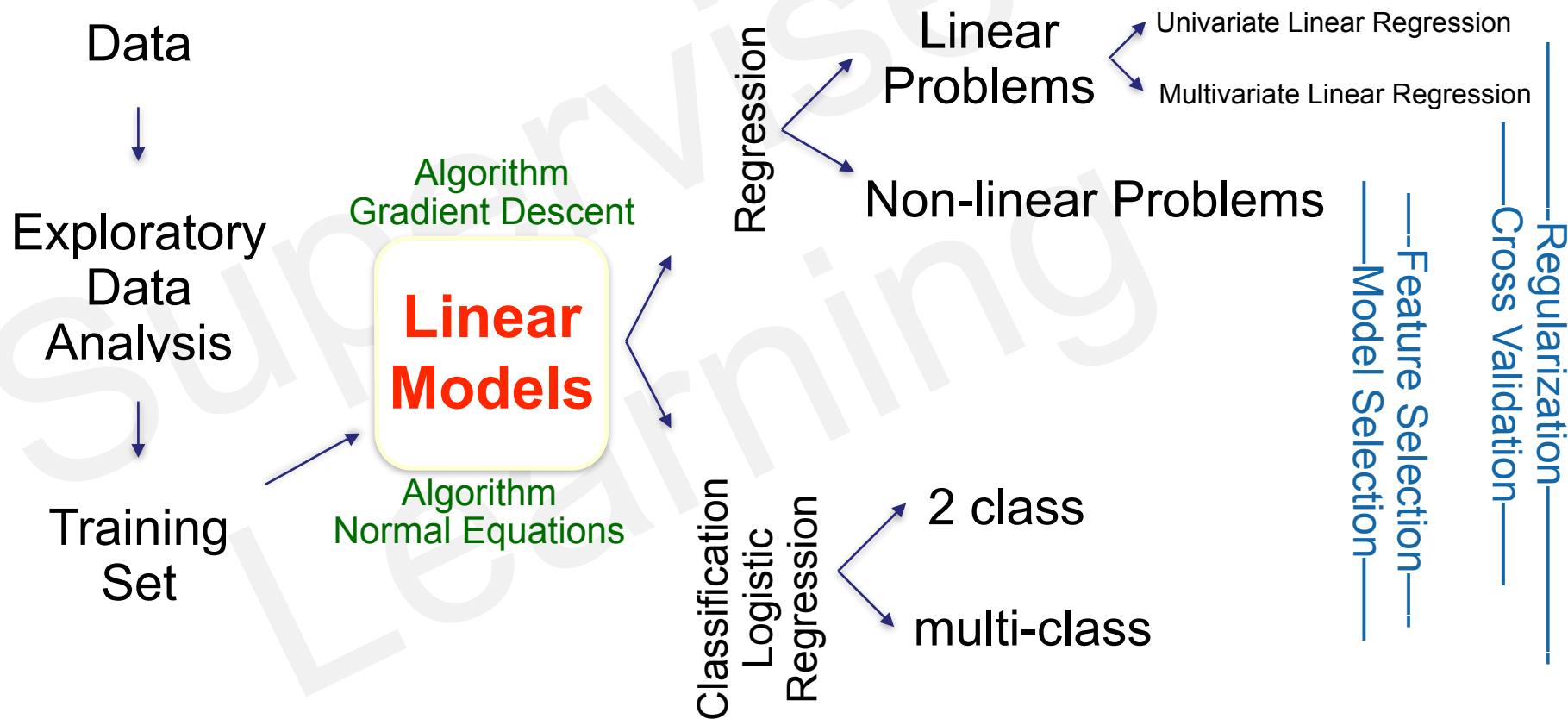


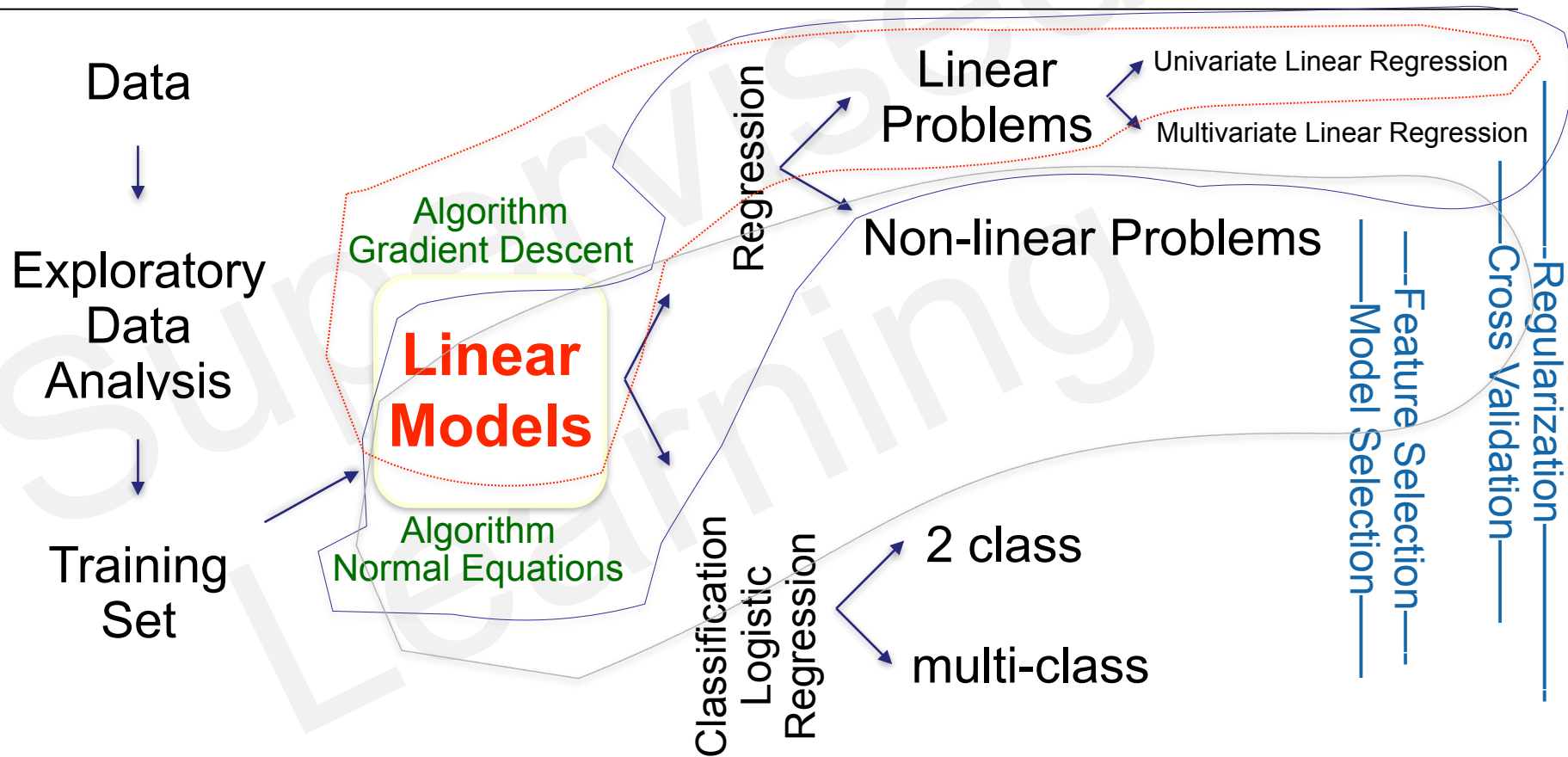
# INTRO TO DATA SCIENCE

## 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

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

Multivariate Linear Regression

