} = \frac{1}{n} \sum_{j=(1,n)} \left| \underbrace{y_j}_{\text{Actual value}} - \underbrace{\hat{y}_j}_{\text{Predicted value}} \right|$$

- Value ranges from **0 - ∞**, lowest value preferred
- Does not quantify **direction of error**
- Individual differences are **weighted equally**

https://medium.com/human-in-a-machine-world/mae-and-rmse-which-metric-is-better-e60ac3bde13d

Loss Function

- *Mean Absolute Error*
- **Mean Squared Error**
- *Root Mean Squared Error*

- MSE measures the average squared magnitude of the errors in a set of predictions, without considering their direction.

$$\underbrace{MSE}_{\text{Mean Squared Error}} = \frac{1}{n}\sum_{j=(1,n)} (\underbrace{y_j}_{\text{Actual value}} - \underbrace{\hat{y}_j}_{\text{Predicted value}})^2$$

- Value ranges from **0 - ∞**, lowest value preferred
- Does not quantify **direction of error**
- Differences are **not weighted equally**
  - Squared weight to large errors
  - Highly sensitive to outliers/variance

https://medium.com/human-in-a-machine-world/mae-and-rmse-which-metric-is-better-e60ac3bde13d

Loss Function

- *Mean Absolute Error*
- *Mean Squared Error*
- **Root Mean Squared Error**

WOMEN WHO CODE SILICON VALLEY x WOMEN WHO CODE DATA SCIENCE

- RMSE is a quadratic scoring rule that also measures the **average magnitude of the error**.

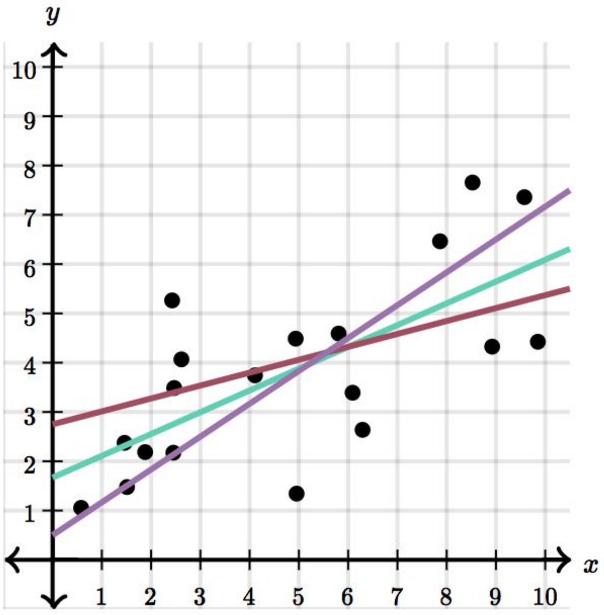
$$\underbrace{\text{RMSE}}_{\text{Root Mean Squared Error}} = \sqrt{\frac{1}{n}\sum_{j=(1,n)}(\underbrace{y_j}_{\text{Actual value}} - \underbrace{\hat{y}_j}_{\text{Predicted value}})^2}$$

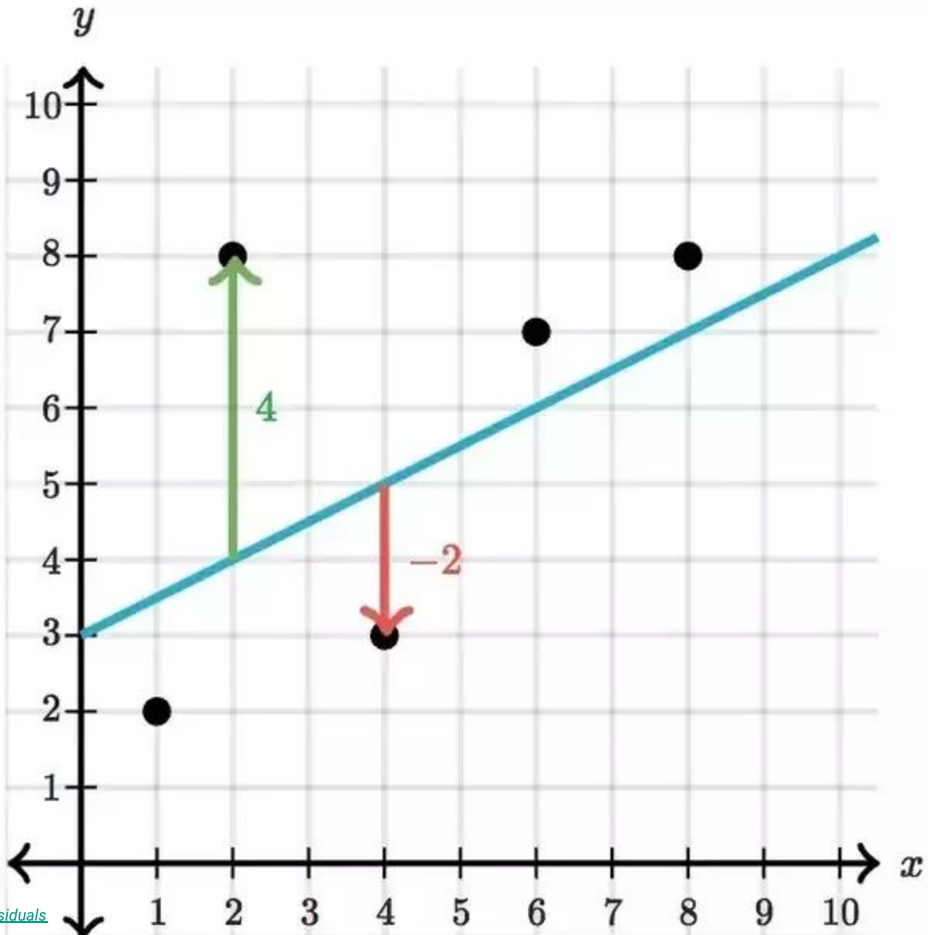
- Value ranges from **0 - ∞**, lowest value preferred
- Does not quantify **direction of error**
- Differences are **not weighted equally**
    - Relatively high weight to large errors
    - Sensitive to outliers/variance

