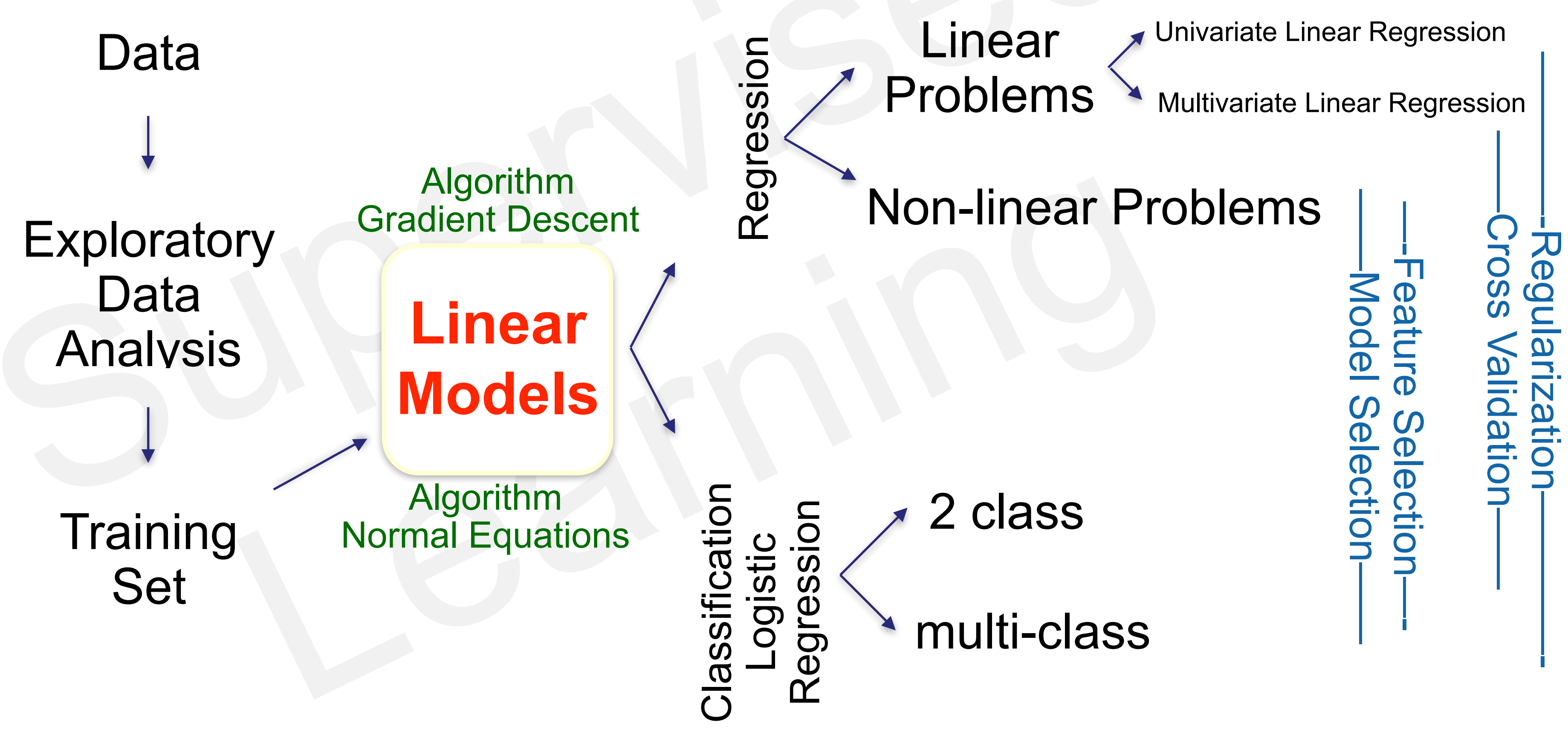
