Linear Regression

Read Chapter 7 (Regression) of the Textbook

Regression Models

Linear Regression - Independent variables are being manipulated in an experiment in order to observe the effect(relationship) on a DEPENDENT variable

- To understand the application of regression analysis in data mining
  - Linear/nonlinear
  - Logistic (Logit)

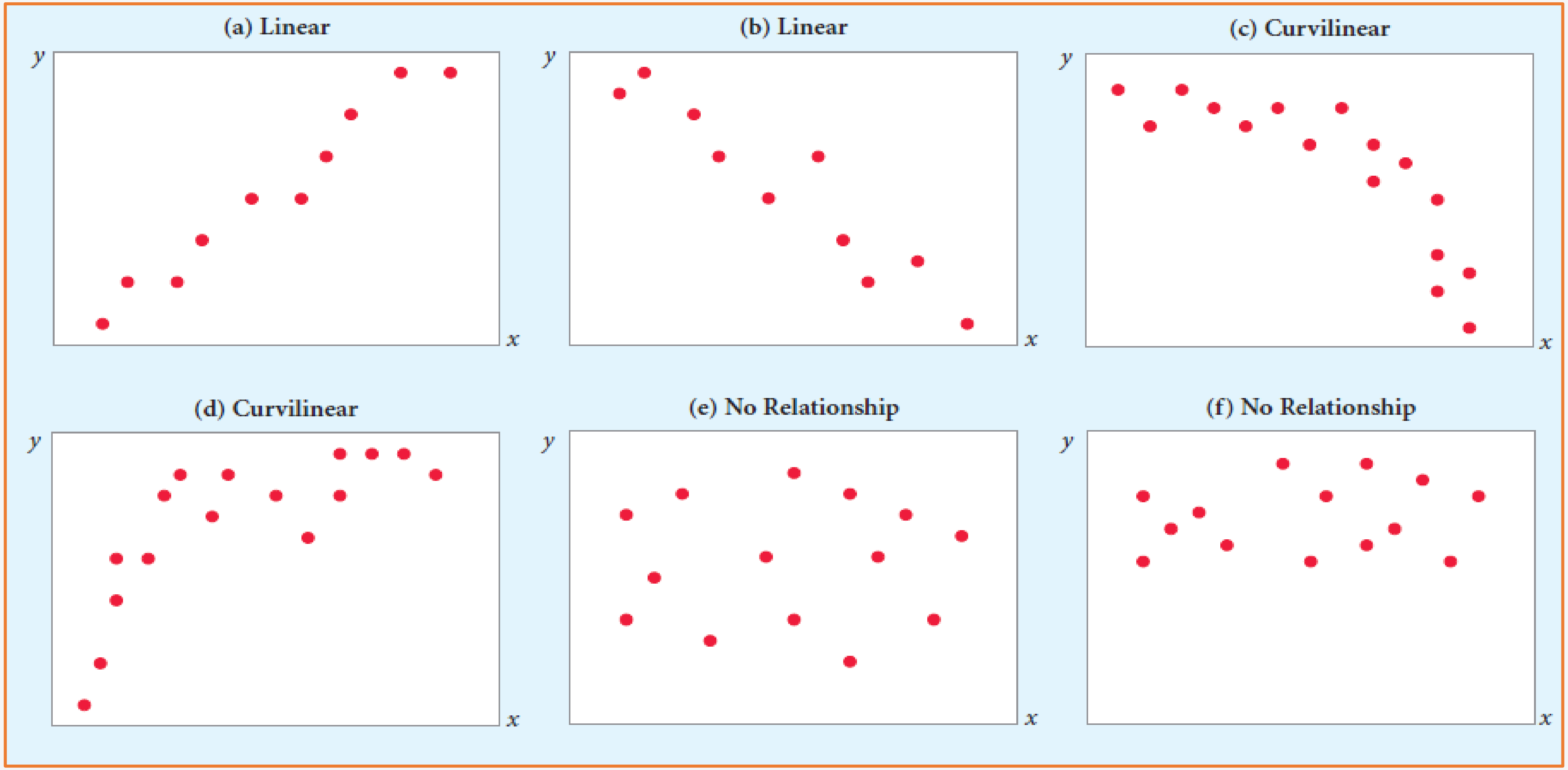
Linear Regression
Purpose: Predict relationship between several independent variables and one DEPENDENT variable.

Objective: The objective is to find the best-fitting curve for a dependent variable in a multi-demensional space, with each independent variable being a dimension.