https://medium.com/human-in-a-machine-world/mae-and-rmse-which-metric-is-better-e60ac3bde13d

Calculate Residuals, Loss

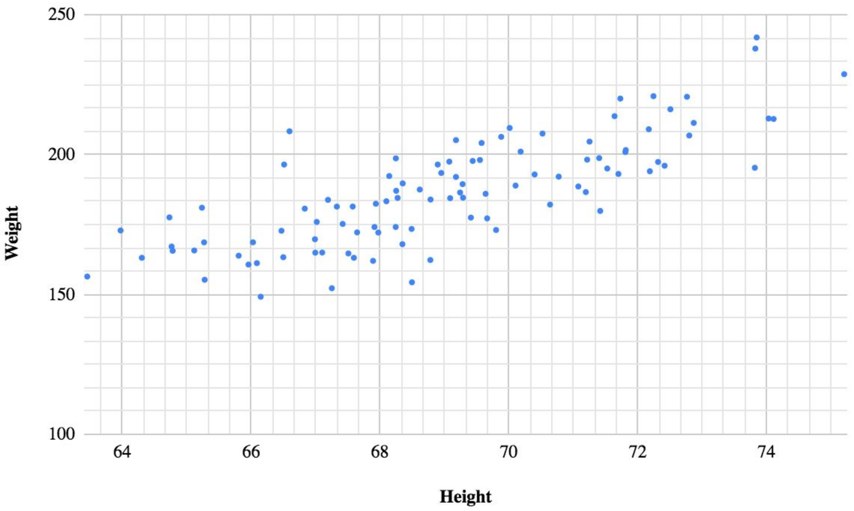
Goal - Minimize the loss (Least Squares)





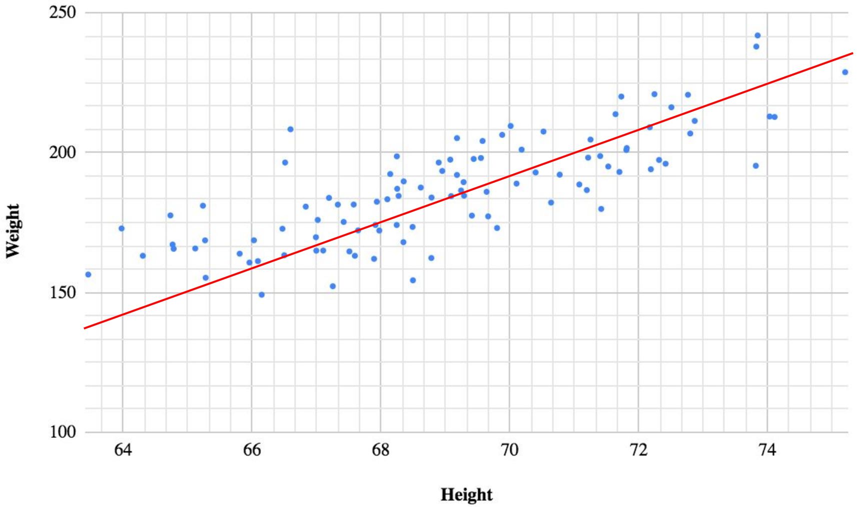
https://www.quora.com/What-does-a-positive-residual-mean-in-statistics
https://www.khanacademy.org/math/statistics-probability/describing-relationships-quantitative-data/regression-library/a/introduction-to-residuals