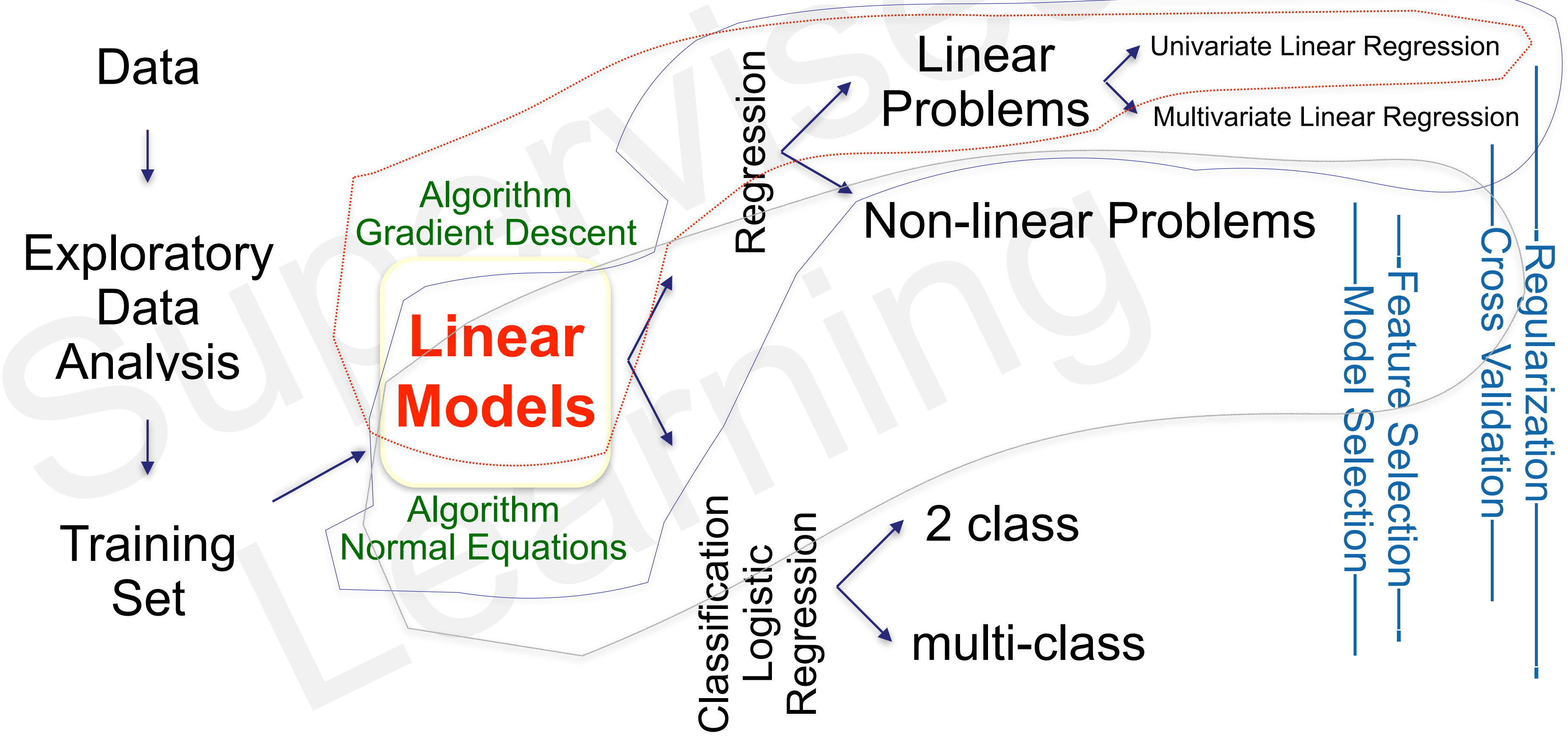
GENERAL ASSEMBLY