- To understand the key statistical measures of fit

Goals of linear regression.
1. List all variables for making the model
2. Establish a dependent variable (DV) of interest ********************
3. Examine visual (if possible) relationships between variables of interests
4. Find a way to predict DV using other variables

Relationships between variables



When the data shows linear relationship

Correlation high = linear relationship

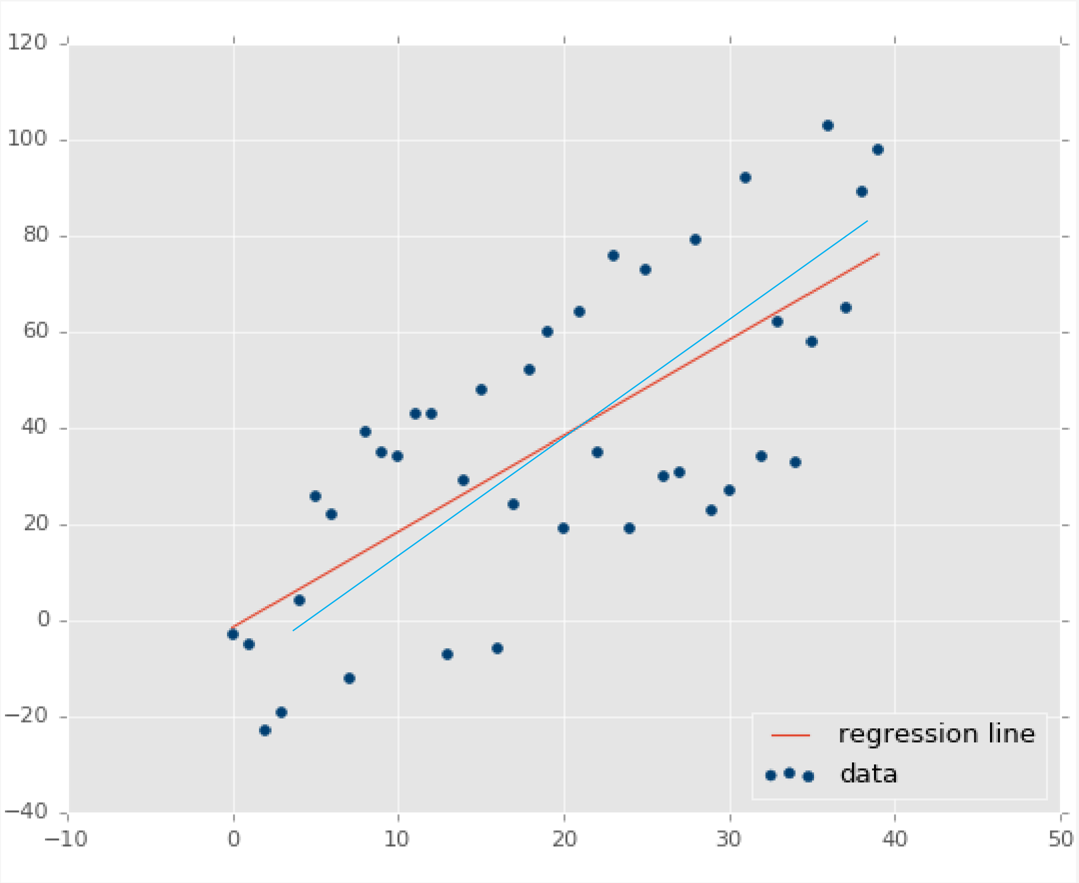
Correlation is high (positive or negative) and Scatter plots display a linear relationship

First model come to mind is

$$Y = m X + b$$

But still, there can be many lines that can “kind of” fit the data as well

Question: How to pick the “best-fit” line?



How to find the best fitting line?

N = number of data points

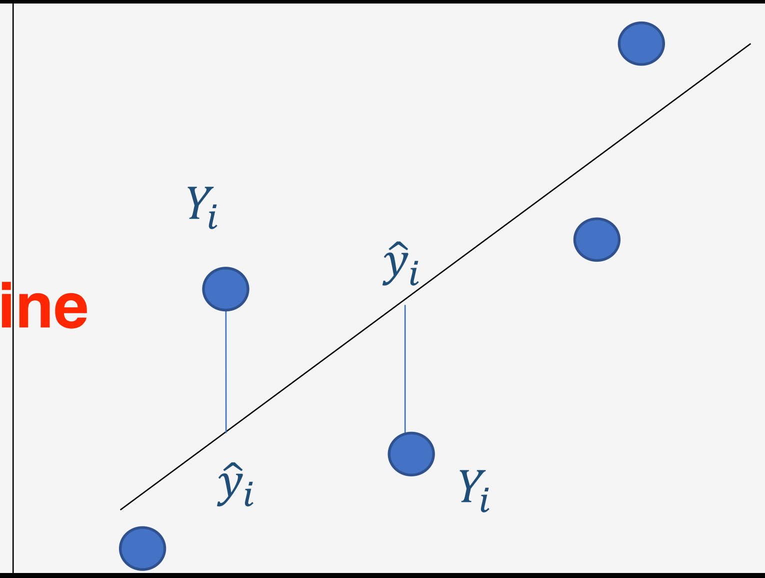
$\hat{y}_i$ = prediction, $Y_i = actual\ value$

