GENERAL ASSEMBLY

INTRO TO DATA SCIENCE
MODEL SELECTION

Assessing Model Accuracy

Multiple linear regression

Some important questions

Deciding on the right variables

Cross validation

Assessing Model Accuracy

What is a model? What is a "good" model?

Assessing Model Accuracy

What is a model? What is a “good” model?

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

-George Box

Assessing Model Accuracy

What is a model? What is a "good" model?

Suppose we fit a model $f(x)$ to some training data...

What is a model? What is a "good" model?

Suppose we fit a model *f(x)* to some training data

$$\mathrm{MSE}_{\mathrm{train}} = \mathrm{Ave}_{i \in \mathrm{train}}[y_i - f(x_i)]^2$$

What is a model? What is a "good" model?

Suppose we fit a model *f(x)* to some training data

$$MSE_{train} = Ave_{i \in train}[y_i - f(x_i)]^2$$

may be biased toward overfitted models….

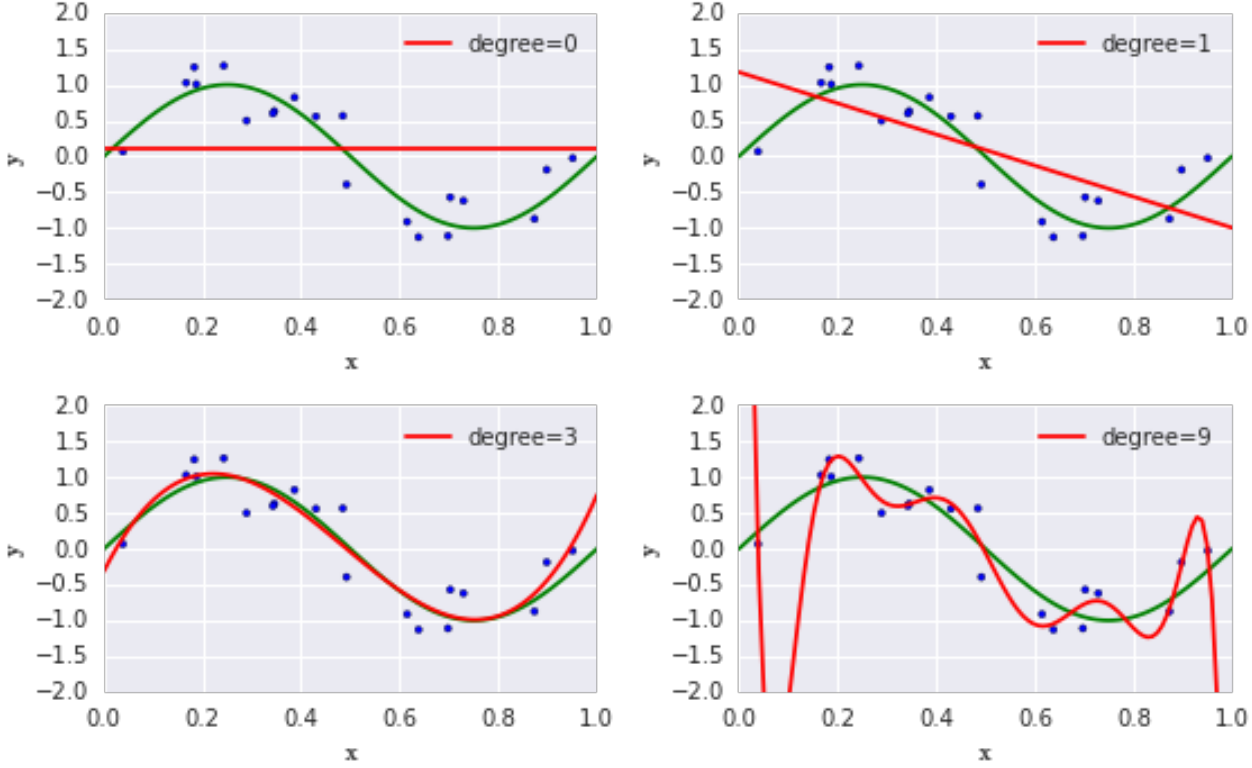
WHAT IS A MODEL? WHAT IS A "GOOD" MODEL?

