CAB420: Overfitting and Linear Regression

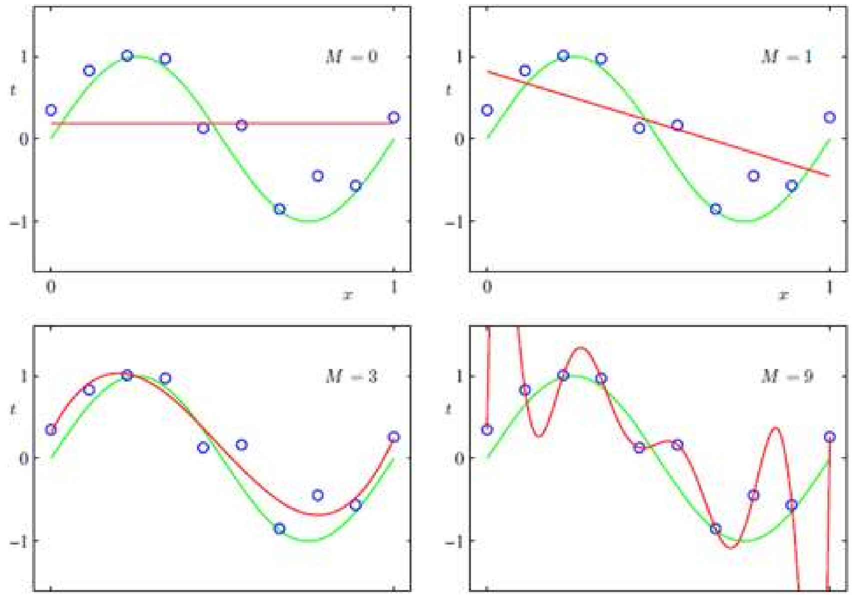
WHAT IS IT? AND WHY DO I CARE?

Overfitting and Regression

- Consider a multi-variate linear regression task
- We can (usually) make the model more accurate on the test set by adding more terms
  - Additional variables
  - Higher order terms

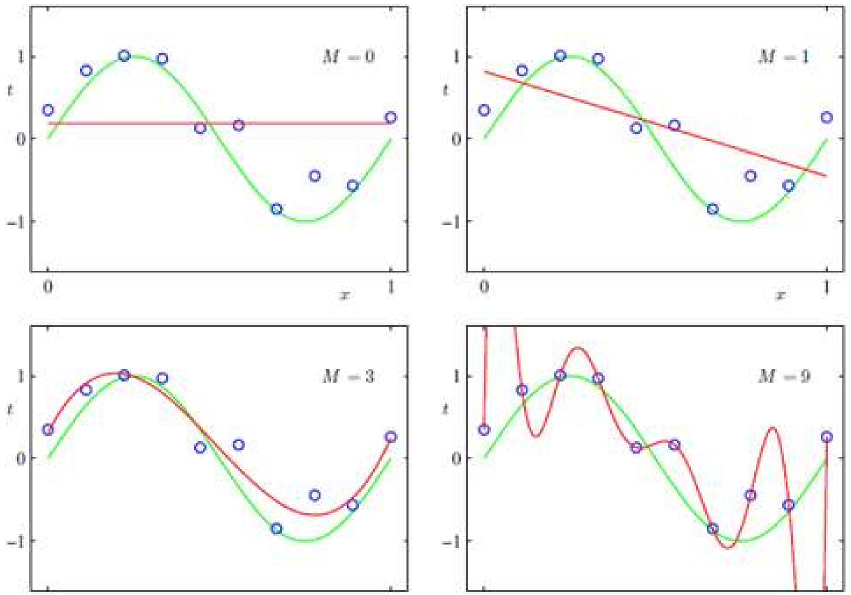
Overfitting and Regression

- On the right we have:
  - A sine wave in green, which has been sampled
  - Samples have been offset by noise
  - We seek to fit a curve (in red) to the sampled data



Overfitting and Regression

- M=9 (9th order polynomial) offers the best fit to the data
  - Hits all the points almost perfectly
- M=3 actually captures the function the best
  - Some error in predictions
  - Overall shape correct however
- Consider, how would M=9 and M=3 perform on a new set of points?
  - Which one would look more correct?