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

$$z_1 = x_1$$

$$z_2 = x_2$$

$$z_3 = x_1 x_2$$

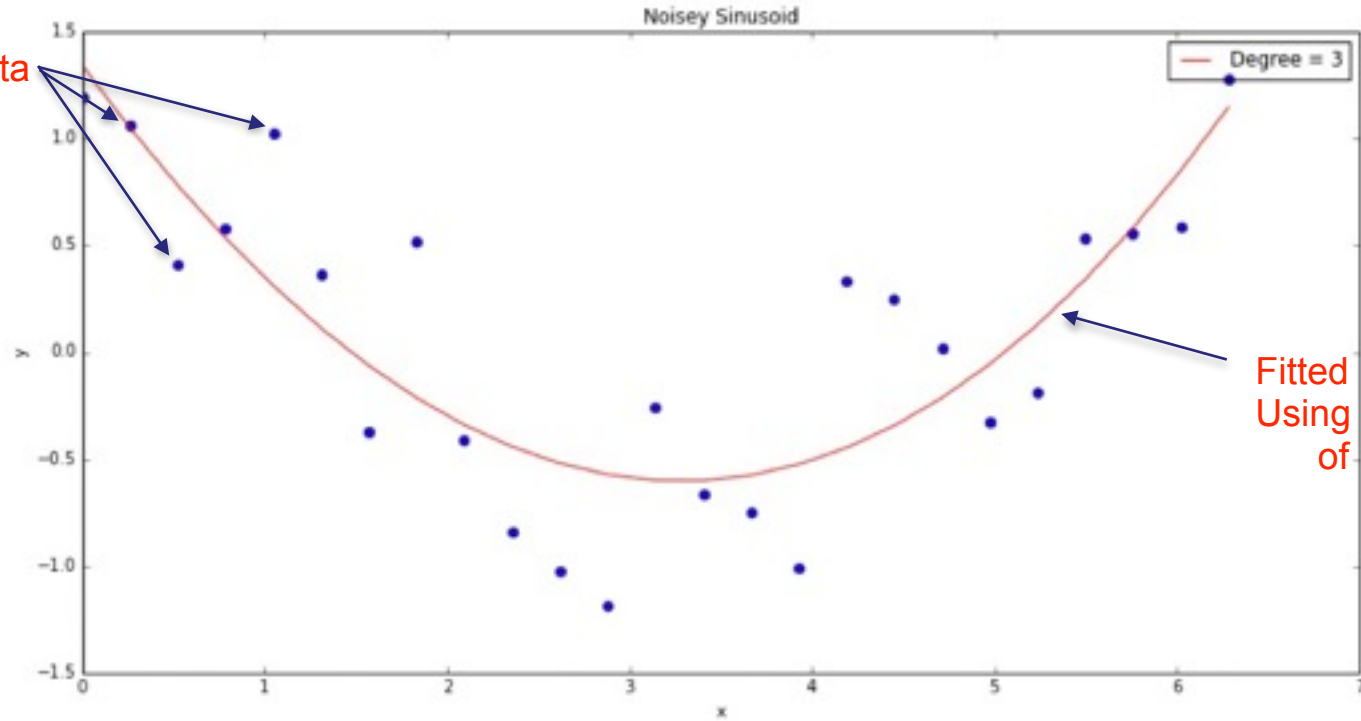
$$z_4 = x_1^2$$

$$z_5 = x_2^2$$

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

Training Data



Fitted Linear Model  
Using a Polynomial  
of Degree 3

---

## 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.  $x_1$ ,  $x_2$
  - interaction terms e.g.  $x_1 * x_2$
  - power terms e.g.  $x_1 * x_1$
- 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())`

