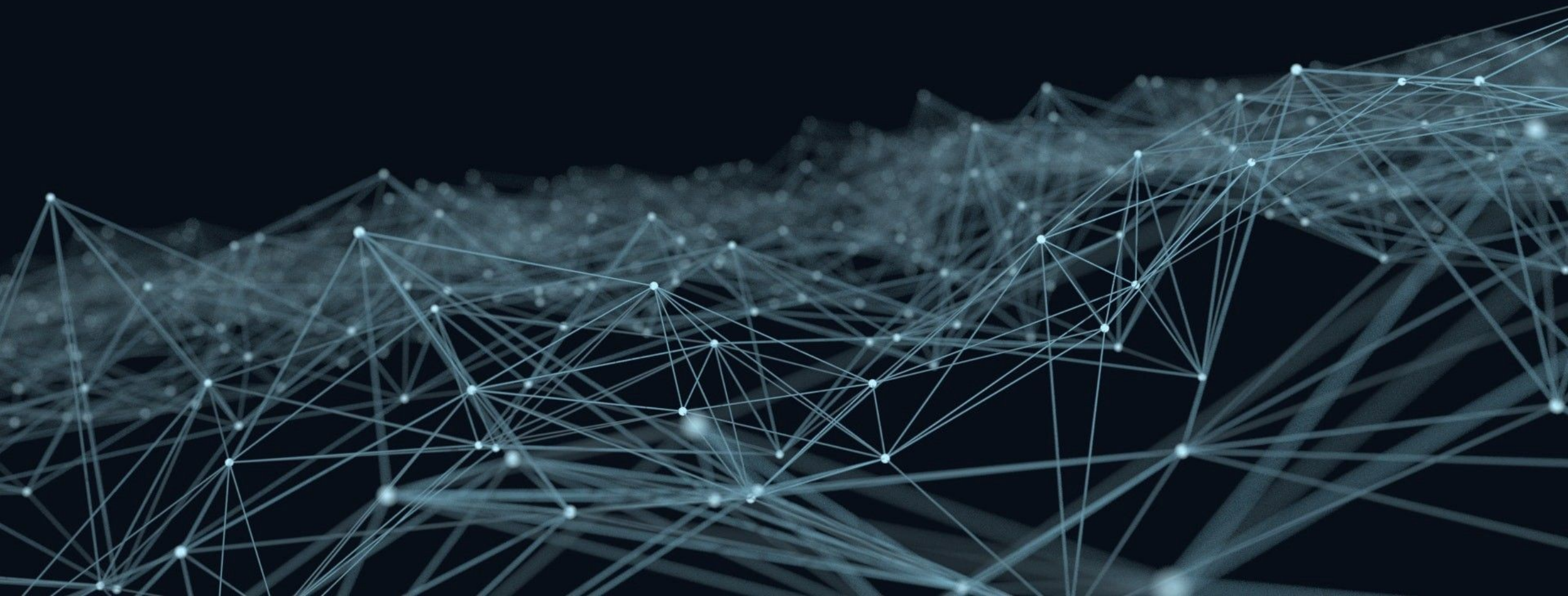
AI Implications for Business Strategy

2. Machine Learning - Regression



Regression

- Unlike classification, a regression model produces a range of outputs rather that a category or classification
- ML has adapted regression techniques from statistics and other sciences
- Regression is based on the hypotheses
  - There are independent variables that serve as inputs to the model
  - There is a dependent variable whose value can be predicted from the inputs
    - For example, predicting house prices (dependent variable) from a combination of two independent variables - zip code and square footage
  - The model is an equation that defines the relationship between the dependent and independent variables
  - Independent variable = the one that we can manipulate
  - Dependent variable = the output of our model

Regression

- The simplest form of regression is a linear case where an independent variable x is used to predict an outcome y
  - Student IQ as a predictor of GPA
  - Engine size as a predictor of gas mileage
  - Calories eaten per day as a predictor of weight change
  - Hat size as a predictor of math ability
- In the linear case, we assume that changes in the input produce corresponding scaled changes in the output
  - For example, doubling the number of calories doubles the number of grams gained

Regression

- The blue dots are measurements
  - There appears to be a linear relation among the points
  - If it were an exact linear relationship, no need for ML
  - The line is our model that predicts the y value given an an x value
- ML regression algorithm is intended to find which line that is the best fit to the data
  - What “best fit” means will be defined later

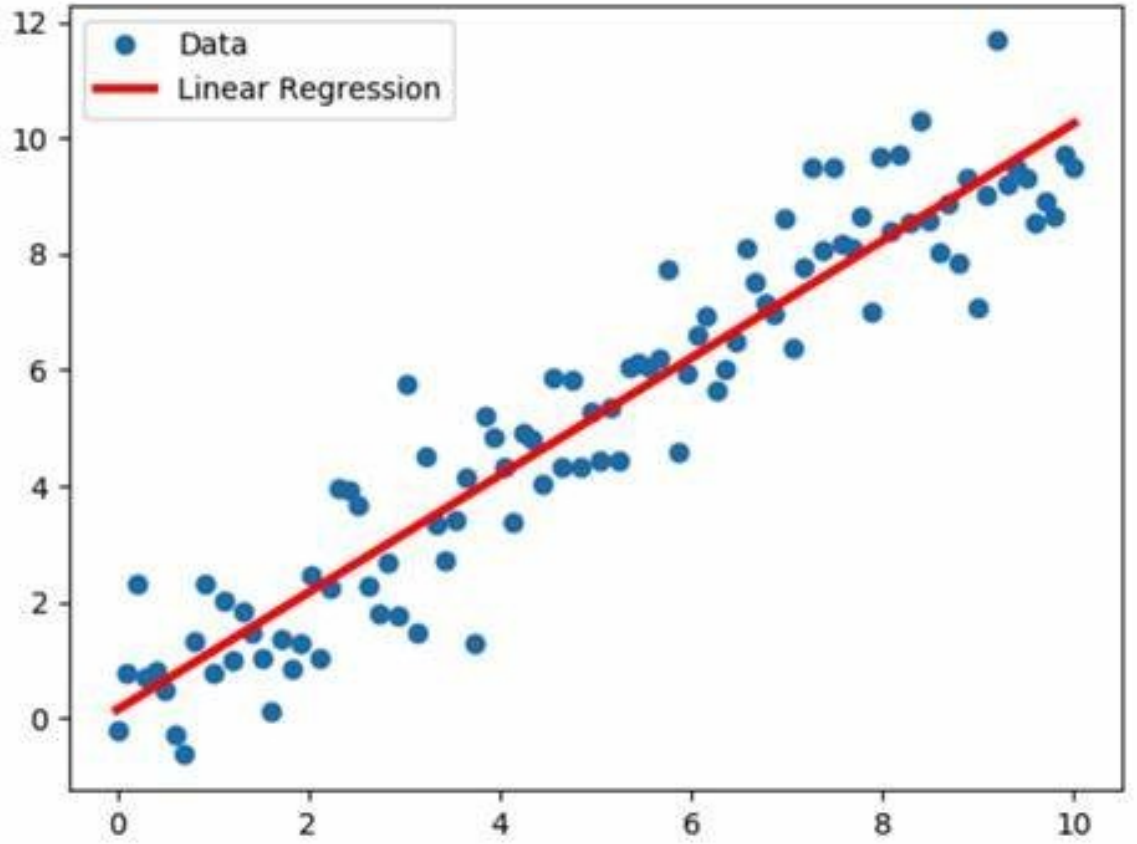


Image Credit: https://www.researchgate.net/figure/Linear-Regression-model-sample-illustration_fig3_333457161