Detecting Overfitting

- We cannot observe overfitting using the training set alone
  - Validation and testing sets are required

- Performance will likely always increase on the training set
  - Need to evaluate performance on other data held out of training
    - Validation data, Testing data

- Often referred to as testing if a model **generalises to unseen data**

Overfitting in Practice

- See ***CAB420_Regression_Example_2_Regularised_Regression.ipynb***

- Demo Overview
  - Load traffic data from Brisbane which contains average travel times between key points on the road network
  - We'll consider the first 9 data series and time of day
    - First 8 series as predictors, with the hour as a categorical
    - 9th series is the response
  - Apply linear regression to data, increase complexity and observe results

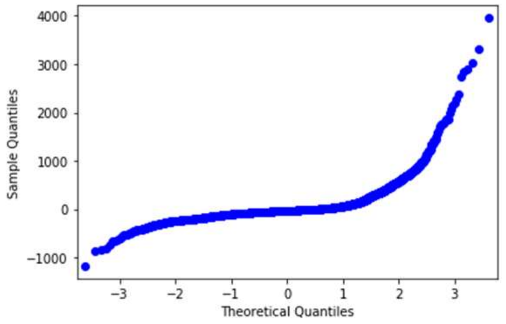
Simple Linear Model

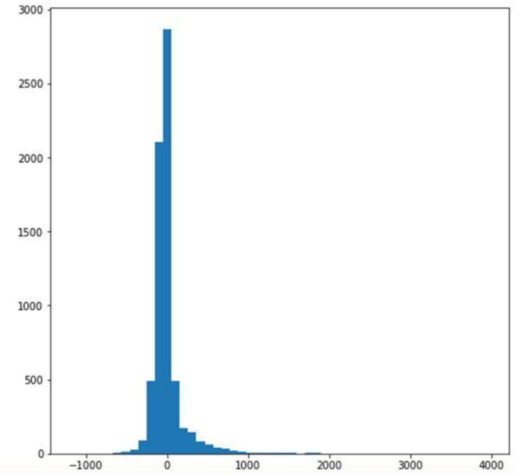
(linear terms with hour of day categorical term)

OLS Regression Results

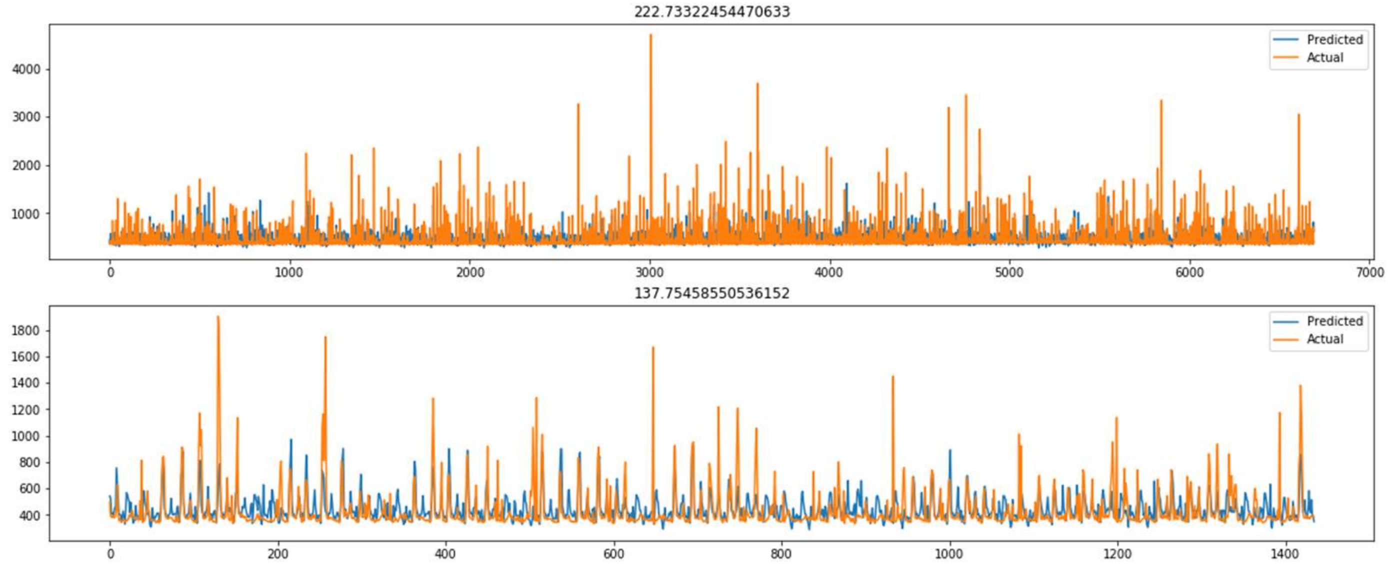
Dep. Variable:	x_1260__1261_	R-squared:	0.256
Model:	OLS	Adj. R-squared:	0.253
Method:	Least Squares	F-statistic:	73.95
Date:	Wed, 13 Jan 2021	Prob (F-statistic):	0.00
Time:	20:03:40	Log-Likelihood:	-45686.
No. Observations:	6694	AIC:	9.144e+04
Df Residuals:	6662	BIC:	9.165e+04
Df Model:	31		
Covariance Type:	nonrobust		