INTRO TO DATA SCIENCE
LECTURE 6: LINEAR MODELS & NON-LINEAR FUNCTIONS

WHERE ARE WE ON THE DATA SCIENCE ROAD-MAP?



WHERE ARE WE ON THE DATA SCIENCE ROAD-MAP?



KEY CONCEPTS - LINEAR MODELS FOR NON-LINEAR PROBLEMS

$$y = \theta_0 + \theta_1 x$$

Univariate Linear Regression

$$y = \theta_0 + \theta_1 x_1 + \theta_2 x_2 + \theta_3 x_3 \ldots \theta_n x_n$$

Multivariate Linear Regression

$$y = \theta_0 + \theta_1 z_1 + \theta_2 z_2 + \theta_3 z_3 \ldots \theta_n z_n$$

$$z_1 = x_1$$

$$z_2 = x_2$$

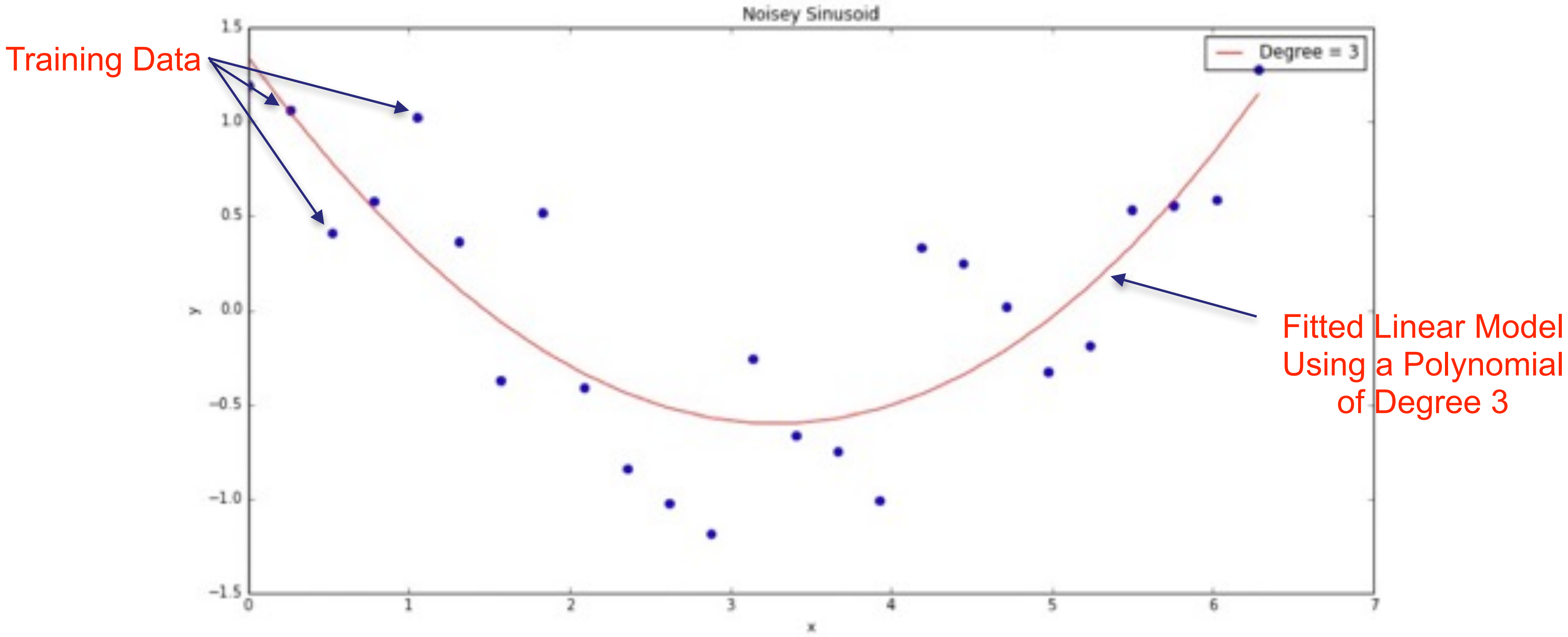
$$z_3 = x_1 x_2$$

$$z_4 = x_1^2$$

$$z_5 = x_2^2$$

This is still linear in the parameters but allows modeling of non-linear combinations of features

Key Concepts - Linear Model for Non-Linear Problem Example



Key Concepts - Creating Non-Linear Features in Python

- Sklearn has ‘PolynomialFeatures()’ in the preprocessing module

- http://scikit-learn.org/dev/modules/preprocessing.html#preprocessing

- Polynomial Features converts an input array into an array consisting of:
  - the original features, e.g. x1, x2
  - interaction terms e.g. x1 * x2
  - power terms e.g. x1 * x1