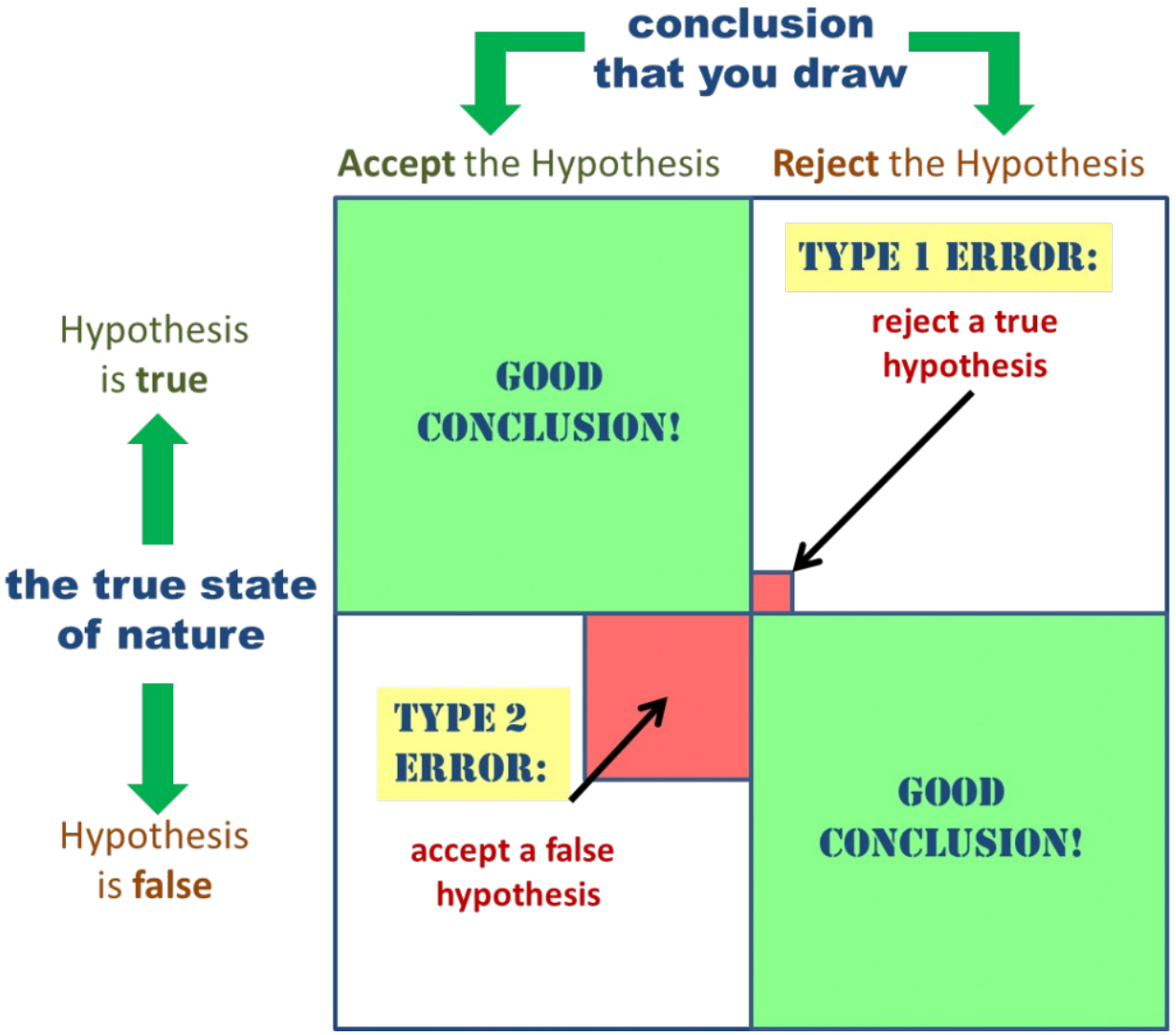
Correlation

- Correlation describes an association between variables
  - When one variable changes, so does the other
  - Correlation is a an observed relationship between variables
  - When variables change together, they are said to exhibit “co-variance”
  - Co-variance does **not** imply any underlying relationship between the variables
- Correlation is strictly observational
  - The ML regressor-predictor hypothesis is a *description* of the relationship between variables that co-vary
  - Our co-variant ML model has predictive value and but has no explanatory power
  - We may have no idea why our variables are co-variant, but we can still measure the relationship

Causation

- Implies a cause and effect relationship
  - Changes in the independent variable *always* produce changes in the dependent variable
  - Causation *always* implies correlation
  - There may be a proposed explanation as to what the causation mechanism is
- We cannot infer causation from correlation
  - The variables may both be influenced by a confounding third variable
    - Heat stroke and ice cream sales may be correlated
    - But they are both affected independently by the outdoor temperature
    - Temperature affects both ice cream sales and the number of cases of heat stroke independently
    - To assume ice cream sales cause heat stroke is not a valid hypothesis
    - Also called a Type I error or a false positive or failing to reject the null hypothesis

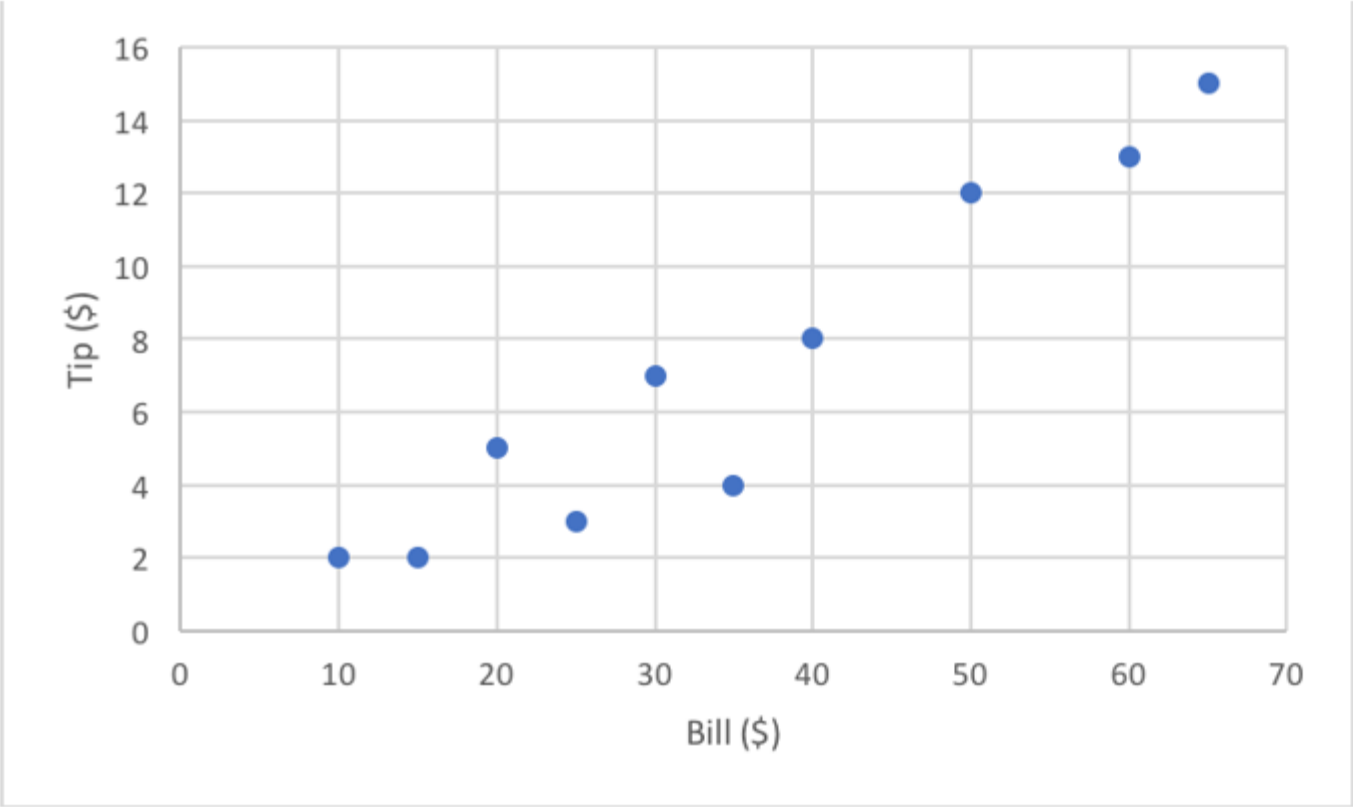
Type I and II Errors



Simple Example

- Looking for a way to estimate a tip from the cost of meal
  - In theory, it should be say 15%
  - In reality, it tends to vary due to other factors
- Creating a casual model is way too difficult
- Given the historical data, can we reproduce the results with a reasonable amount of “wiggle-room?”
  - Our model is an estimator of what, over a large number of observations, the expected value of the tip should be given the cost of the meat
- This depends on the population we sampled from
  - Four star Michelin restaurants in Manhattan versus local diners in rural Arkansas

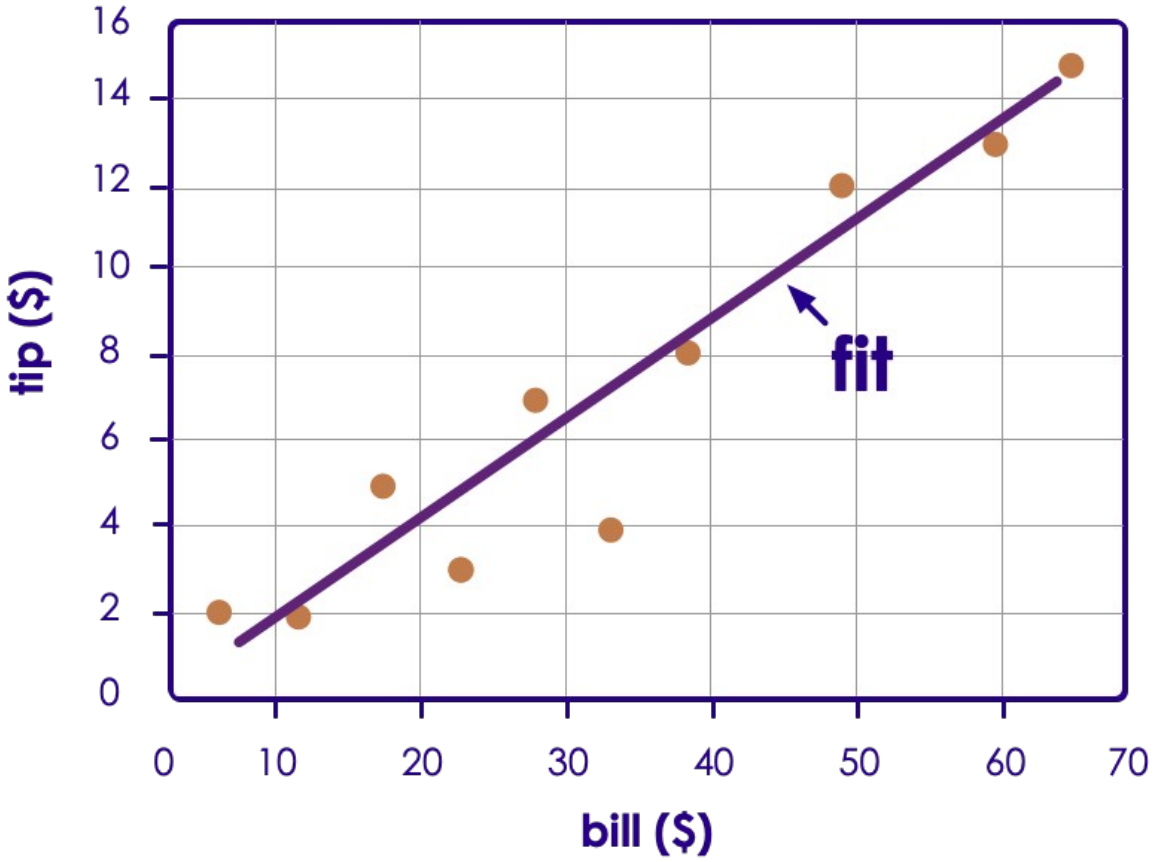
Simple Example



Meal #	Bill ($)	Tip ($)
1	50	12
2	30	7
3	60	13
4	40	8
5	65	15
6	20	5
7	10	2
8	15	2
9	25	3
10	35	4