So, we use a ***train*** and ***test*** set and minimize errors for the test set

$$\mathrm{MSE}_{\mathrm{test}} = \mathrm{Ave}_{\mathrm{i} \in \mathrm{test}}[\mathrm{y}_\mathrm{i} - \mathrm{f}(\mathrm{x}_\mathrm{i})]^2$$

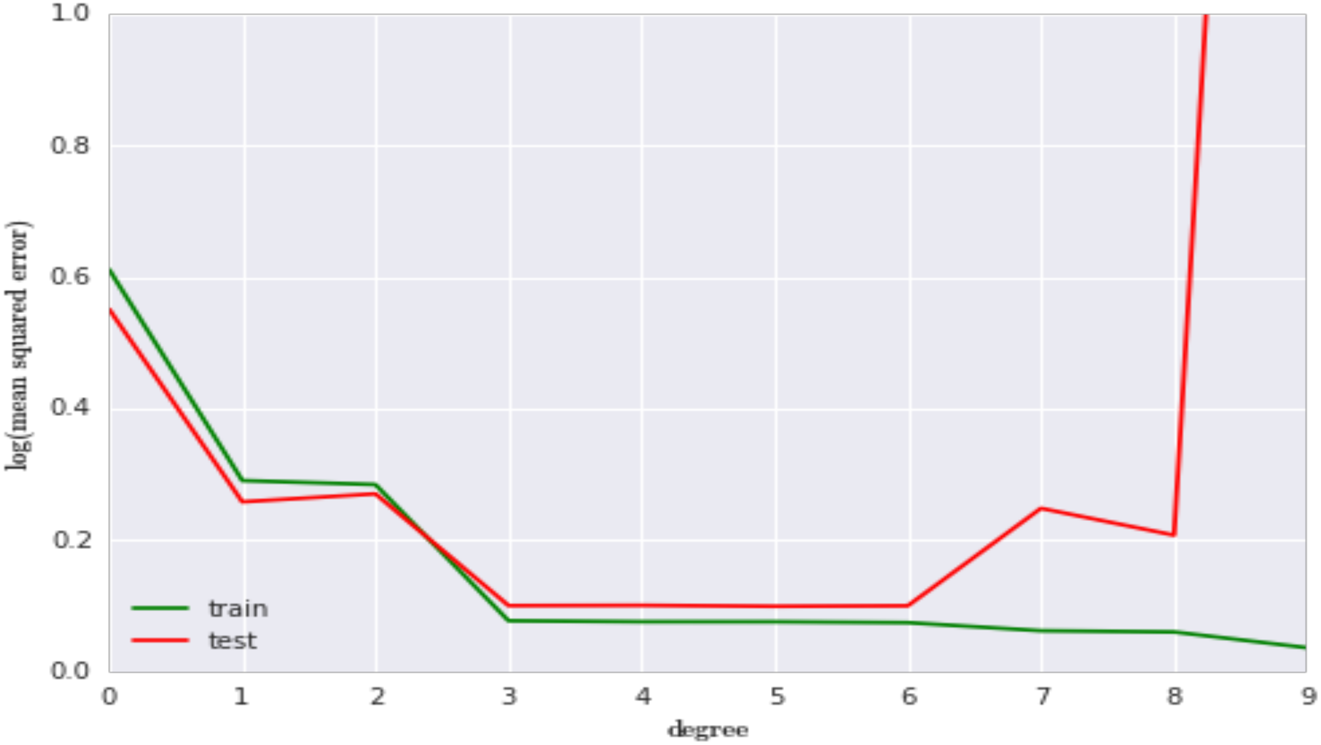
Assessing Model Accuracy

How to decide among multiple models?



Assessing Model Accuracy

How to decide among multiple models?