- There is an option to use only the interaction features

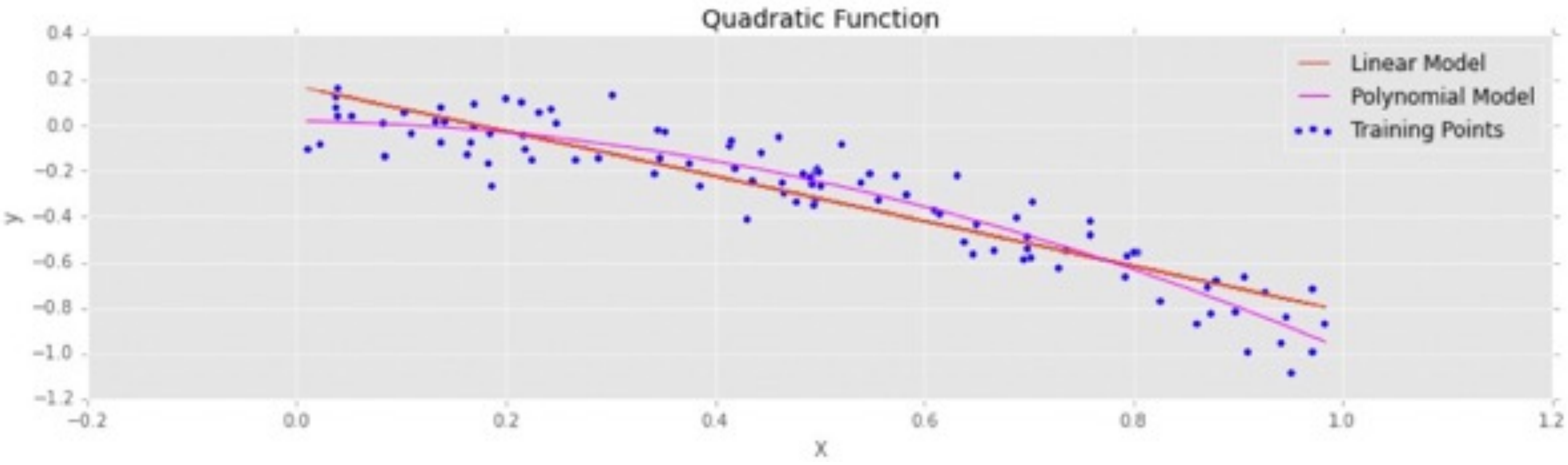
Key Concepts - Creating Non-Linear Features in Python

- Usually this is used in conjunction with the ‘pipeline’
- http://scikit-learn.org/stable/modules/generated/sklearn.pipeline.Pipeline.html#sklearn.pipeline.Pipeline
- In general I tend to use the ‘make_pipeline’ function, to combine preprocessing and a linear model into a consolidated model
  - e.g. make_pipeline(PolynomialFeatures(degree_of_polynomial_required), LinearRegression())