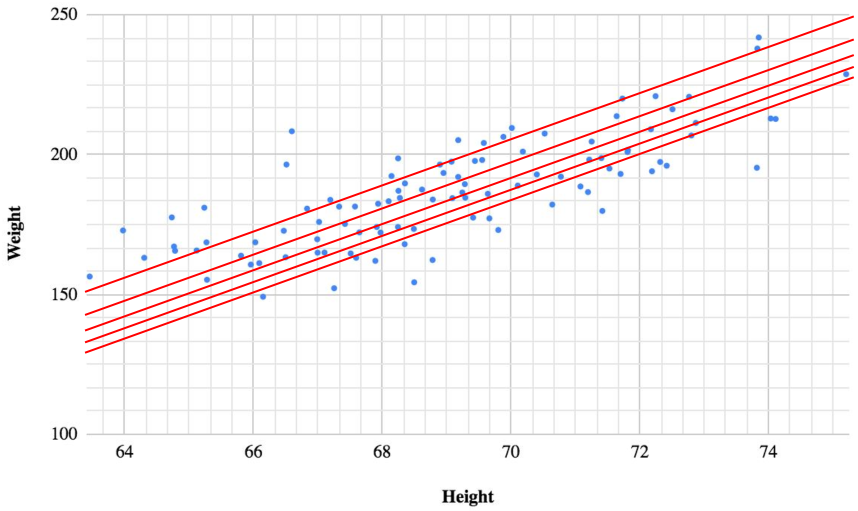
Consider a dataset of 100 people



$$y = \underbrace{b_1}_{\text{Slope}} x_1 + \underbrace{b_0}_{\text{y intercept}}$$

$$y = \underbrace{5.85}_{\text{Estimated Slope with Least Squares}} x_1 + \underbrace{b_0}_{\text{Estimate with Gradient Descent}}$$