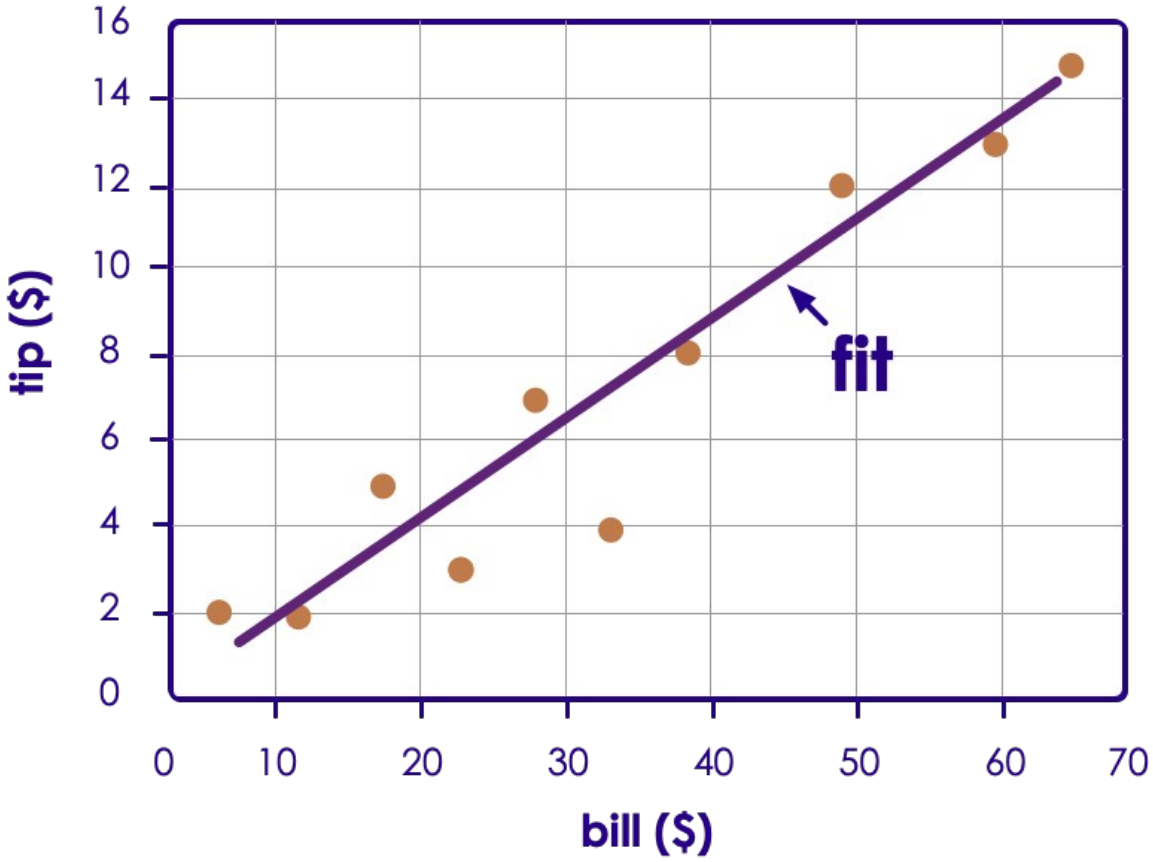
Simple Example

- The proposed line looks like a good fit
  - But we need a quantitative measure of how good the fit is
- This approach is quite ad hoc so far
  - Just eye-balling it
- The ML algorithm should be able to find this best fit line automatically



Simple Example

- The resulting model is a line ax+b = 0
  - The “a” is the regression coefficient
- A more common terminology is to say that the line is a hyperplane
  - Given a space of dimension n, a hyperplane is a sub-space of dimension n-1
  - We start with an assumption of lineality



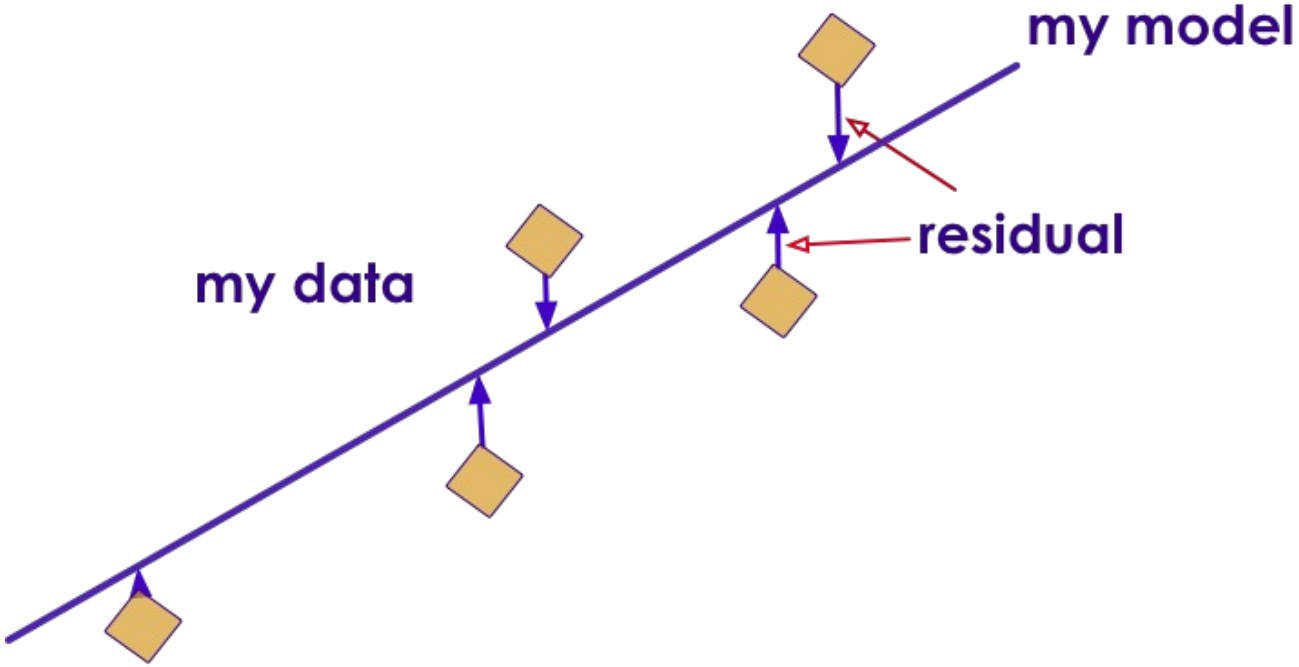
Accuracy

- Assume that we have decided that our model is
  - ax + b = 0
- Where
  - a = 0.2428571
  - x = amount of bill
  - b = -1.40
- How good is this estimate for the training data?
  - Can we do better?