KEY CONCEPTS - LINEAR MODEL FOR NON-LINEAR PROBLEM EXAMPLE

Random points generated using a quadratic function and added noise

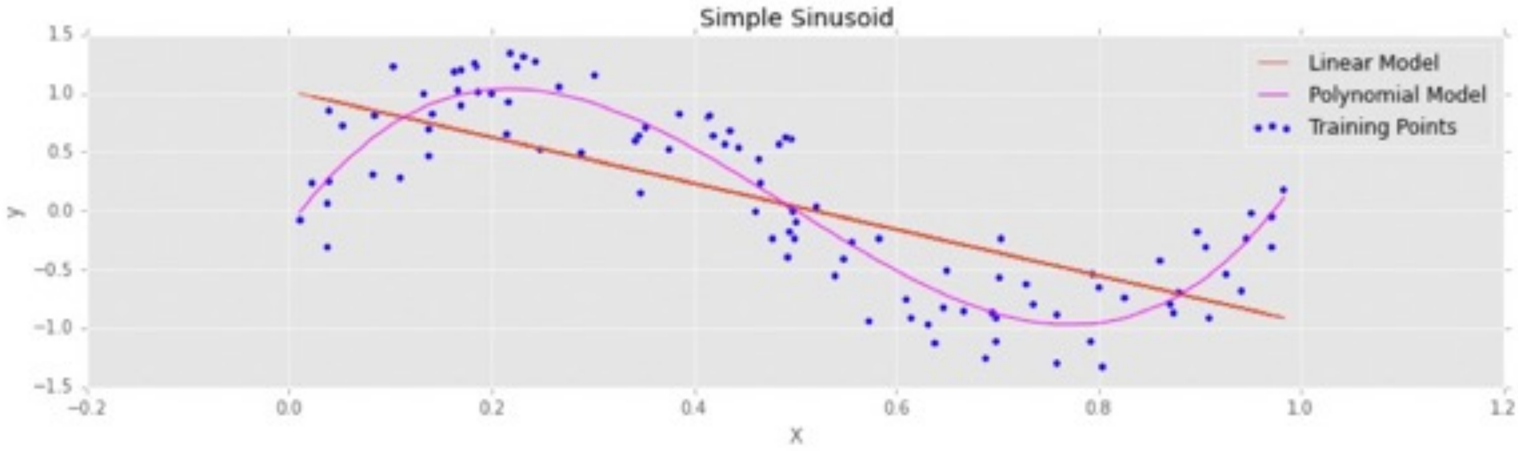
Ideal polynomial to fit is of degree 2



KEY CONCEPTS - LINEAR MODEL FOR NON-LINEAR PROBLEM EXAMPLE

Random points generated using a simple sinusoid and added noise

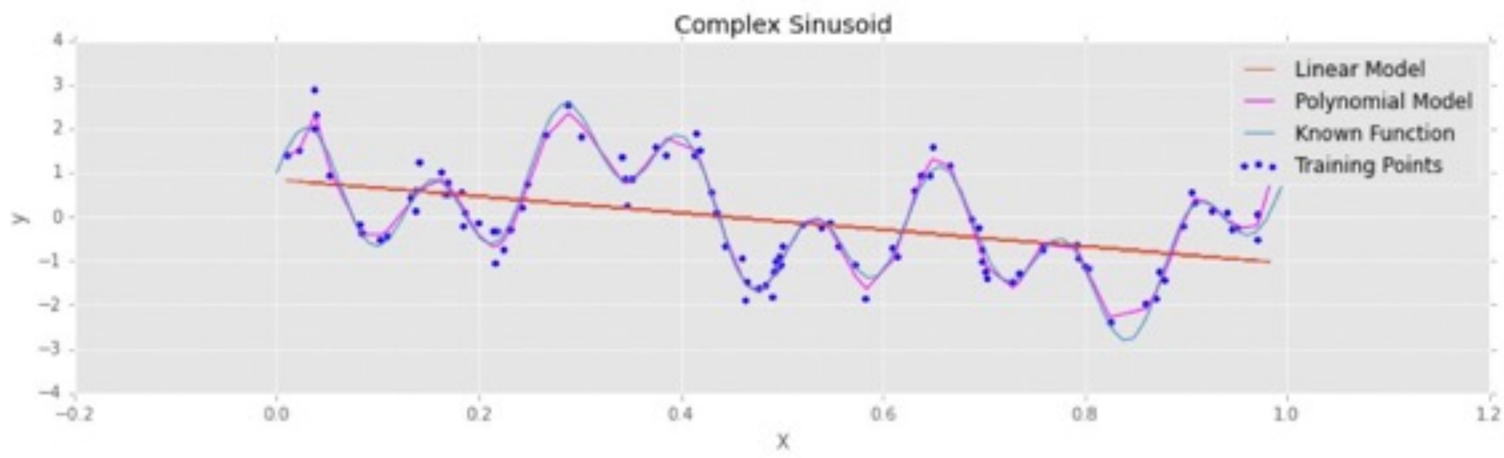
Ideal polynomial to fit is of degree 3



KEY CONCEPTS - LINEAR MODEL FOR NON-LINEAR PROBLEM EXAMPLE

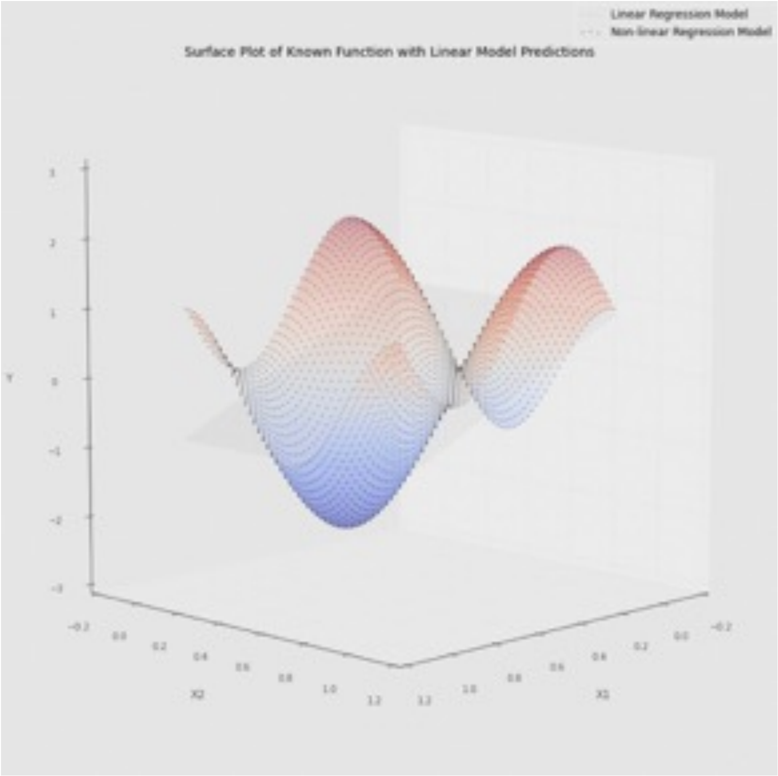
Random points generated using a complex sinusoid and added noise

Ideal polynomial to fit is of degree ?



Multivariate Regression

A 2D polynomial with a fit using multivariate
linear regression and non-linear regression



Model Selection

- To (hopefully) improve our model we add non-linear components
  - e.g.1 House price might be related to the square of the number of baths
  - e.g.2 In a polynomial regression we have x1 * x1, x2 * x2 terms
- In polynomial regression we determine the complexity of the model by the degree of the polynomial

Feature Selection

- Choosing which input features can be successfully modeled to outputs
  - e.g.1 What features of the a house optimally predict it's price
  - e.g.2 In a polynomial regression problem there is just x1, x2, …, xN

Key Concepts - Model Selection and the Concept of Fit

“Essentially, all models are wrong, but some are useful”

George Box, statistician

(and who has been referred to as one of the greatest statistical minds of the 20th Century

Key Concepts - How Do We Test Fit?