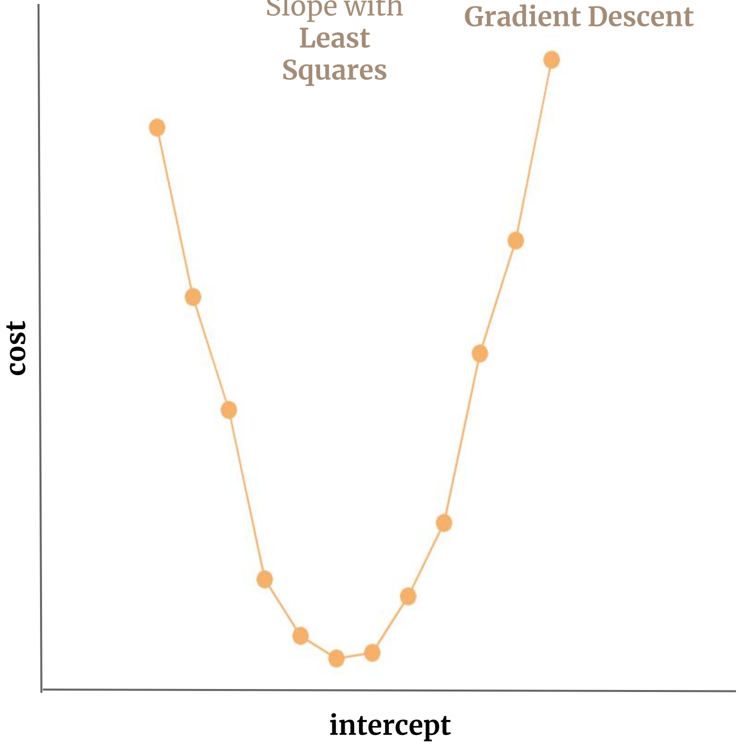




WOMEN WHO CODE SILICON VALLEY X WOMEN WHO CODE DATA SCIENCE

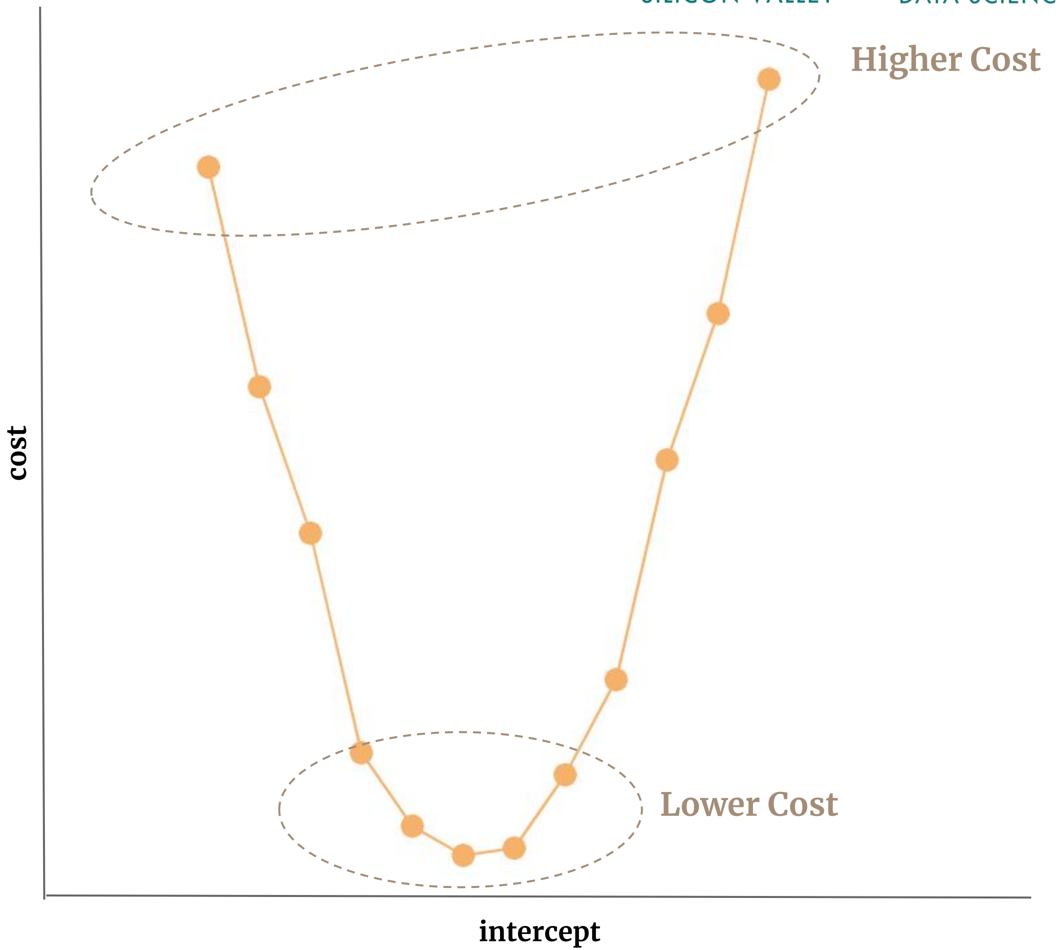
$$y = \underbrace{5.85x_1}_{\text{Estimated Slope with Least Squares}} + \underbrace{b_o}_{\text{Estimate with Gradient Descent}}$$

If we were to test for **all possible values of b_o**, the number of calculations for this can add up!

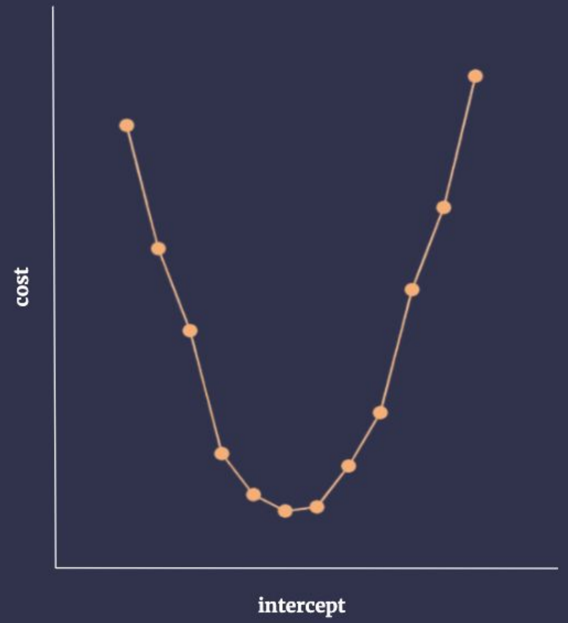


Optimization Function

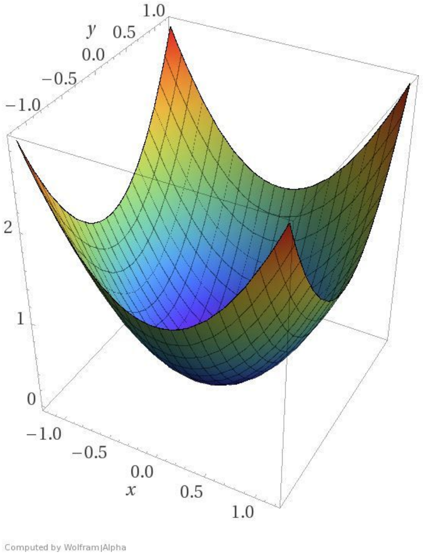
Calculations tend to be infinite if all possible values of b_0 is tested!