	Bill ($)	Actual tip ($)	estimated tip
observed / known data	50	12	10.742855
	30	7	5.885713
	60	13	13.171426
	40	8	8.314284
	65	15	14.3857115
	20	5	3.457142
	10	2	1.028571
	15	2	2.2428565
	25	3	4.6714275
	35	4	7.0999985
New Data	70	च	15.599997
	80	च	18.028568
	90	च	20.457139
	100	च	22.88571

Residual

- The difference between the actual data and the predicted value is called the residual
- The error for the regressor is some measure of the cumulative residuals
- Then the algorithm can try different lines until it finds one where the residuals are minimized
- Then this would be our optimal model
  - Optimal = lowest total error over all of the data points



Sum of Square Errors

- Also known as
  - Residual Sum of Squares (RSS)
  - Sum of Squared Residuals (SSR)
- In this formula
  - Yi: actual value
  - Ŷi: predicted value
- Properties
  - A good all purpose error metric that is widely used
  - SSE also ‘amplifies’ the outliers (because of squaring)
- For example, if SSE for model-A = 75 and SSE for model-B = 50
  - Model-B might be better fit

$$SSE = \sum_{i=1}^{n} (y_i - \hat{y}_i)^2$$

Mean Squared Error (MSE) (L2)

- Can be sensitive to outliers; predictions that deviate a lot from actual values are penalized heavily
- Easy to calculate gradients
- We get convergence to an optimal model faster

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2$$

Root Mean Squared Error

- Has the advantage that error is in the same units as the data
- RMSE tells us the average distance between the predicted values from the model and the actual values in the dataset.
- So lower the RMSE the better; RMSE=0 would indicate a perfect model

$$RMSE = \sqrt{\frac{\sum_{i=1}^{n} (y_i - \hat{y}_i)^2}{n}}$$

Mean Absolute Error (MAE)

- More robust and is generally not affected by outliers
- Use if ‘outliers’ are considered ‘corrupt data’ (or not critical part of data)

$$MAE = \frac{1}{n}\sum_{i=1}^{n}|y_i - \hat{y}_i|$$

Which Error Measure to Use

- Error functions tell us ‘how far off’ our prediction from actual value is.
- We have seen some popular error functions for regression
- Which one to use?
  - No ‘hard’ rules!, follow some practical guide lines
  - Try them all and see which one gives better results! :-)
  - Most ML libraries allow us to configure the error function very easily

Correlation Measures

- Correlation measures quantify the relationship between two variables
  - Positive correlation: Two variables move in the same direction
  - Negative correlation: Two variables move in opposite directions
  - Measured on a scale from -1 to +1
  - A value of 0 means that no correlation exists
- Correlation is a measure of how well variance in one variable is predicted by variance in the other variable
  - In other words, how well the data fits the model

Correlation Measures

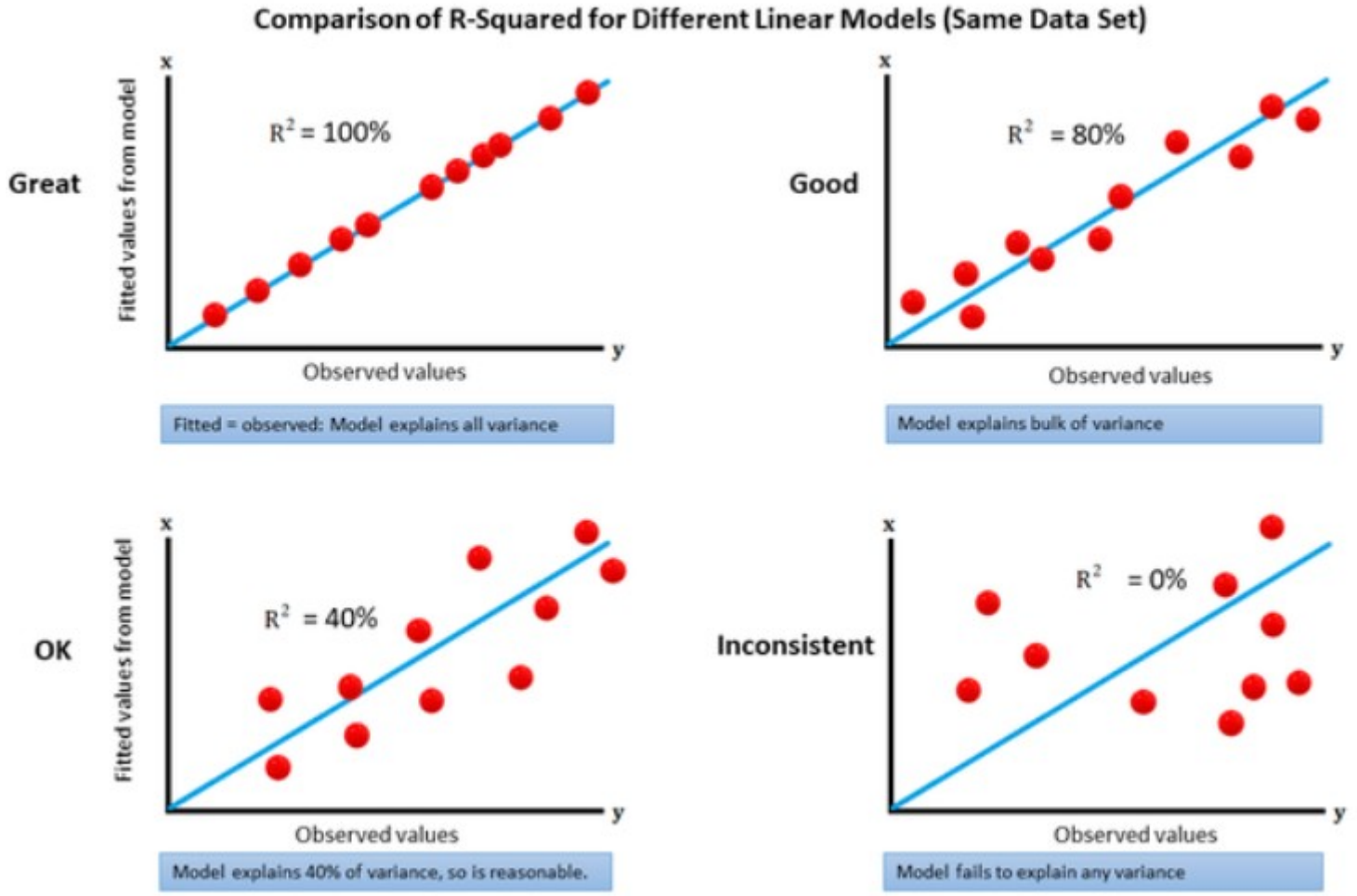


Image Credit: https://remote-imposter.medium.com/covariance-correlation-and-r-squared-explained-with-python-df1d52bfa44b

R^2 Coefficient of Determination

- Coefficient of Determination tells us how well our model ‘fits’ the data
- R^2 usually between 0 and 1.0
  - 0 : The model does not predict the outcome.
  - Between 0 and 1 : The model partially predicts the outcome
  - 1 : The model perfectly predicts the outcome.
  - Values < 0 often mean the wrong model was used or some other problem
- So we want R^2 to be as close 1.0 as possible

$$R^2 = \frac{N\sum xy - \sum x \sum y}{\sqrt{\left[N\sum x^2 - (\sum x)^2\right]\left[N\sum y^2 - (\sum y)^2\right]}}$$

What R^2 Measures

$$
\begin{aligned}
R^2 &= 1 - \frac{\text{Residual variance}}{\text{Total variance}} \\
&= \frac{\text{Total variance} - \text{Residual variance}}{\text{Total variance}} \\
&= \frac{\text{Explained variance}}{\text{Total variance}} \\
&= \text{Fraction of total variance explained}
\end{aligned}
$$

Image Credit: https://remote-imposter.medium.com/covariance-correlation-and-r-squared-explained-with-python-df1d52bfa44b

Multiple Regression

- The basic 2-d case can be extended to an arbitrary number of dimensions
  - Assumption that the contribution of each variable is linear
  - All of the formulae and correlation measures work in n-space and the regression line is now a hyperplane
  - Assume that every prediction is a linear combination of the independent variables

$$y = \beta_0 + \beta_1 X_1 + \ldots + \beta_n X_n + \epsilon$$

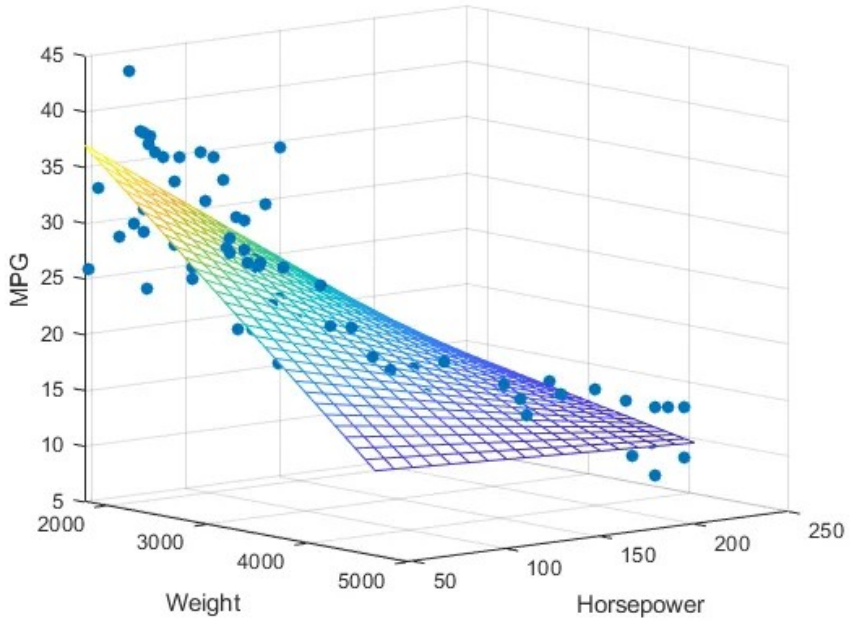


Image Credit: https://towardsdatascience.com/graphs-and-ml-multiple-linear-regression-c6920a1f2e70

Multi Co-linearity

- Can occur when we have more that one independent variable
  - We usually assume that the independent variables are not correlated
  - However, when two or more variable are predictable from the others we have co-linearity or multi co-linearity
    - *Example, we a have the three variables height, weight and BMI for medical data*
    - *BMI is a function of height and weight and can be predicted from the other two so adding the BMI as a variable does not improve the model but does increase the computational load*
- Typical co-linearity cases
  - One predictor variable is a multiple of another (height in inches and height in cms)
  - One variable is a transformed version of the other (original cost and cost in 2023 dollars)
  - Dummy variable - single categorical variable “gender” is converted to two binary variables, male and female
  - Common hidden variable - both of the variables are correlated with the same third variable which may not be expressed in our data set