- One way would be to measure the mean squared error
  - Take the data
  - Fit the model
  - Train the model until a suitable reduction in MSE is achieved

- The problem with this is that the model will be ‘tuned’ to the dataset it was trained on and may not ‘generalize’ well

- In the next class we will discuss this in great detail

KEY CONCEPTS - HOW DO WE TEST FIT?

- What does it mean to ‘generalize’?
- A machine learning algorithm generalizes well when it performs well on unseen data
- Objectively this can be measured using the cost function when the ML algorithm is run on <u>unseen data</u>
- In order to achieve this we must first partition the data

KEY CONCEPTS - TRAINING, VALIDATION, TEST SET

- Data Partitioning

- Ideally we have enough data to divide the dataset into 3 subsets,
  1. training set,
  2. validation set and
  3. test set.

- If the original dataset is large these may be 3 equal subsets

- Often, however, they may be equal in size

KEY CONCEPTS - TRAINING, VALIDATION, TEST SET

- The training set is used to train the model, BUT

- Feature & Model selection are optimized by the model's performance on the validation set
  - because we want the model to generalize

- This process is called <u>validation</u> or <u>cross validation</u>

- The test set is used to test the final model and report results
- The test set is not used to influence model construction in any way

KEY CONCEPTS - TRAINING, VALIDATION, TEST SET

- A common data science community 'belief' is that most practitioners do not worry about an independent 3rd set