Define Mean Squared Error (MSE)
To be the square of the distance between actual and predict Y values

MSE Purpose: Metric to be used for judging how good your BEST FIT LINE is.

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

yi with the bar is the prediction of the best line



Best fitted line is the line that minimize the MSE =>

Least Square Methods

MSE = Cost function. Objective function to come up with our model parameters. Model coefficient. We are using the training data set. Once we have the model. We then apply the model to testing dataset and THEN WE USE A PERFORMANCE METRIC (i.e. R-Squared) on the testing dataset. So we are using R-Squared as a model Performance metric.

R-square as metrics for determining "goodness" of the fit

- Determining the relationship between predictor & outcome
- Relationship Among SST, SSR, SSE

1. R-Sqaured > 70% is GREAT
2. Higher R-Squared means lower SSE (less error) which means a better model.

$$r^2 = \mathrm{SSR}/\mathrm{SST}$$

$$\overset{\text{SST}}{\sum(y_i-\bar{y})^2} = \overset{\text{SSR}}{\sum(\hat{y}_i-\bar{y})^2} + \overset{\text{SSE}}{\sum(y_i-\hat{y}_i)^2}$$

where:

SST = total sum of squares
SSR = sum of squares due to regression
SSE = sum of squares due to error

Higher R-square => Lower SSE => Better Model

R-square is 0% to 100%, anything > 70% is great

Common Theme, Toolbox and Research workflow in Data Science

Apply different algorithms to solve different problems based on the same <Theme> and <Research Workflow>

Algorithms

- SVM
- KNN
- Naïve Bayes
- Neural Network
- Logistics Regression
- NLP

Theme

Research Workflow

Toolbox

Problems

- Regression
- Classification
- Recommendation System
- Clustering
- Association

Common Theme, Toolbox and Research workflow in Data Science

Will use Linear Regression for many of the general practices in building models, some of them are

- Split the dataset into training set and a testing set
- Use standard metrics to judge model performance
- K-fold cross validation

Ex: Dataset that has 10,000 data points. Lets say we want 2,000 out of 10,000 for testing, the 8,000 will be used for training (training set) the model and the 2,000 will be for testing (testing set or out-of-sample-set) the model. DONT use the testingset in your EDA (exploratory data analysis).

K-Fold is using ANOTHER training set because sometimes the data is sensitive. SO by repeating the process K amount of times. We can be confident with the model. Thats why it's called K-Fold for K amount of trials.

Linear Regression

Learning by doing

z1 = x z2 = x^2, .. z20 = x^20

m1, m2, m3, …, m20

y = SUM m_i x_i + b

y_pred = m1 x1 + m2 x2 + .. m20 20 + b

Linear Regression Continued

Pick the HIGHEST correlation FIRST and for the SECOND variable pick the factor that has the LOWEST correlation with the FIRST correlation BUT with a high enough correlation with the dependent (Y Variable) variable. So X1 and X2 can have NO/LOW correlation to one another but as long as the 2nd (X2) is correlated highly to the Y variable it's good.