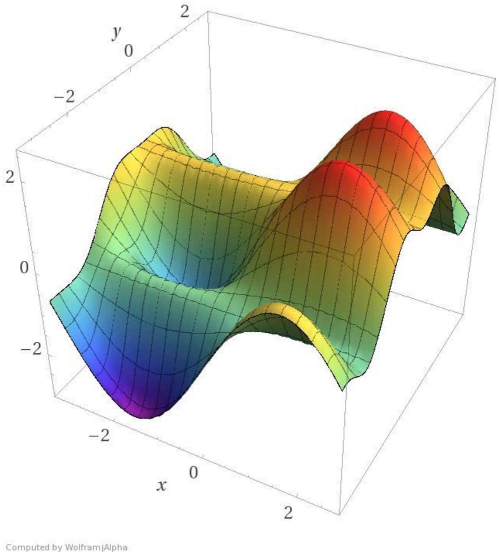


Optimization Function



- In the case of loss function, an optimization problem consists of minimizing the loss by systematically choosing input values from within an allowed set and computing the loss of the function.





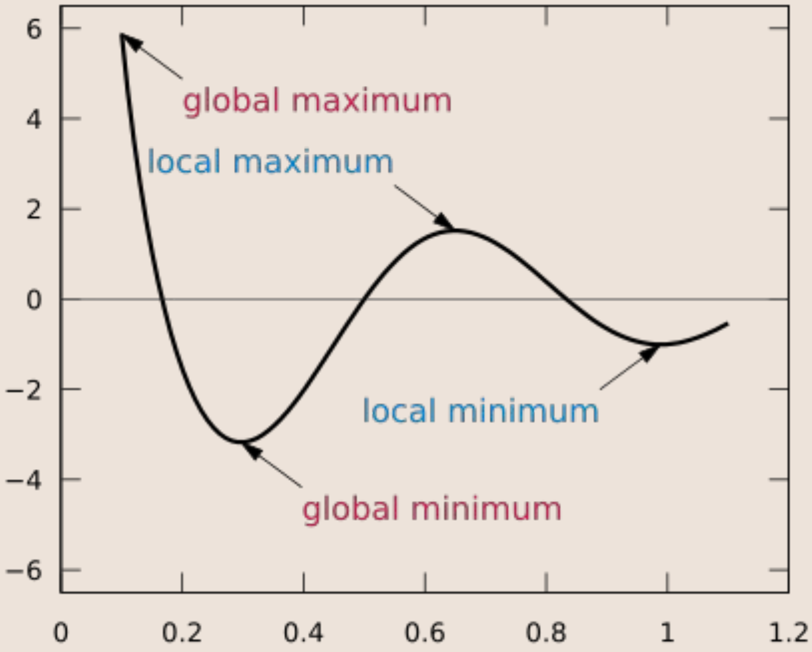
https://blog.paperspace.com/intro-to-optimization-in-deep-learning-gradient-descent/

Terminology

WOMEN WHO CODE SILICON VALLEY x WOMEN WHO CODE DATA SCIENCE

Extrema

- The **largest/smallest** value that can be **produced by a function F**
  - **Local**: within a given range
  - **Global**: on the entire domain/data
- Types
  - **Maxima** The largest value that can be produced by a function F
  - **Minima** The smallest value that can be produced by a function F



https://en.wikipedia.org/wiki/Maxima_and_minima

Gradient Descent

Proposed by **Augustin-Louis Cauchy** in 1847

- Iterative optimization algorithm for **finding a local minimum of a differentiable function**.
- Gradient descent is also known as **steepest descent.**
- Gradient descent is based on the observation that
  - if the multivariable function **F(x)** is defined and **differentiable** in a neighborhood of a point **a**,
  - then **F(x) decreases fastest** if one goes from **a** in the **direction of the negative gradient of F**.

https://en.wikipedia.org/wiki/Maxima_and_minima

Gradient Descent

Analogy

Person = algorithm

Mountain Base = global minima

Path down = sequence of parameters the algorithm will explore

Steepness = Slope of the Error Surface

Instrument = Differentiation

Time taken = Learning Rate

- A person is stuck in the mountains and is trying to get down
- There is heavy fog such that visibility is extremely low
- The path down is not visible and we have to use gradient descent along with local information to find this
  - Find steepness of hill at current position
  - Find the direction of steepest descent
- Assume also that the steepness of the hill is not immediately obvious with simple observation
  - Requires a sophisticated instrument
  - Minimize the usage to reach the bottom by sunset

https://en.wikipedia.org/wiki/Maxima_and_minima

Gradient Descent

Types

- **Batch**
- *Stochastic*
- *Mini Batch*

- **It is an iterative** method for **optimizing an objective function**
- **Whole dataset** is used in each iteration to calculate the function
- **Some features of it are**
  - Brute-force
  - produces a stable learning path
  - takes longer time when the training set is large