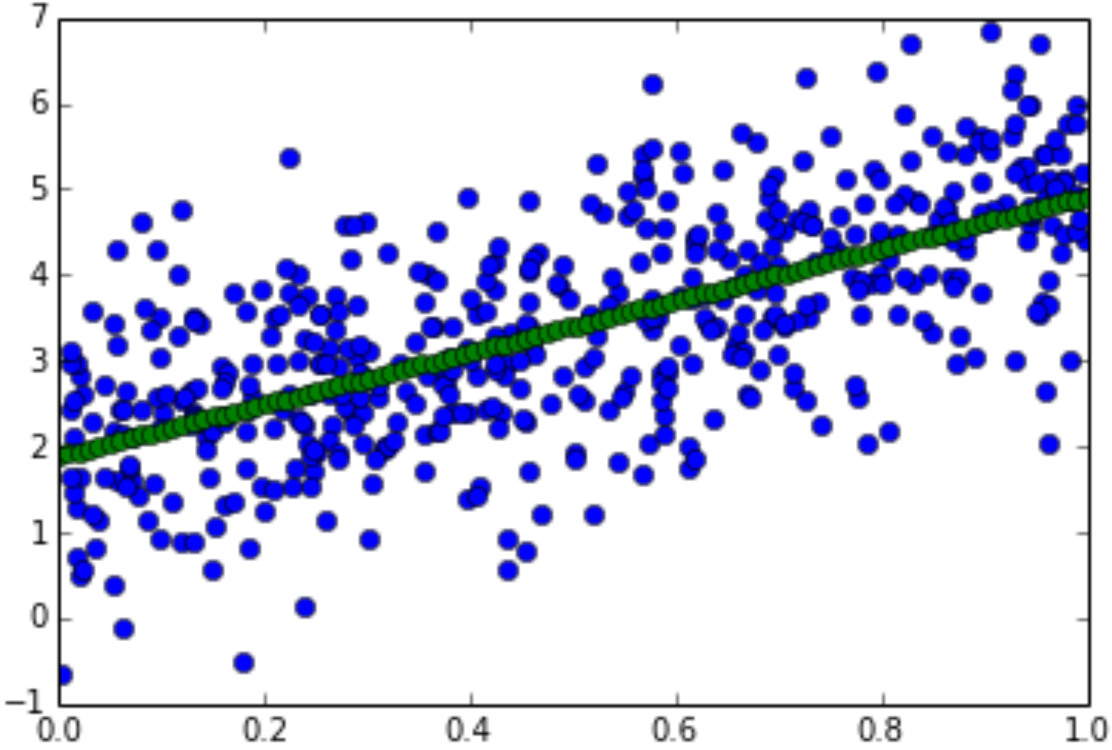


ASSESSING MODEL ACCURACY

In [3]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
from sklearn.linear_model import LinearRegression
np.random.seed(0)
samples = 500
X = np.random.random(size=(samples, 1)) # The size of the input array is expected to be [n_samp
y = 2 + 3 * X.squeeze() + np.random.normal(size=samples)
model = LinearRegression(fit_intercept=True)
model.fit(X, y)
print ("Model coefficient: %.5f, and intercept: %.5f" % (model.coef_, model.intercept_))
X_test = np.linspace(0, 1, 100).reshape(100,1)
y_hat = model.predict(X_test)
plt.plot(X,y,'o')
plt.plot(X_test, y_hat, 'o')
```