Challenges Number 1 multi-linear regressions

Y = beta_0 + beta_1 X_1 + beta_2 X_2 +

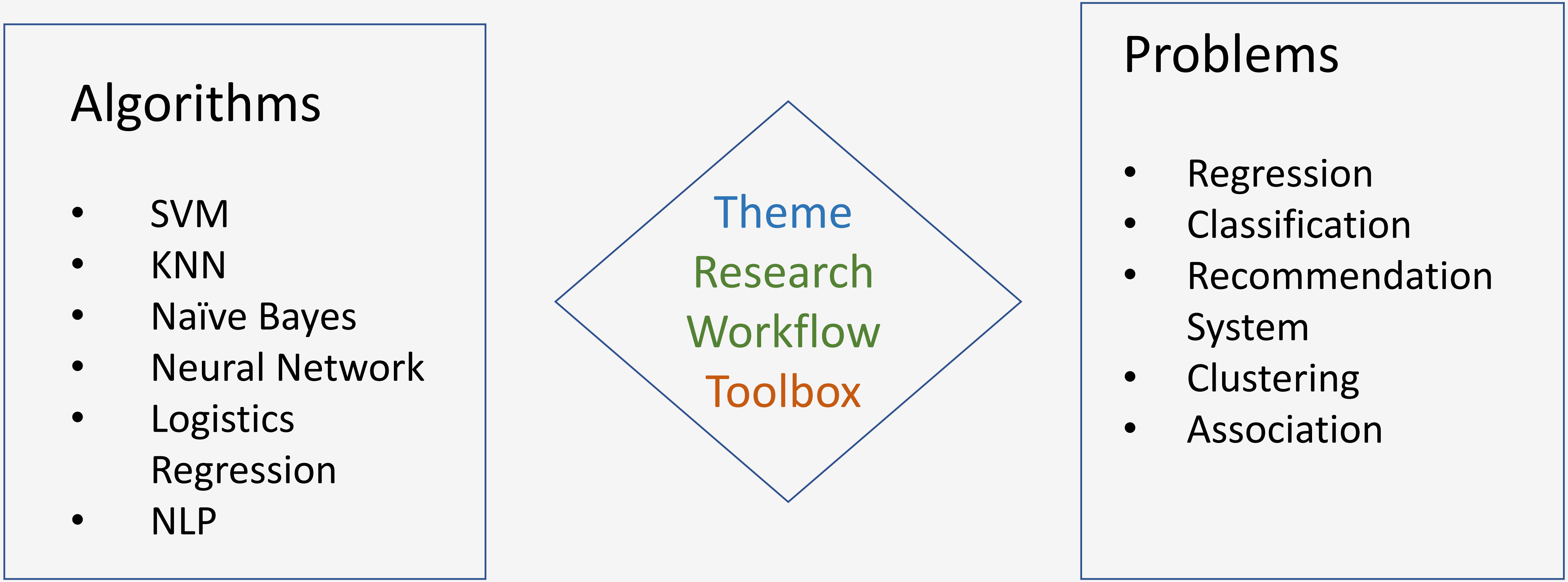
- Collinearity
  - Pick the factors with highest correlation first, but what about the second factors?
    - Second highest correlation coefficients or lowest correlation with the first factor, but with high enough correlation with the dependent variable
  - Solution is: find an Orthogonal independent vectors
    - PCA (Principal Components Analysis)

Lowest correlation with the 1st INDEPENDENT variable but high enough correlation with the DEPENDENT variable

=> Features Engineering

Common Theme, Toolbox and Research workflow in Data Science

Apply different algorithms to solve different problems based on the same <Theme> and <Research Workflow>