https://towardsdatascience.com/an-introduction-to-gradient-descent-c9cca5739307

Gradient Descent

Types

- *Batch*
- **Stochastic**
- *Mini Batch*

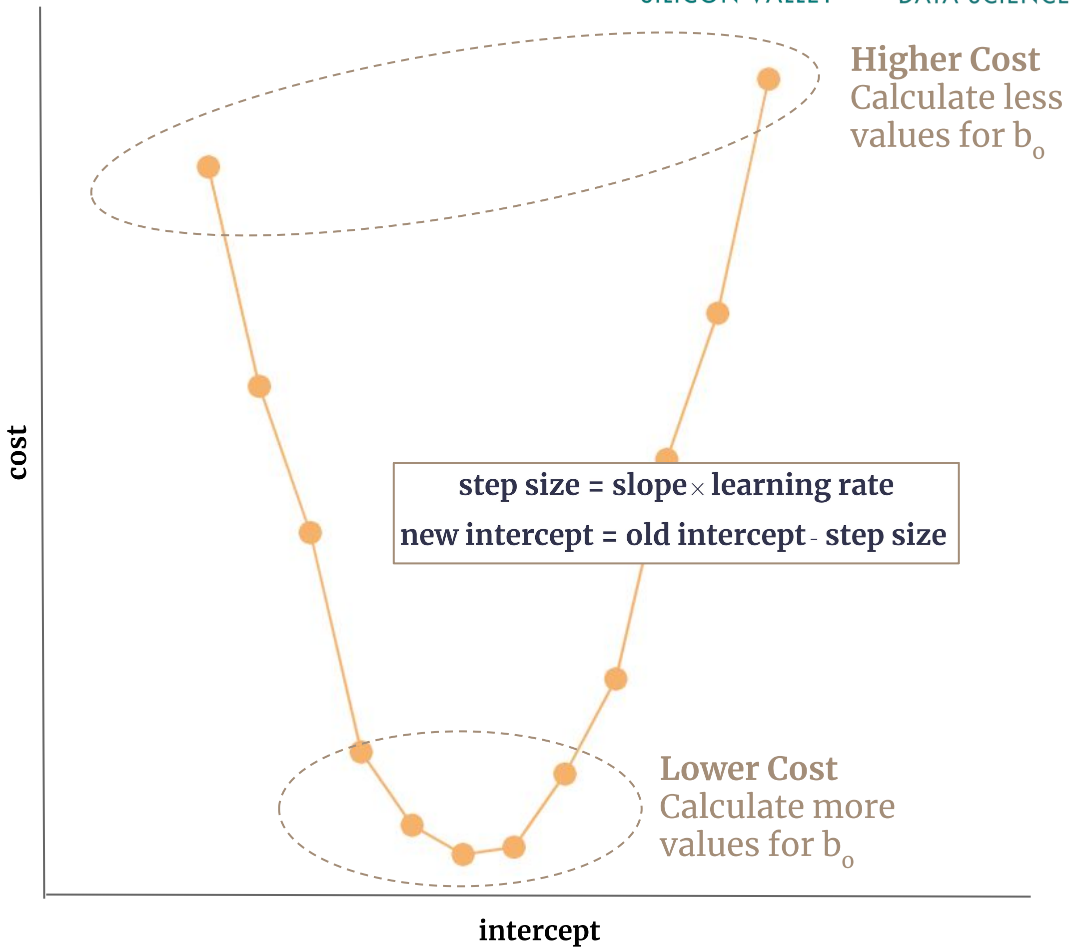
- **Stochastic**: randomly determined; having a random probability distribution or pattern that may be analyzed statistically but may not be predicted precisely.
- It is a stochastic **approximation** of gradient descent optimization
  - it replaces the actual gradient (calculated from the entire data set) by an **estimate** of it (calculated from a **random subset** of the data).
  - This is especially valuable in big data applications as it r**educes the computational burden** with lower convergence rate.