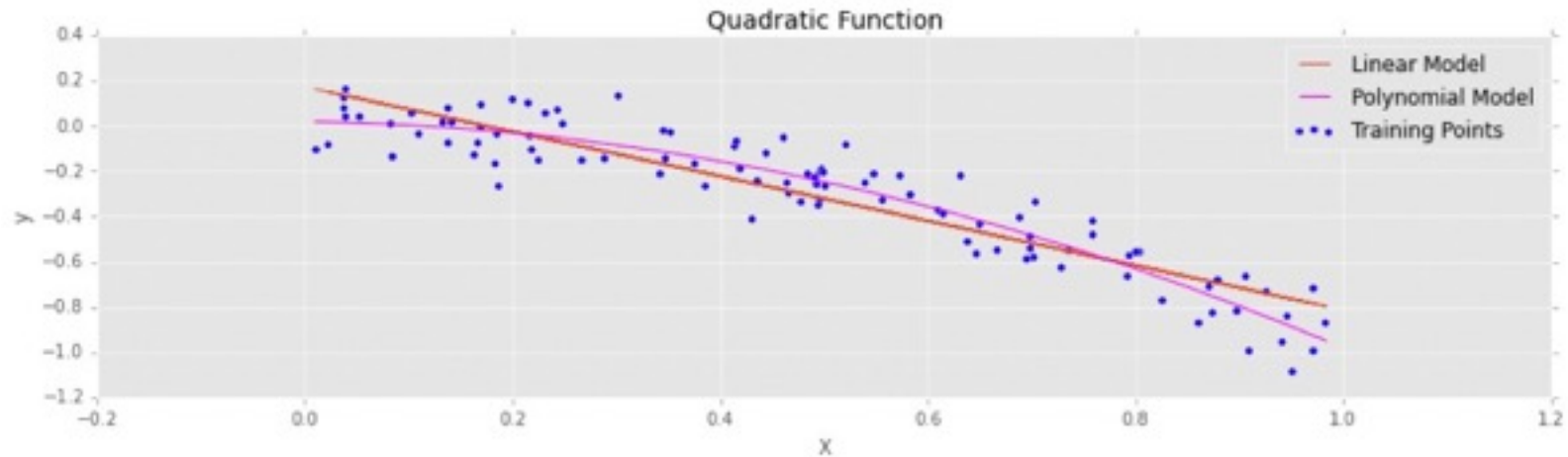
---

## 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





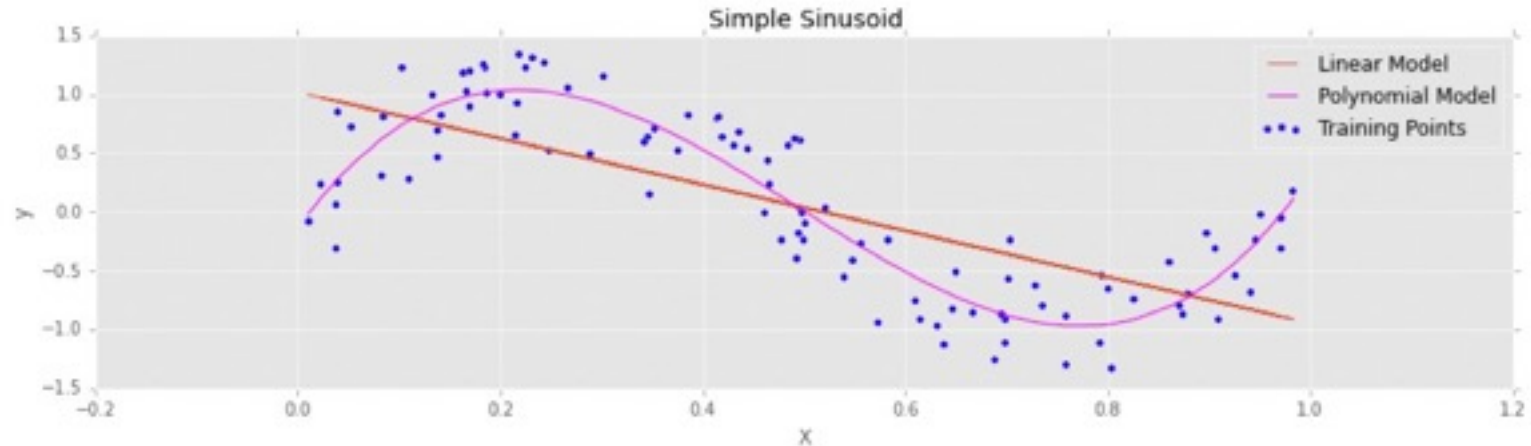
---

## 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



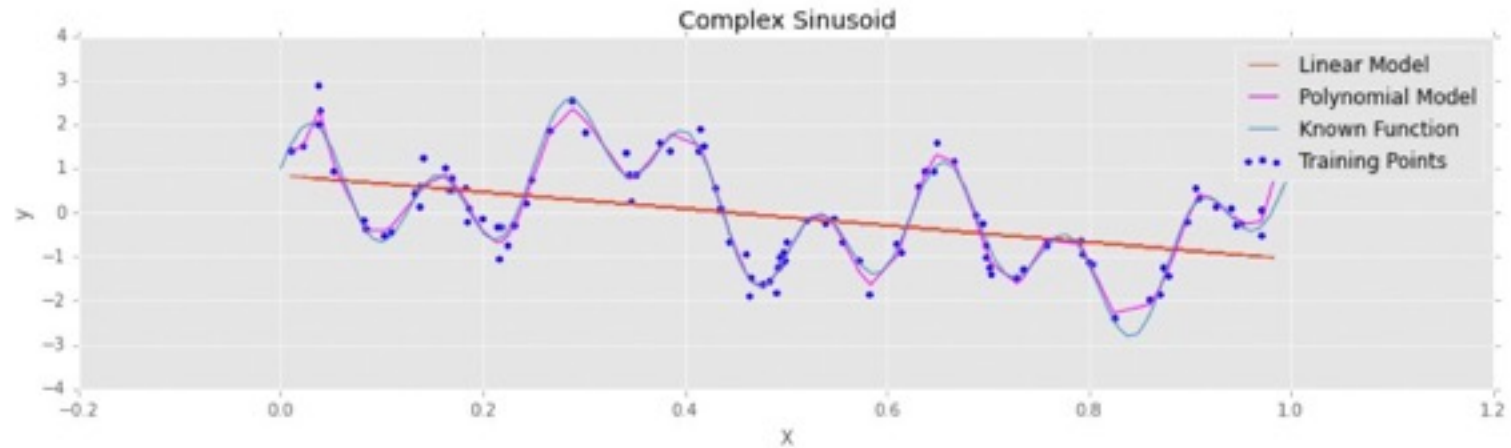
---

## 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 ?



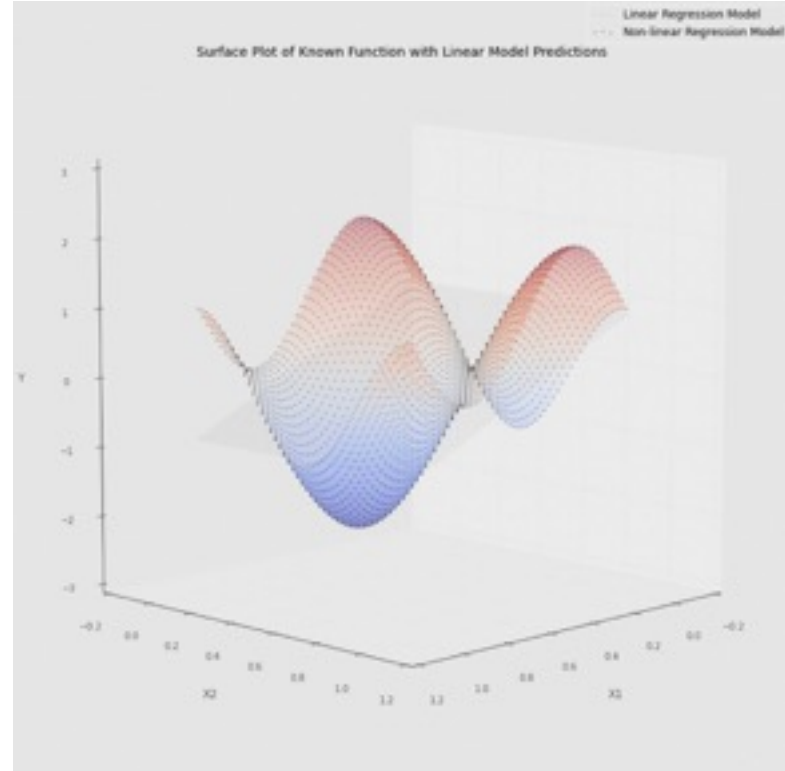
---

## KEY CONCEPTS - LINEAR MODEL FOR NON-LINEAR PROBLEM EXAMPLE

---

### 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  $x_1 * x_1$ ,  $x_2 * x_2$  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  $x_1, x_2, \dots, x_N$

---

## 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 unseen data
- 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 validation or cross validation
- 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](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