Algorithms

- SVM
- KNN
- Naïve Bayes
- Neural Network
- Logistics Regression
- NLP

Theme

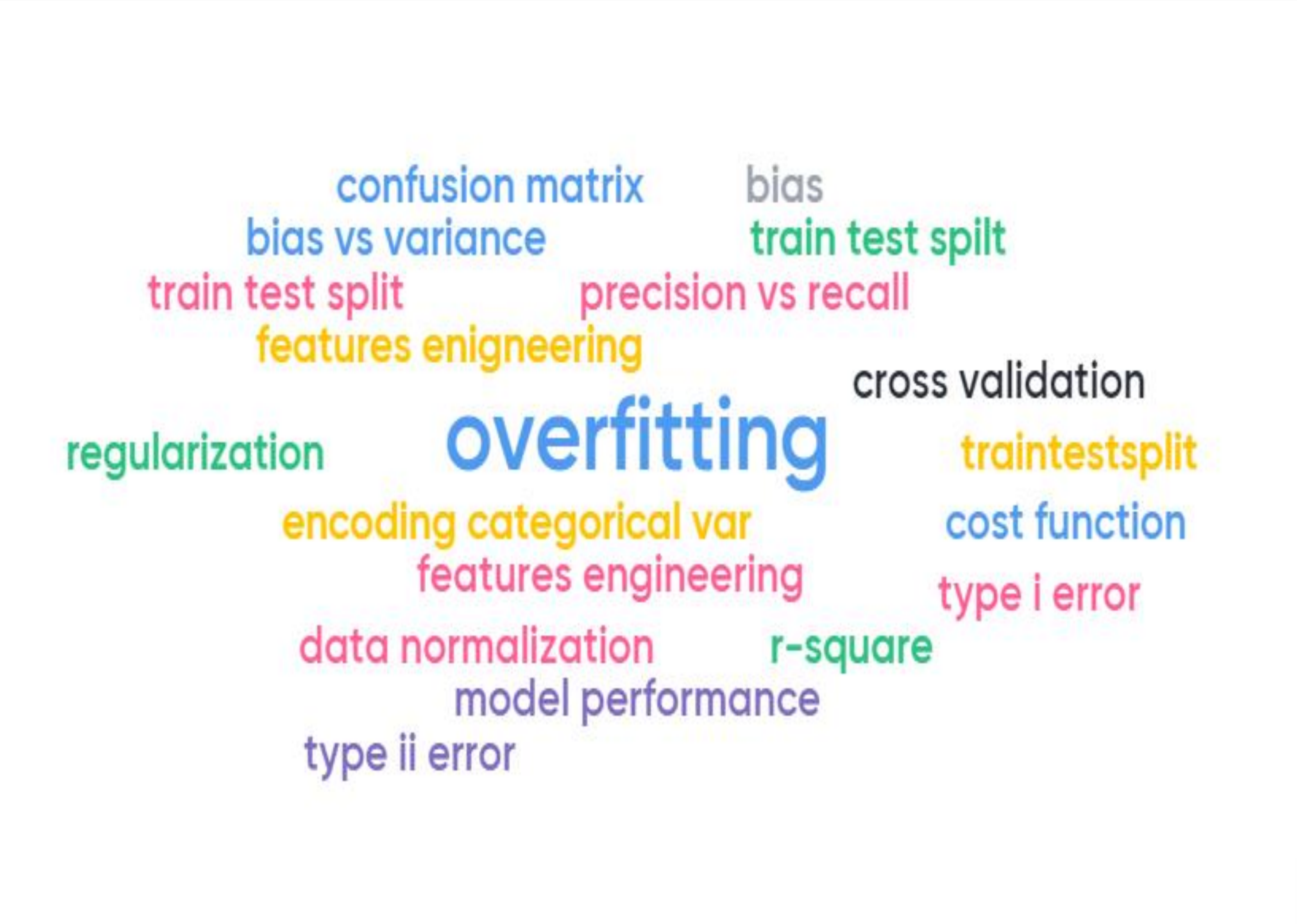
Research Workflow

Toolbox

Problems

- Regression
- Classification
- Recommendation System
- Clustering
- Association

Common Theme in Machine Learning



Linear Regression

Transform a funtion that is non-linear to a linear function

Challenge Number 2:

- Relationship is NOT linear
- Solution: may become linear after transformation

Y = a X^2 + b X + c => Y = b1 Z1 + b2 Z2 + b3
where Z1 = X^2 and Z2 = X

N = N_0 exp(-lambda * t) => ln (N/N_0) = - lambda * t + c

=> Y = m X + b
where Y = ln (N)
X = t

Polynomial Regression

Learning by doing

Simple vs More complicated model

- Using a model with more parameters (more features, more predictors), you are guaranteed to fit your in-sample data (training data) better
  - More parameters => R-squares always increases

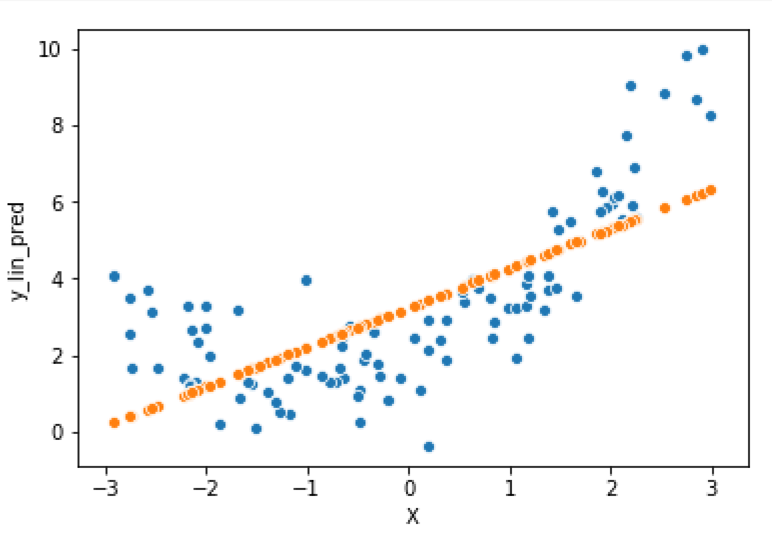
Meaning more variables the MORE R-squared INCREASES but it does NOT mean the model is better