https://en.wikipedia.org/wiki/Stochastic_gradient_descent

Gradient Descent

Types

- *Batch*
- **Stochastic**
- *Mini Batch*



Gradient Descent

Types

- *Batch*
- *Stochastic*
- **Mini Batch**

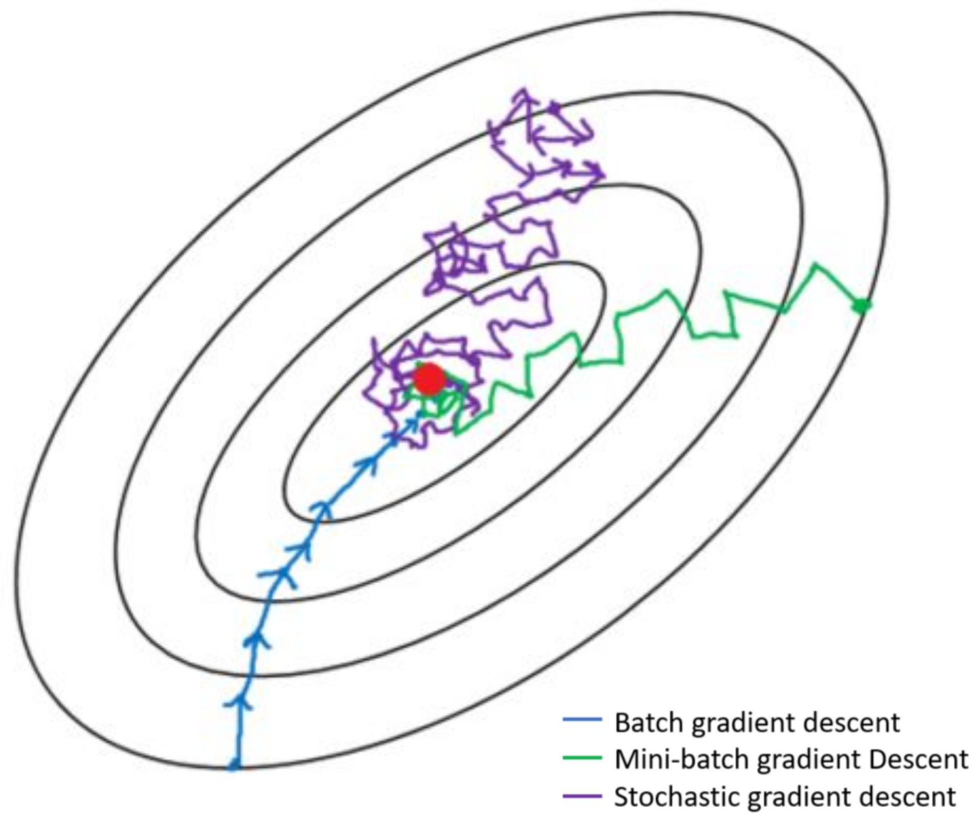
- Combines the concepts of Batch and Stochastic Gradient Descent
- At each step, the algorithm computes gradient based on a **subset of training data** instead of
    - full data set (Batch)
    - only one record (Stochastic)
- Cost function decreases more smoothly
- More stable than Stochastic

https://towardsdatascience.com/an-introduction-to-gradient-descent-c9cca5739307

Gradient Descent

Types

- Batch
- Stochastic
- Mini Batch



https://towardsdatascience.com/an-introduction-to-gradient-descent-c9cca5739307

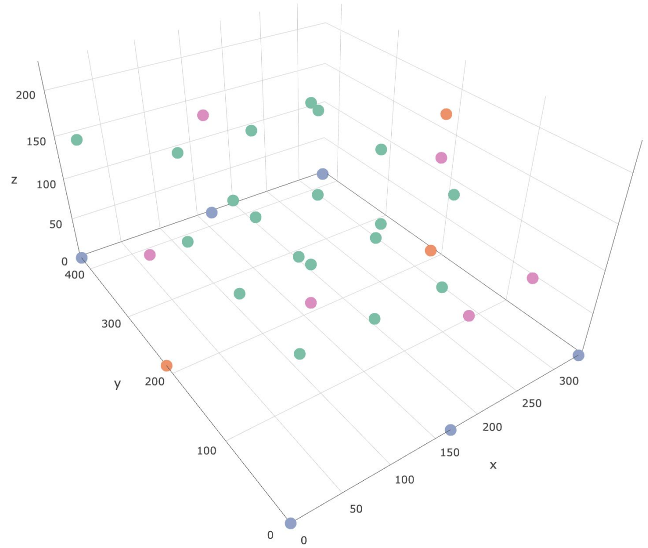
WOMEN WHO CODE SILICON VALLEY x WOMEN WHO CODE DATA SCIENCE

Sparse Learning

What happens if the plotted data points are really sparse and has a high loss for the best result?

In many modern applications, the number of predictors can be extremely large.

- Eg. computational genomics, where the gene data points may correspond to measurements from thousands of genes



https://cims.nyu.edu/~cfgranda/pages/OBDA_spring16/material/sparse_regression.pdf
https://stackoverflow.com/questions/54726853/how-to-interpolate-data-between-sparse-points-to-make-a-contour-plot-in-r-plot

Sparse Learning

- **Feature Subsetting**
- *Regularization*
- *Dimension Reduction*

- Select a small subset of relevant predictors from a training set and then evaluate least-square fit of all possible s sparse models
- Brute-force = Computationally heavy
- If n is the number of features, and s is the number of input features being modeled
  - There are more than 10^{13} possible models when n = 100, s = 10

https://cims.nyu.edu/~cfgranda/pages/OBDA_spring16/material/sparse_regression.pdf

Sparse Learning

- *Feature Subsetting*
- **Regularization**
- *Dimension Reduction*

- If there is noise in the training data, then the estimated coefficients won't generalize well to the future data.
- This method that assigns "weights" for each feature and regularizes/shrinks the estimates for each feature towards zero
- In other words, this technique discourages learning a more complex or flexible model, so as to avoid the risk of overfitting.
- The fitting/shrinking procedure involves a loss function, known as residual sum of squares or RSS

$$\mathrm{RSS} = \sum_{i=1}^{n} \left(y_i - \beta_0 - \sum_{j=1}^{p} \beta_j x_{ij} \right)^2 .$$

https://medium.com/human-in-a-machine-world/mae-and-rmse-which-metric-is-better-e60ac3bde13d