Multi Co-linearity

- Co-linearity effects
  - Weakens the statistical measures overall and reduces the ability to identify significant variables
  - Reduces potential explanatory power of the model
  - Tends to increase Type II inference errors
- Measuring Co-linearity
  - We can measure pairwise correlation between independent variables
  - Or use a special measure called VIF = Variance Inflation Factor
- Fixing Co-linearity
  - Remove the duplicate variable
  - Reduce the variables to a smaller set with no co-linearity (partial least squares regression)

Multi Co-linearity

Multivariate

Correlations

	Weight	Height	BMI
Weight	1.0000	0.5129	0.8668
Height	0.5129	1.0000	0.0220
BMI	0.8668	0.0220	1.0000

Scatterplot Matrix

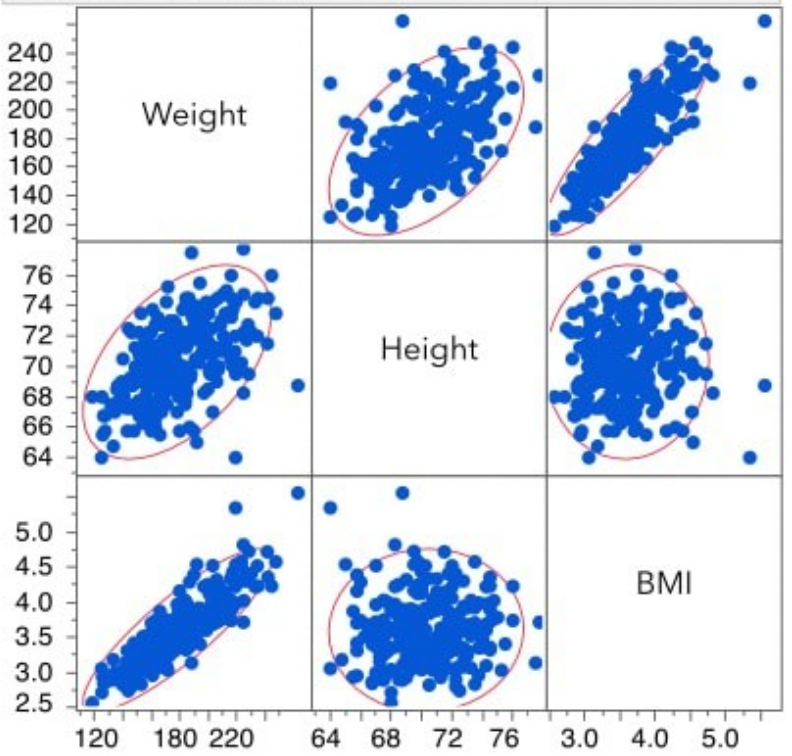


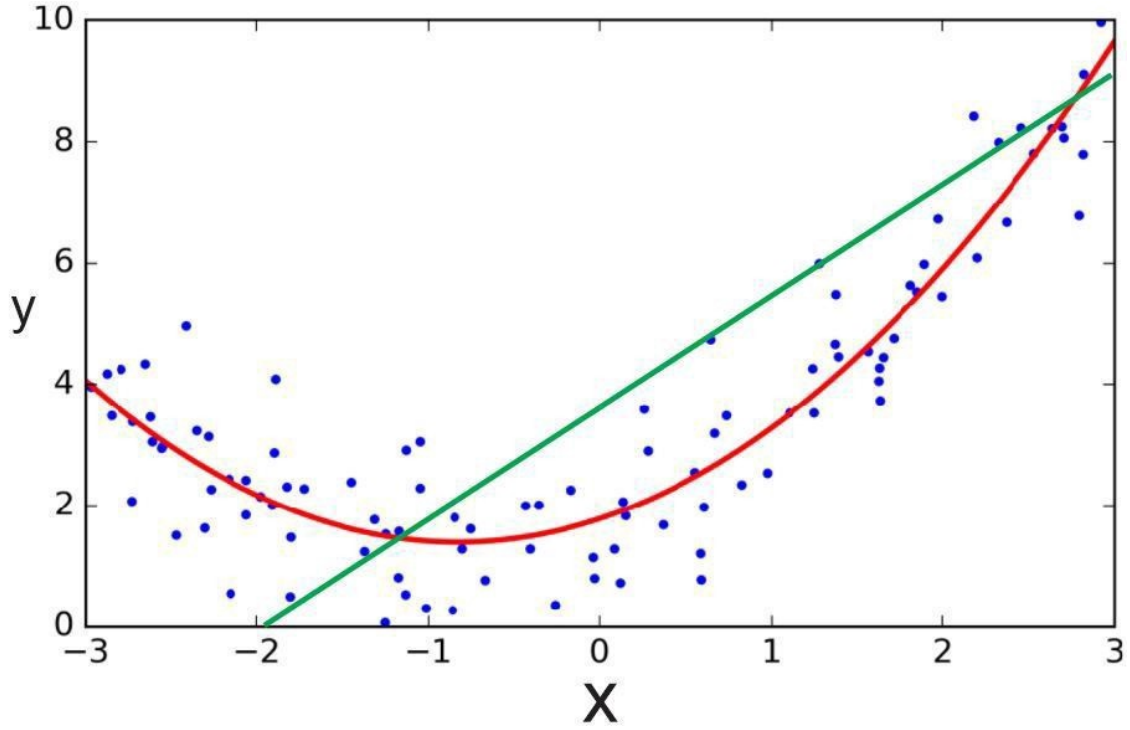
Image Credit:https://www.jmp.com/en_us/statistics-knowledge-portal/what-is-multiple-regression/multicollinearity.html

Variable Importance

- Not all independent variables are equal
  - Eg. Percent body fat is probably a better predictor of diabetes than height
  - In other words, some variables contribute more the prediction than others
- Model performance is improved by discovering and prioritizing the important independent variables
- Standardize or normalize all of the variables then compare coefficients
  - Without normalizing, this comparison is meaningless
  - Variables using vastly different scales may mask which ones are important
- Changes in R^2 as each variable is included in the measure
  - The more important a variable, the greater the change in R^2 when it is added
  - Look for the variable that produces the greatest increase in R^2

Polynomial Regression

- Sometimes the best predictor is not linear but a higher degree function
  - Like a parabola for example



$$Y = \theta_0 + \theta_1 x + \theta_2 x^2 + \theta_3 x^3 + \ldots + \theta_n x^n$$

Image Credit:https://deeppatel23.medium.com/polynomial-regression-29c0a1225f44

Polynomial Regression

- Multi co-linearlity is an issue
  - We are using different powers of the same independent variable
- Very complex to compute
- Often can be piece-wise approximated by a series of linear functions
- The additional computational load of a polynomial model may not be justified
  - Linear approximations are easier to compute
  - Linear approximation are often “good enough”

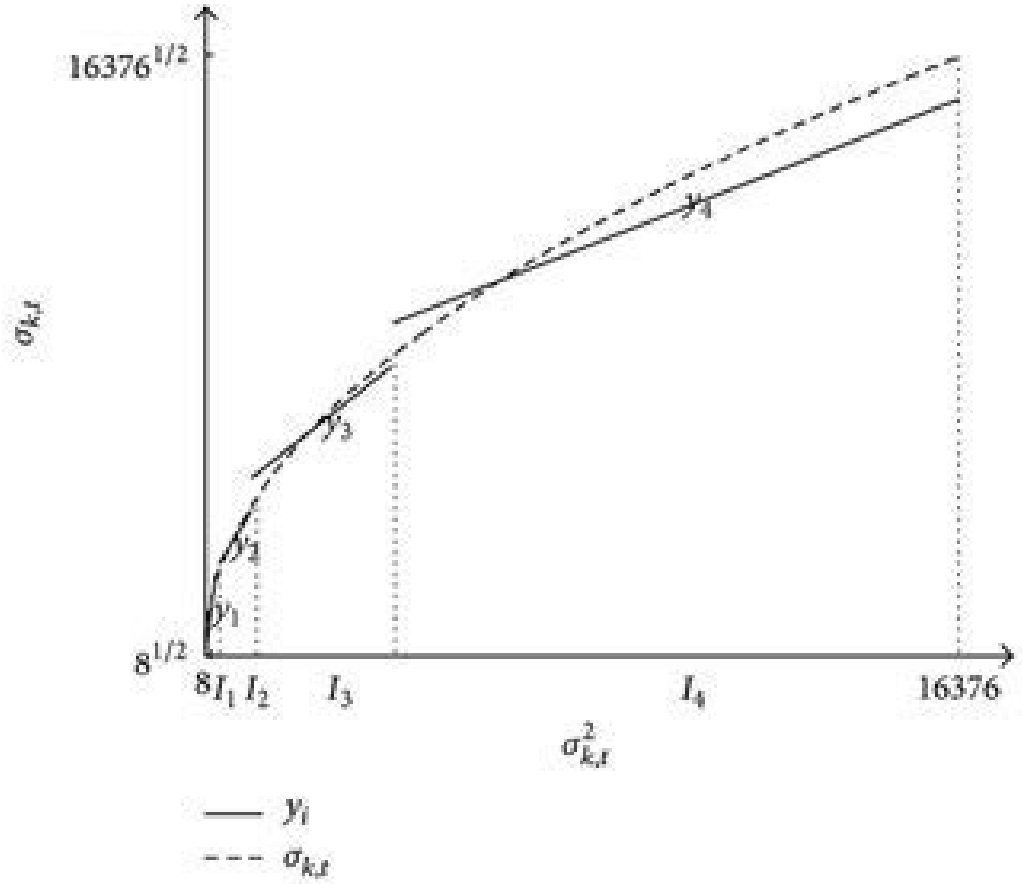
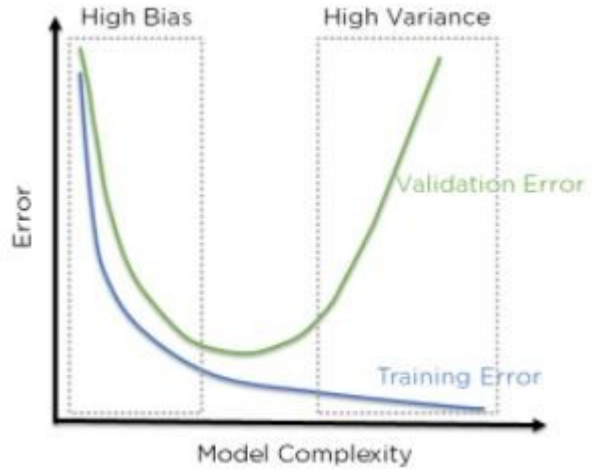