- BUT it doesn't mean you have a better model
  - Adjusted R-squares (R-squares adjusted by penalizing models with more parameters)

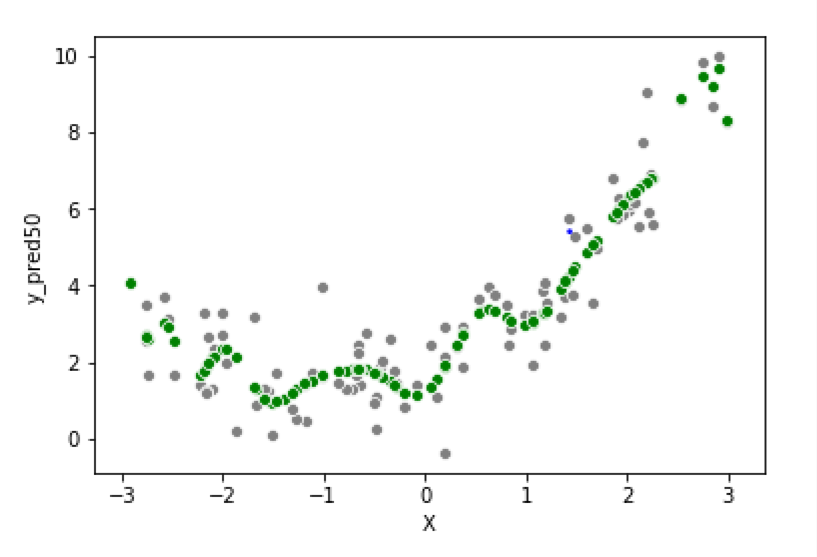
**Similar equation to R-Squared (can google this)
DONT need to remember formula, just what makes it different**

Lesson Learned from Polynomial Regression

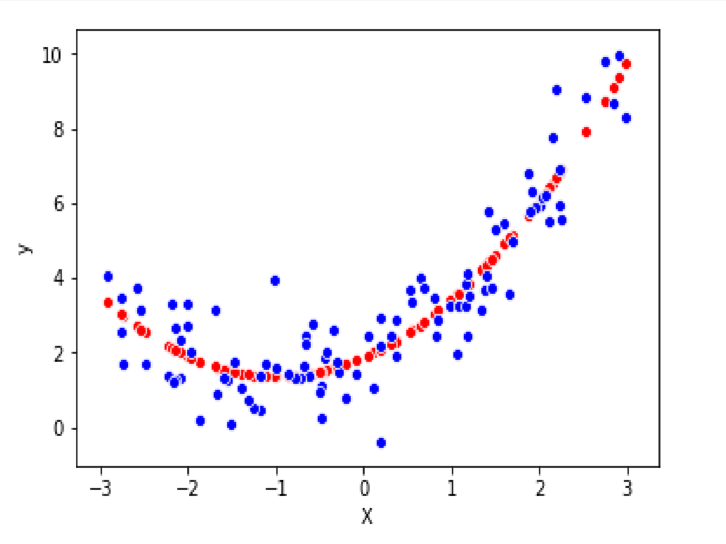
Underfit



Overfit



Good fit



A more sophisticated model tends to have smaller errors in the training set, but can perform worse in testing dataset because it overfit

A too simplistic model will never be able to fit well on both the training set as well as the testing dataset

Will never be able to fit well for BOTH sets.

Bias vs Variance

High Bias = Too Simple of a model

- Bias means your model is intrinsically wrong (off, biased) that you will not fit the data well. If you use a too simplistic model, you will have high bias.

Means your model is too simple and will NOT work with a different dataset

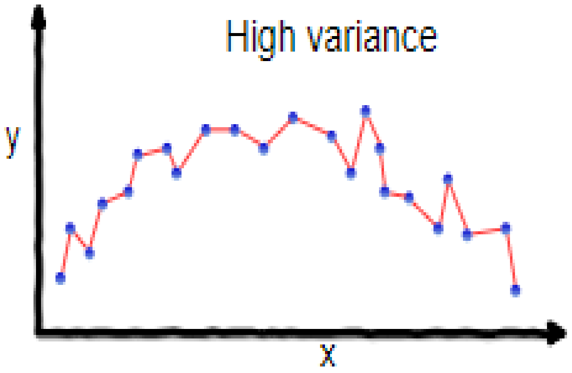
- On the other hand, using a more complicated model, you will have low bias. However, your model will not generalize well to testing dataset (out-of-sample data). The “variance” of your prediction will be high