Sparse Learning

- *Feature Subsetting*
- **Regularization**
  - **Lasso, L1**
  - *Ridge, L2*
- *Dimension Reduction*

- LASSO - Least Absolute Shrinkage and Selection Operator
- It adds "absolute value of magnitude" of coefficient as penalty term to the loss function.

$$\sum_{i=1}^{n}(Y_i - \sum_{j=1}^{p} X_{ij}\beta_j)^2 + \lambda \sum_{j=1}^{p} |\beta_j|$$

https://towardsdatascience.com/l1-and-l2-regularization-methods-ce25e7fc831c

Sparse Learning

- *Feature Subsetting*
- **Regularization**
  - *Lasso, L1*
  - **Ridge, L2**
- *Dimension Reduction*

- Ridge regression adds “squared magnitude” of coefficient as penalty term to the loss function.

$$\sum_{i=1}^{n}\left(y_i - \sum_{j=1}^{p} x_{ij}\beta_j\right)^2 + \boxed{\lambda \sum_{j=1}^{p} \beta_j^2}$$

https://towardsdatascience.com/l1-and-l2-regularization-methods-ce25e7fc831c

Sparse Learning

- *Feature Subsetting*
- **Regularization**
- *Dimension Reduction*

Understanding Lambda

- If your **lambda value is too high**, your model will be simple, but you run the **risk of underfitting** your data.
  - Your model won't learn enough about the training data to make useful predictions.
- If your **lambda value is too low**, your model will be more complex, and you run the **risk of overfitting** your data.
  - Your model will learn too much about the particularities of the training data, and won't be able to generalize to new data.
- From there you can see that as λ becomes larger, the variance decreases, and the bias increases. This poses the question: **how much bias are we willing to accept in order to decrease the variance?**

https://towardsdatascience.com/l1-and-l2-regularization-methods-ce25e7fc831c
https://developers.google.com/machine-learning/crash-course/regularization-for-simplicity/lambda

Sparse Learning

- *Feature Subsetting*
- *Regularization*
- **Dimension Reduction**

- Dimensionality reduction is simply, the process of reducing the dimension of your feature set.
- Your feature set could be a dataset with a hundred columns (i.e features) or it could be an array of points that make up a large sphere in the three-dimensional space.
    - Dimensionality reduction is bringing the number of columns down to say, twenty or converting the sphere to a circle in the two-dimensional space.
- This can be done using linear transformations including some other methods.

https://medium.com/human-in-a-machine-world/mae-and-rmse-which-metric-is-better-e60ac3bde13d

Sparse Learning

- *Feature Subsetting*
- *Regularization*
- **Dimension Reduction**

Some dimensionality reduction methods are ones that apply linear transformations like

- **Principal Component Analysis (PCA)**: PCA rotates and projects data along the direction of increasing variance. The features with max variance are the principal components.
- **Factor Analysis**: The values of observed data are expressed as functions of a number of possible causes in order to find which are the most important.
- **LDA (Linear Discriminant Analysis):** projects data in a way that the class separability is maximised.

https://medium.com/human-in-a-machine-world/mae-and-rmse-which-metric-is-better-e60ac3bde13d

Theory Recap

WOMEN WHO CODE SILICON VALLEY x WOMEN WHO CODE DATA SCIENCE

- **Linear Regression**
  - Single, Multiple
- **Residuals**
- **Loss Functions**
  - MAE, MSE, RMSE
- **Gradient Descent**
  - Optimization Function
  - Stochastic GD
- **Sparse Learning**
  - Feature Subsetting
  - Regularization
  - Dimensionality Reduction

WOMEN WHO CODE SILICON VALLEY x WOMEN WHO CODE DATA SCIENCE

Google Colab Project

Homework #1

Highly recommend **Artificial Intelligence - All in One** channel on Youtube

https://www.youtube.com/channel/UC5zx8Owijmv-bbhAK6Z9apg

Here are some useful videos to understand these concepts we learnt today by Andrew Ng

- Linear Regression with One Variable
- Linear Regression with Multiple Variables
- Loss/Cost Function
- Gradient Descent
- Regularization
- Dimensionality Reduction

Homework #2

Try to solve an end-to-end project on the New York Taxi Fare Prediction dataset.

You will need to download the data and load it into Google Colab. You can follow along with this video to know the steps
https://www.youtube.com/watch?v=mNTqIw-Oy44&t=1s

See you next week!

WOMEN WHO CODE SILICON VALLEY x WOMEN WHO CODE DATA SCIENCE

Questions?

Join us on slack (bit.ly/wwcodedatascience-slack) and post it on our **#help-me** channel.

Register?

Register for all sessions at linktr.ee/wwcodedatascience_registration