Image Credit:https://www.researchgate.net/figure/Piecewise-linear-polynomial-approximation-of-the-s-k-t-function-implemented-in-the_fig3_257816607

Bias and Variance



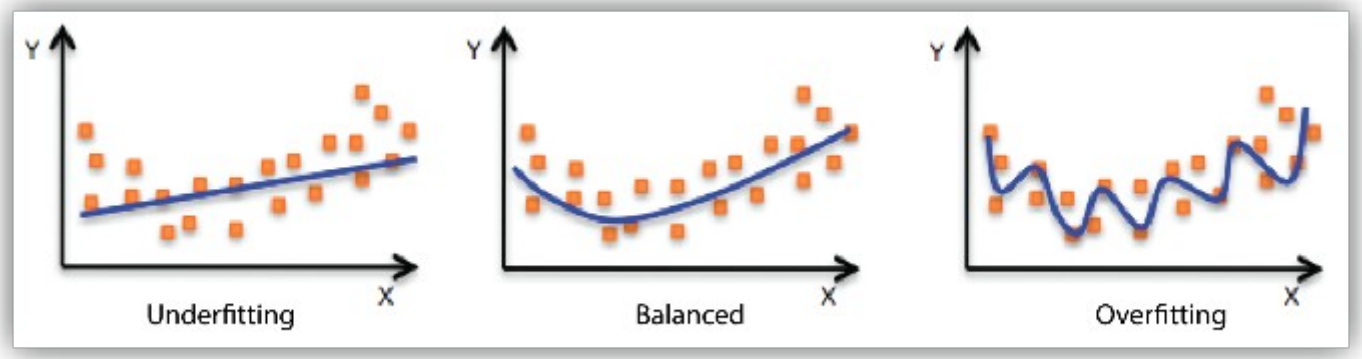


Image Credit: https://www.simplilearn.com/tutorials/machine-learning-tutorial/bias-and-variance

Gradient Descent

- How do we actually train a model
- We have a loss function (normally denoted by “J”) that
  - For a given model, computes a total error for the model for a set of points
  - We take the gradient to look for the direction to proceed that gives us the best reduction in error

Gradient descent algorithm

$$\text{repeat until convergence } \{$$

$$\theta_j := \theta_j - \alpha \frac{\partial}{\partial \theta_j} J(\theta_0, \theta_1)$$

$$(\text{for } j = 1 \text{ and } j = 0)$$

$$\}$$

Gradient Descent

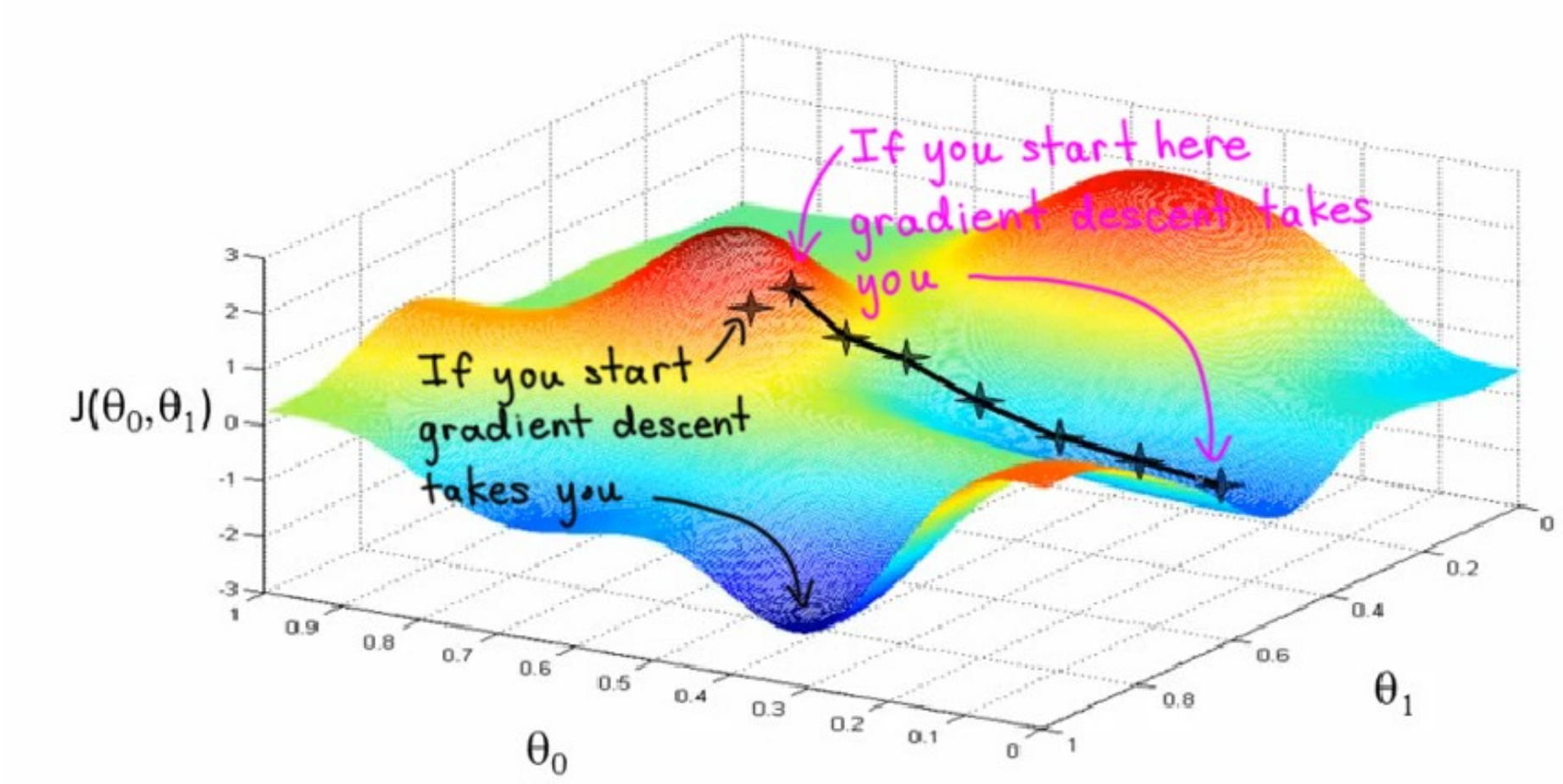


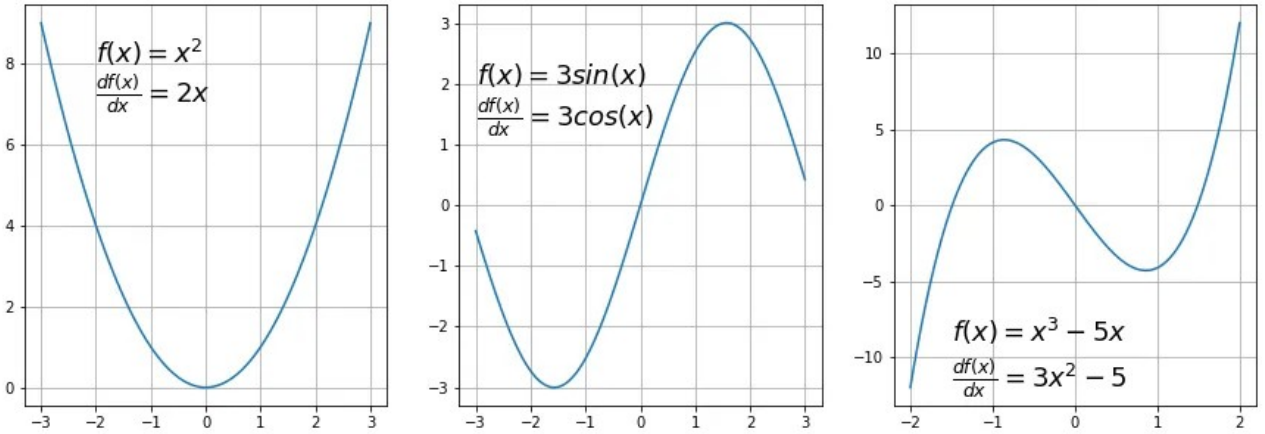
Image Credit https://regenerativetoday.com/machine-learning-gradient-descent-concept/