```
Model coefficient: 3.02895, and intercept: 1.88551
```

ASSESSING MODEL ACCURACY



ASSESSING MODEL ACCURACY: *STATSMODELS*

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

MULTIPLE LINEAR REGRESSION

Recall the model for Simple Linear Regression:

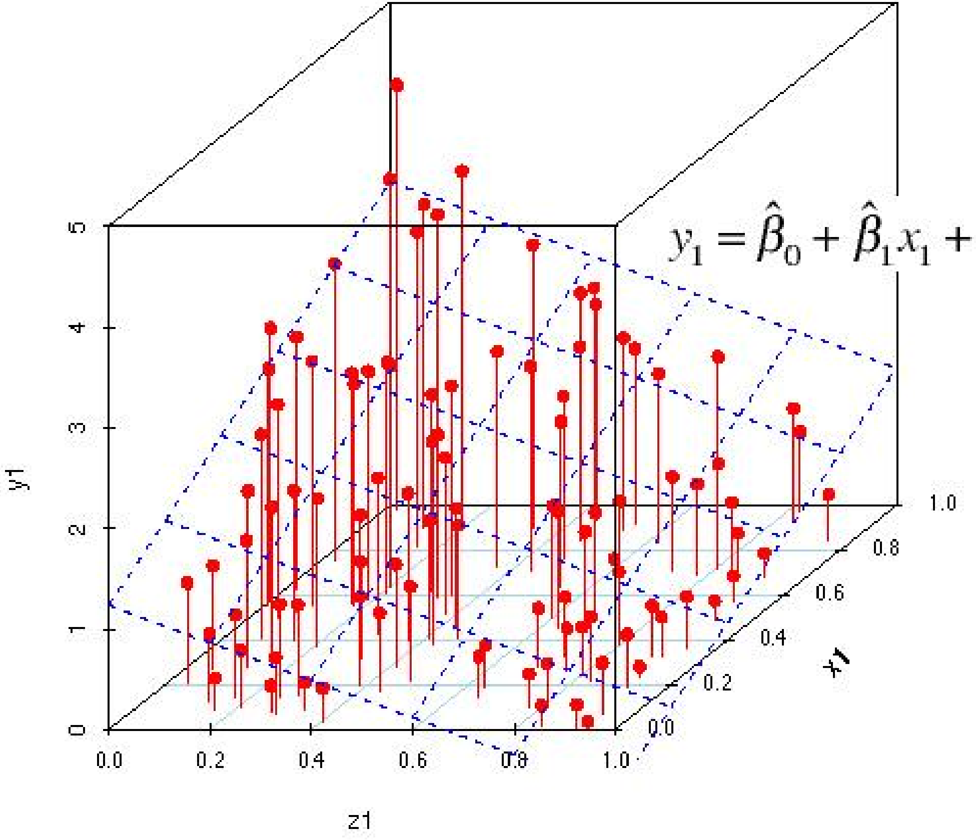
$$y = \alpha + \beta x$$

MULTIPLE LINEAR REGRESSION

For Multiple Linear Regression, this becomes:

$$y = \alpha + \beta_1 x_1 + \beta_2 x_2 + \cdots + \beta_n x_n$$

MULTIPLE LINEAR REGRESSION



$$y_1 = \hat{\beta}_0 + \hat{\beta}_1 x_1 + \hat{\beta}_2 z_1$$

MULTIPLE LINEAR REGRESSION:

INTERPRETING REGRESSION COEFFICIENTS

- Ideal scenario is when the predictors are uncorrelated (a balanced design)
    - Each coefficient can be estimated and tested separately.
    - Interpretations are possible like "a unit change in $\boldsymbol{X_i}$ is associated with a $\boldsymbol{\beta_i}$ change in $\boldsymbol{Y}$, while all the other variables stay fixed"
- Correlations among predictors cause problems:
    - variance of all coefficients tends to increase, sometimes dramatically
    - Interpretations become hazardous — when $\boldsymbol{X_i}$ changes everything else changes

SOME IMPORTANT QUESTIONS

1. Is at least one of the predictors $X_1, X_2, \ldots, X_p$ useful in predicting the response?
2. Do all the predictors help to explain Y , or is only a subset of the predictors useful?
3. How well does the model fit the test data?
4. Given a set of predictor values, what response value should we predict, and how accurate is our prediction?

Cross validation: motivation

Cross validation can be useful for:

- Mitigating against overfitting
- Evaluating and comparing the performances of different models for prediction
- Selecting subsets of model parameters
- Generating test/train sets when the original dataset is too small

CROSS VALIDATION

Features of k-fold cross-validation:

- More accurate estimate of prediction error
- More efficient use of data than single train/test split
  - Each record in our dataset is used for both training and testing
- Presents tradeoff between efficiency and computational expense
  - 10-fold CV is 10x more expensive than a single train/test split
- Can be used for model selection

CROSS VALIDATION

Steps for k-fold cross-validation:

1. Randomly split the dataset into n equal partitions
2. Use partition 1 as test set & union of other partitions as training set
3. Find generalization error
4. Repeat steps 2-3 using a different partition as the test set at each iteration
5. Take the average generalization error as the estimate of accuracy