- The Python data science stack is, as you will see, designed around a partition into only 2 sets - training and test
- http://scikit-learn.org/stable/modules/generated/sklearn.cross_validation.train_test_split.html#sklearn.cross_validation.train_test_split

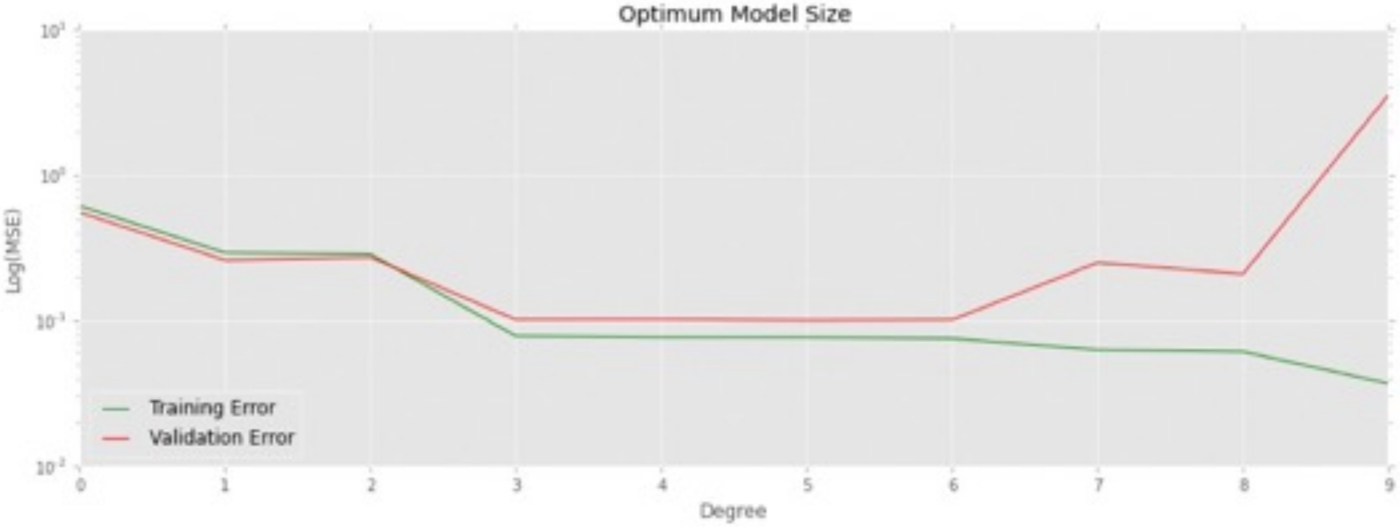
- The story goes that most people train using the training set, and optimize their models using the validation set, and report their results using that same validation set

- OK for class… but beware…

KEY CONCEPTS - TRAINING, VALIDATION, TEST SET

- If the dataset is small however, equal separation into 3 groups maybe impossible.
- The training set usually contains the largest proportion of the data, e.g.60%
- The validation set usually contains the majority of the remaining data, e.g.20%
- The test often contains the remainder of the data, ‘just-enough’ e.g.20%