Low bias = High Variance (Prediction will be all over the place). Meaning model is too complicated

- We call this the Bias vs Variance trade-off

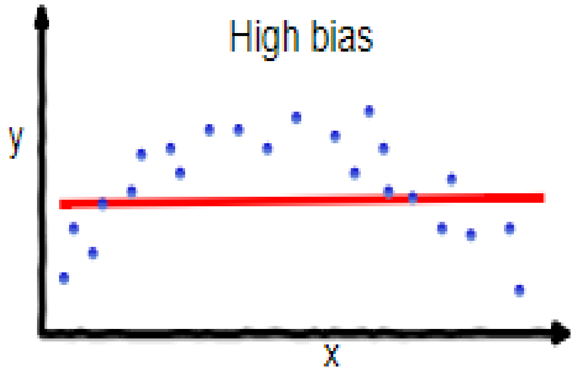
This effect is called the Bias vs Variance trade-off

Bias vs Variance Tradeoff



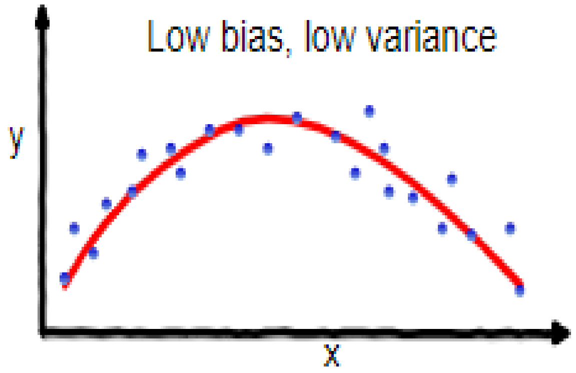
overfitting

High Variance



underfitting

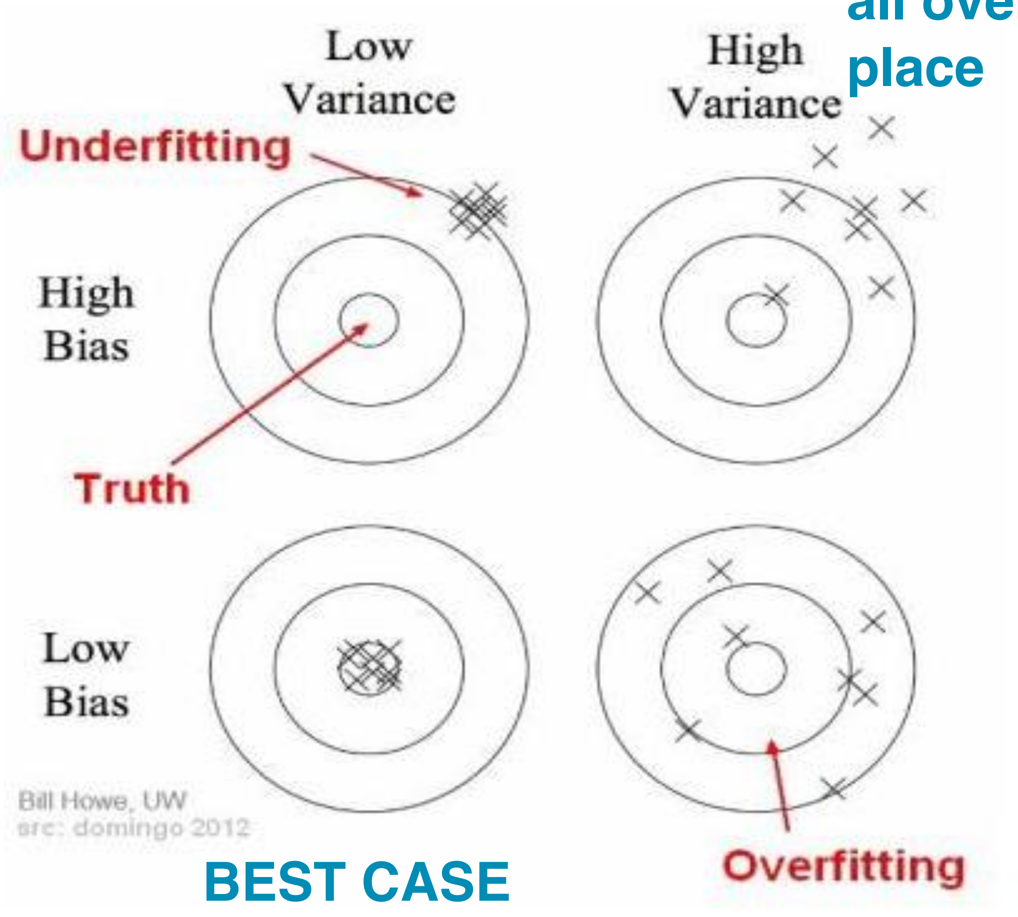
High Bias



Good balance

Low Bias and Low Variance

Bias vs Variance Tradeoff

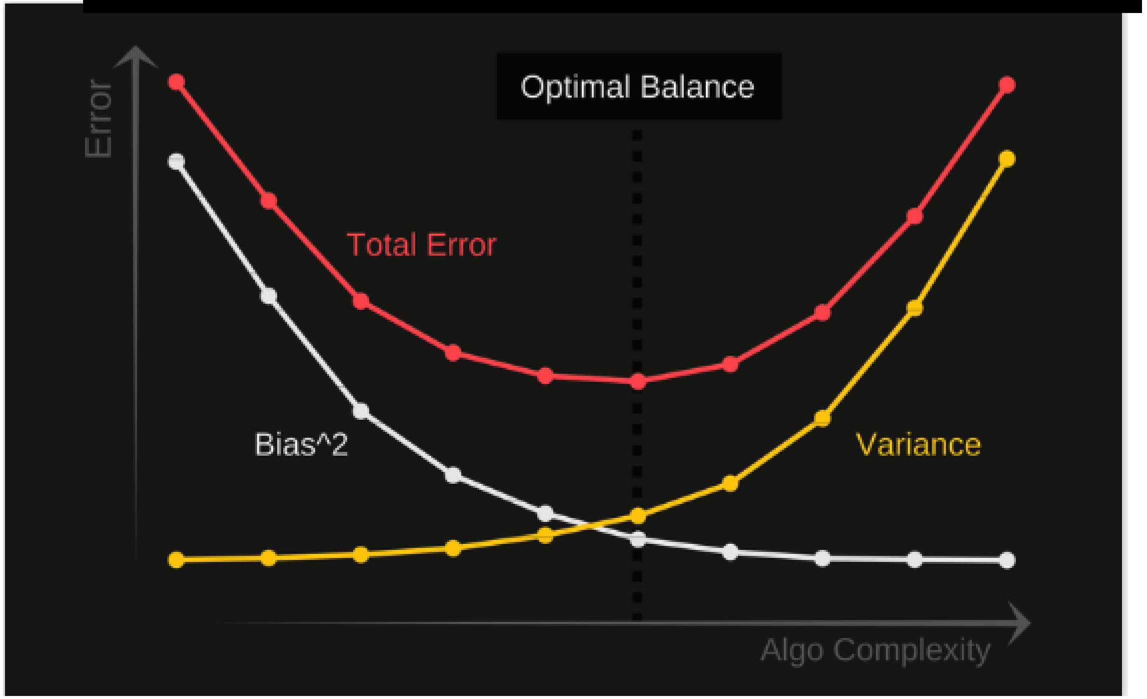


Prediction all over the place

BEST CASE

Total Error = Bias^2 + Variance + Irreducible Error

Shows most optimal type of model
Should be balanced in bias and variance



Recall Linear Regression can still apply to non-linear relationship

Y = a X^2 + b X + c => Y = b1 Z1 + b2 Z2 + b3
where Z1 = X^2 and Z2 = X

N = N_0 exp(-lambda * t)
=> ln (N/N_0) = - lambda * t + c
=> Y = m X + b where Y = ln (N) and X = t

If Y = log (P / (1-P)) = beta_0 + beta_1 * X_1 + beta_2 * X_2 + …. Beta_N + X_N

where P is the probability of something happens

It is called Logistic Regression, which we will cover next

Classification Problem

Linear Regression: Target variable can take any numeric value

Binary Classification Problem: Target variable is either 1 or 0, Yes or No

Multi-class Classification Problem: Target variable is a list of possible values (such as classify a picture of animal as a cat, dog, bird, fish picture)

In classification, your forecast is positive or negative. Your actual is either positive or negative. Count number of cases. True Positive, True Negative. Accuracy = number of (TP + TN) / total # of cases. Confusion Matrix

$\Rightarrow$ NEXT TOPICS

=>Classification Problem and Logistics Regression