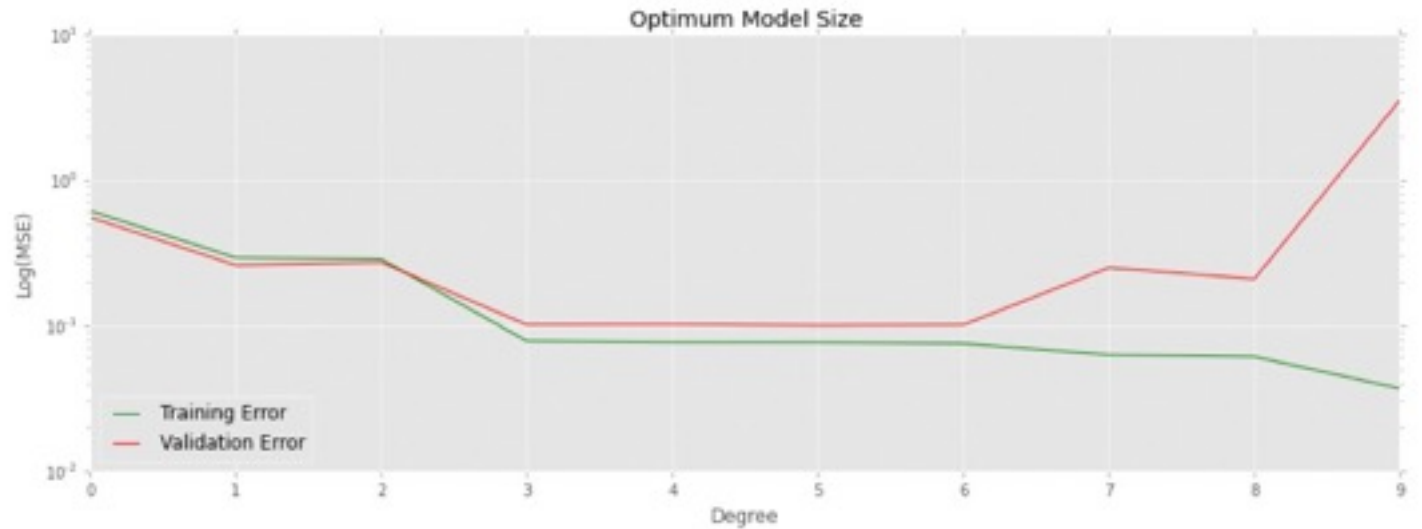
---

- 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

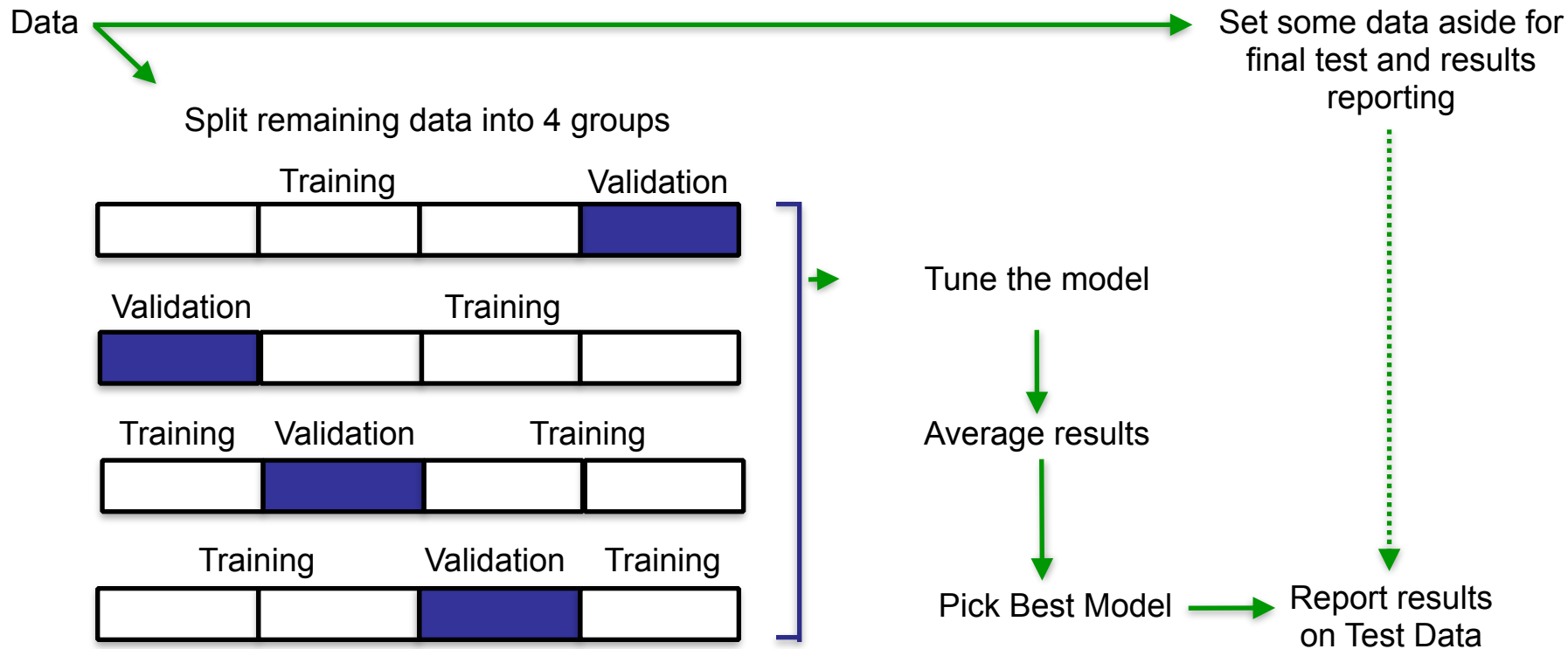
---

## KEY CONCEPTS - S-FOLD CROSS-VALIDATION

---

- Split the training data in to 'S' folds (e.g.  $S=4$ , 4-fold cross-validation)
  - Sklearn - K folds
- 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/>

---

## ASSESSING MODEL FIT: *STATSMODELS*

---

### OLS Regression Results

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

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

---

## KEY CONCEPTS - FEATURE SELECTION

---

TO BE CONTINUED...