Key Concepts - Linear Model for Non-Linear Problem Example



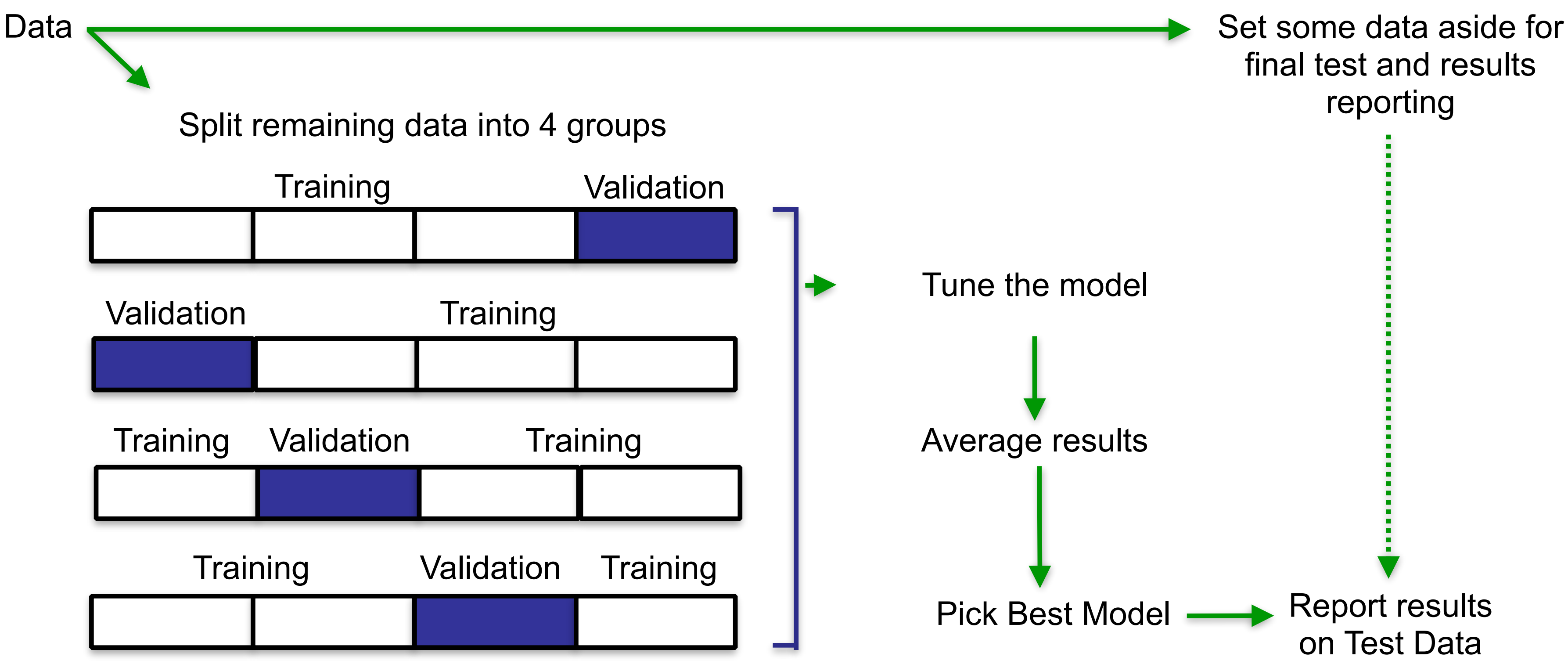
Key Concepts - Cross Validation

- What happens when data is scarce?
- We would still like to use the majority of the data to train our model on
- However, we need a way to tune the model using some ‘independent’ data that the model has not been exposed to
- …and some data should be held back for an independent test from which to report results

KEY CONCEPTS - S-FOLD CROSS-VALIDATION

- Split the training data in to ’S’ folds (e.g. S=4, 4-fold cross-validation)
- Use the 3 groups of data to train, 1 to validate
- Repeat, swapping the groups around
- Average the performance
- In the case of 4-fold cross validation there will be 4 models, each trained on 3/4 of the data and each tested on the 1/4 of the data that was held out. Held-out data being different each time

Key Concepts - Cross-validation



Key Concepts - Other Ways to Assess Fit

- Historically various ‘information criteria’ have been proposed in an attempt to assess model fit without the need for validation. Two of the most well know are:

1. Akaike Information Criteria (AIC)

2. Bayesian Information Criteria (BIC)

- In general they tend to favor overly simple models.

Key Concepts - Statsmodels

- For those seeking more statistical information on model fit there is a python library called ‘Statsmodels’

- http://statsmodels.sourceforge.net/

OLS Regression Results

Dep. Variable:	y	**R-squared:**	0.858
Model:	OLS	**Adj. R-squared:**	0.858
Method:	Least Squares	**F-statistic:**	3015.
Date:	Wed, 08 Oct 2014	**Prob (F-statistic):**	1.19e-213
Time:	18:57:43	**Log-Likelihood:**	-867.25
No. Observations:	500	**AIC:**	1736.
Df Residuals:	499	**BIC:**	1741.
Df Model:	1		

	coef	**std err**	**t**	**P>|t|**	**[95.0% Conf. Int.]**
x1	5.8530	0.107	54.911	0.000	5.644 6.062

TO BE CONTINUED…