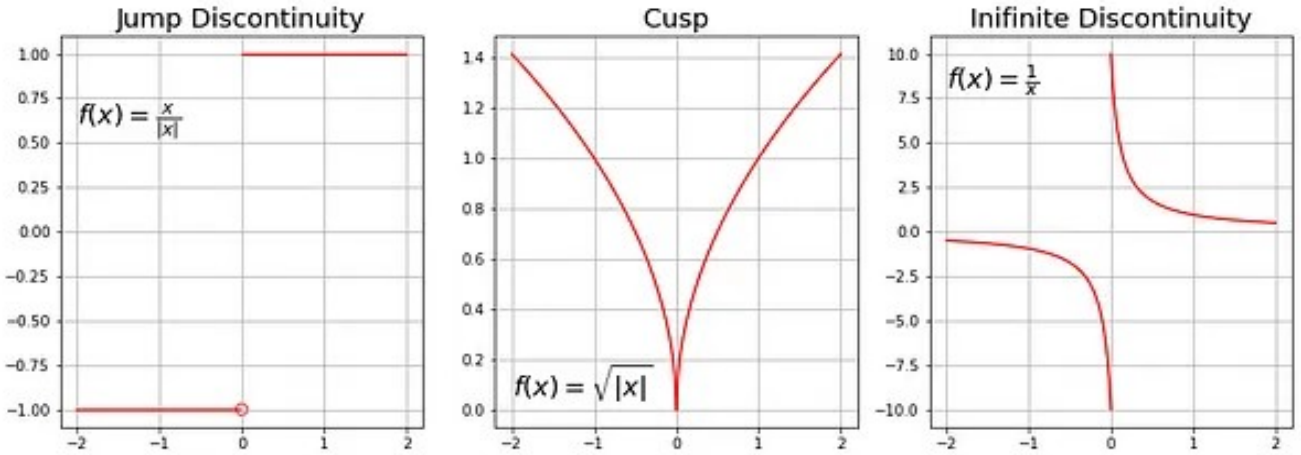
Gradient Descent

- Essentially, we minimize the cost or loss function for the model
  - We have seen a variety of loss functions earlier
- For GD to work the loss function must be:
  - Differentiable
  - Convex
- We control the rate of convergence with a hyper-parameter called the learning rate
  - This specifies the “size” of the increment to be taken in the direction that reduces loss

https://towardsdatascience.com/gradient-descent-algorithm-a-deep-dive-cf04e8115f21

Differentiable / Non-differentiable

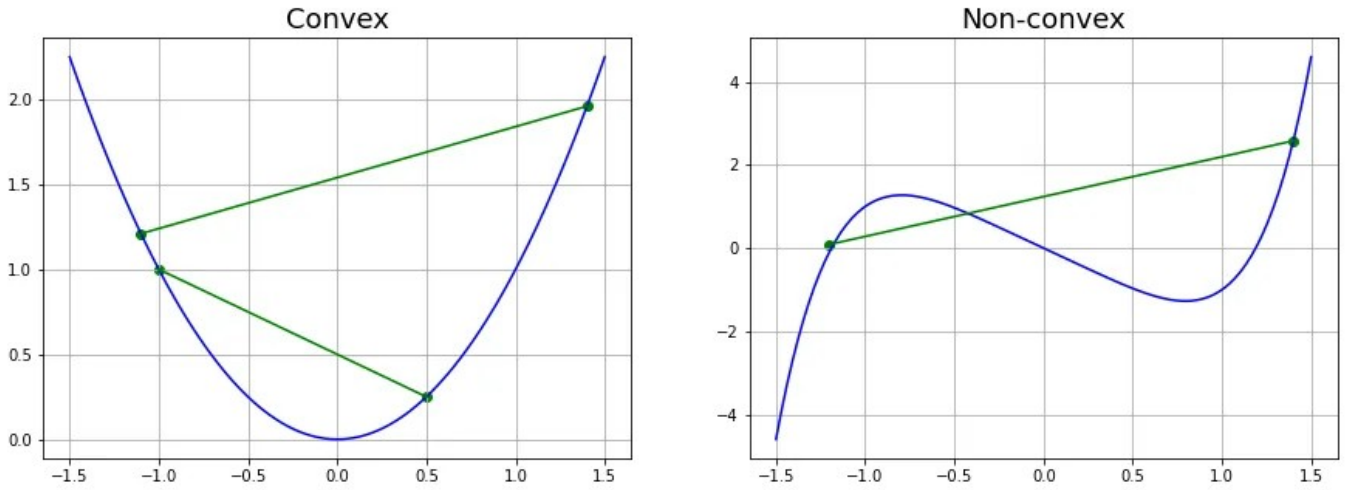




https://towardsdatascience.com/gradient-descent-algorithm-a-deep-dive-cf04e8115f21

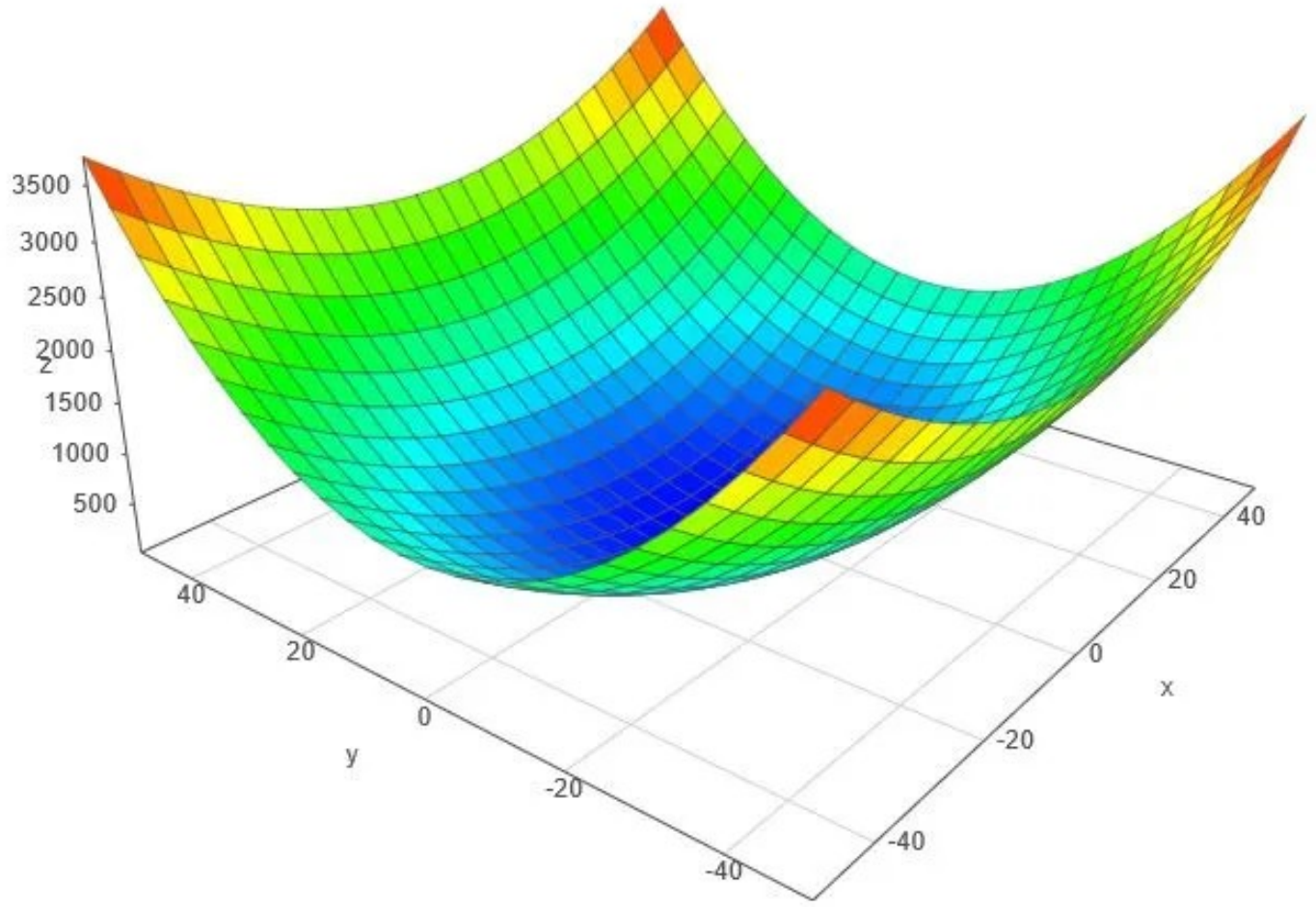
Convex / Non-convex

- Convex function – any two points on the function can be connected by a line segment that doesn't pass through any other points on the function



https://towardsdatascience.com/gradient-descent-algorithm-a-deep-dive-cf04e8115f21

Convex Surface



https://towardsdatascience.com/gradient-descent-algorithm-a-deep-dive-cf04e8115f21

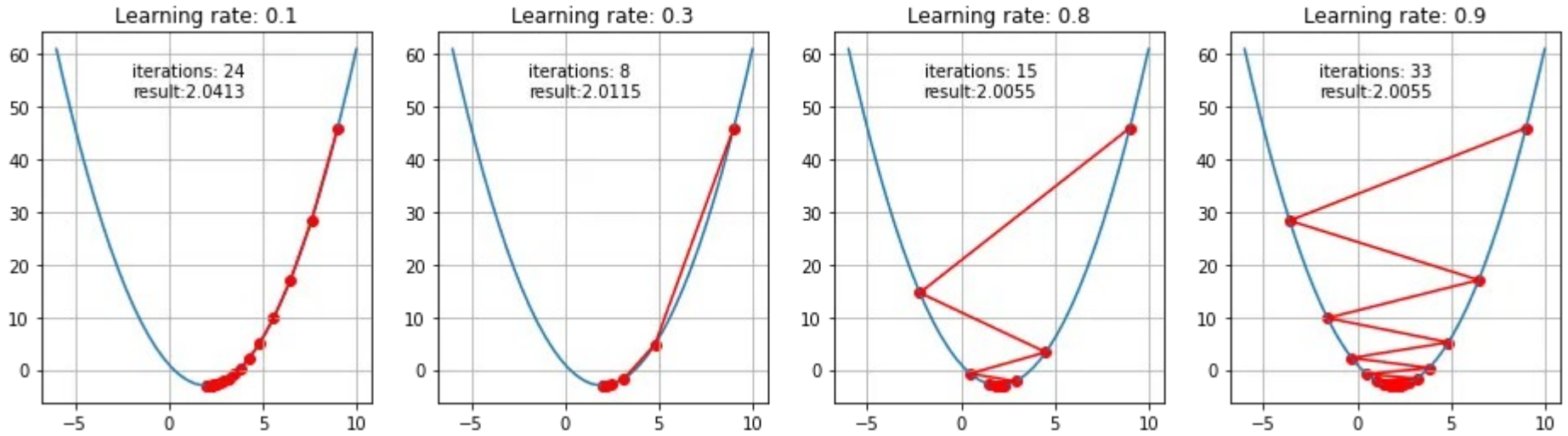
Gradient Descent Process

- The GD Process
  - Choose a starting point
  - Calculate the gradient from the derivative
  - Choose the direction that the gradient is steepest
  - Move a set increment (learning rate) in that direction to a new point
  - If we have reached the maximum number of iterations then stop
  - If the required accuracy has been reached then stop
  - Otherwise make the new point our starting point and repeat

https://towardsdatascience.com/gradient-descent-algorithm-a-deep-dive-cf04e8115f21

Learning Rate

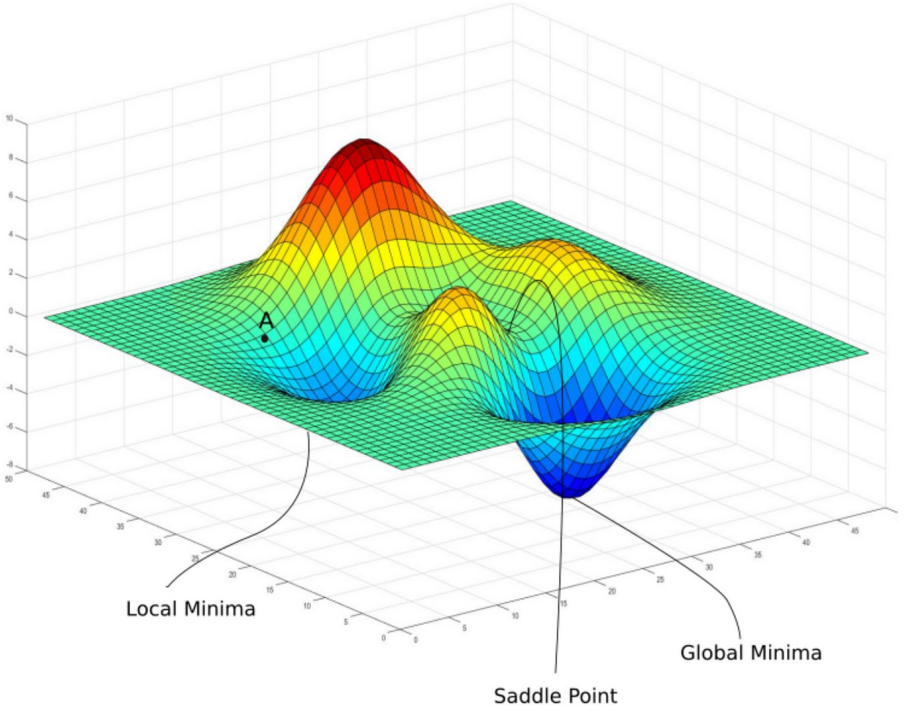
- We “tune” or select the learning rate
  - Too high and the GD function bounces around
  - Too low and the GD converges too slowly

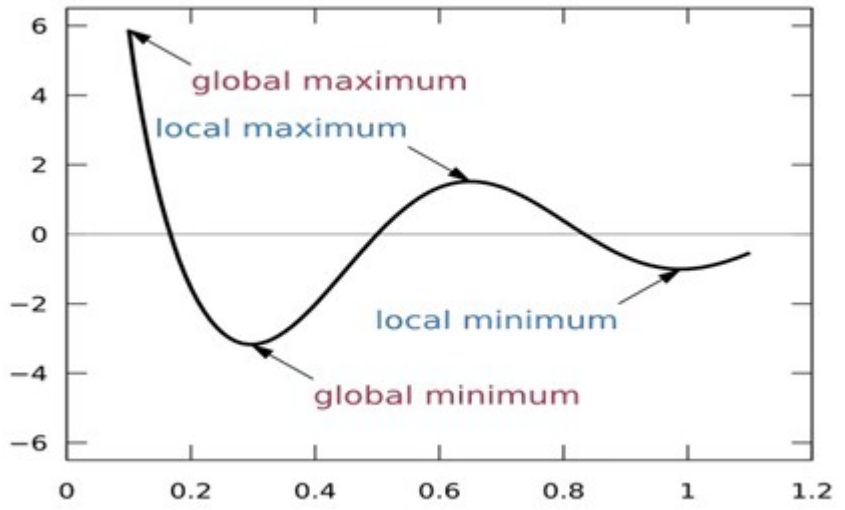


https://towardsdatascience.com/gradient-descent-algorithm-a-deep-dive-cf04e8115f21

Semi-convex Functions

- These are functions with saddle points
  - Represent local minima and maxima
  - All semi-convex functions are locally convex
- Final result depends on the range of the data
- Also depends on the starting point chose
- Often local minimal are good enough for GD

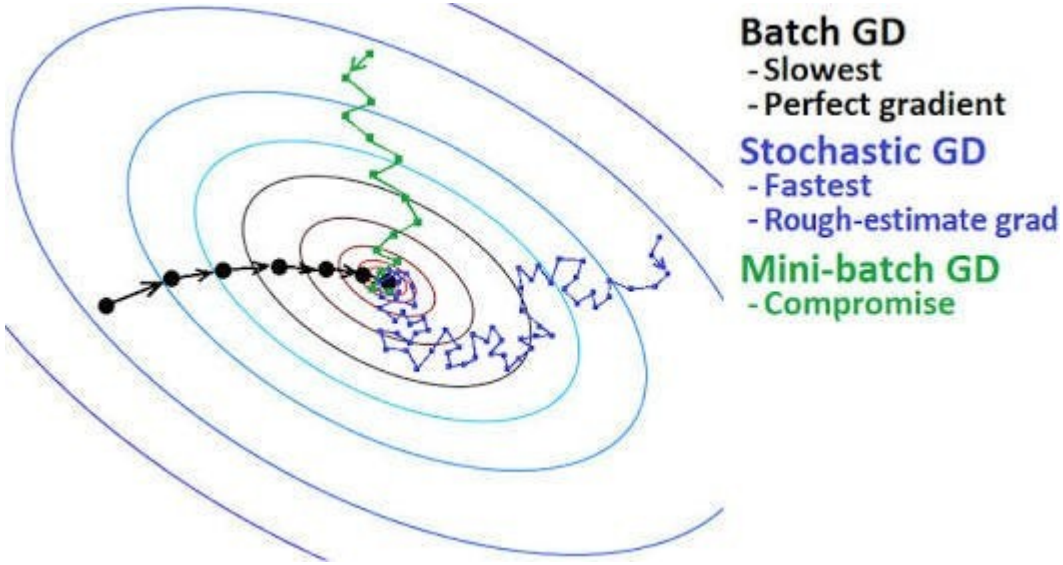




https://towardsdatascience.com/gradient-descent-algorithm-a-deep-dive-cf04e8115f21

Stochastic Gradient Descent

- Computing GD can be very expensive when
  - Training data sets are large
  - Computing the GD function is computationally expensive
- Stochastic GD samples the training data
  - Computes GD convergence on the sample
  - Then shuffles the data and does it again
  - If more than one data point is in the sample, it is “batch SGD”
  - Also helps avoids local maxima and minima for semi-convex functions



https://towardsdatascience.com/gradient-descent-algorithm-a-deep-dive-cf04e8115f21

End of Module