	coef	std err	t	P>|t|	[0.025	0.975]
const	119.9389	26.334	4.555	0.000	68.316	171.562
x_1098__1056_	0.0462	0.057	0.804	0.422	-0.066	0.159
x_1058__1059_	0.2443	0.082	2.980	0.003	0.084	0.405
x_1057__1056_	2.7346	0.167	16.356	0.000	2.407	3.062
x_1017__1007_	0.2636	0.048	5.492	0.000	0.169	0.358
x_1115__1015_	1.3343	0.161	8.268	0.000	1.018	1.651
x_1015__1115_	0.1983	0.272	0.730	0.466	-0.334	0.731
x_1103__1061_	-0.1058	0.134	-0.792	0.429	-0.368	0.156
x_1135__1231_	0.7734	0.112	6.891	0.000	0.553	0.993
1	-50.3136	38.054	-1.322	0.186	-124.912	24.284
2	33.6517	43.325	0.777	0.437	-51.279	118.583
3	-41.9974	27.007	-1.555	0.120	-94.940	10.945
4	-70.6742	24.455	-2.890	0.004	-118.614	-22.734
5	-5.7150	24.002	-0.238	0.812	-52.767	41.337
6	128.0683	24.261	5.279	0.000	80.510	175.627
7	115.4191	24.745	4.664	0.000	66.912	163.927
8	52.9131	24.839	2.130	0.033	4.221	101.605
9	2.7002	24.065	0.112	0.911	-44.475	49.876
10	-86.9350	24.414	-3.561	0.000	-134.793	-39.077
11	-94.3841	24.791	-3.807	0.000	-142.982	-45.786
12	-121.3032	25.006	-4.851	0.000	-170.323	-72.283
13	-126.5093	25.039	-5.053	0.000	-175.593	-77.425
14	-165.4192	25.107	-6.588	0.000	-214.638	-116.201
15	-188.6012	26.228	-7.191	0.000	-240.016	-137.186
16	-153.1972	26.073	-5.876	0.000	-204.309	-102.086





Simple Linear Model



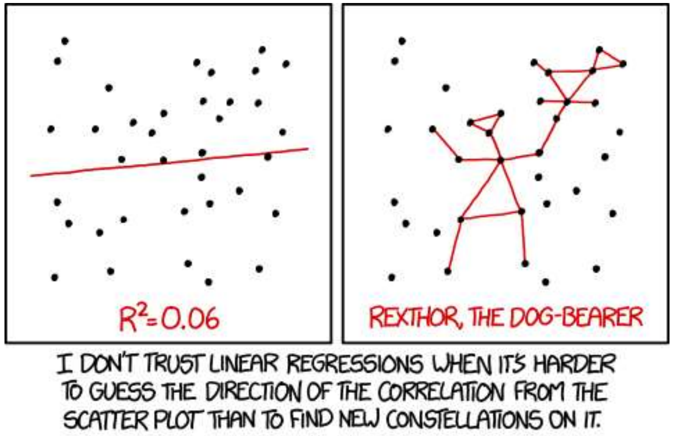
Simple Linear Model

- R-squared quite low
- Lots of data
- Most terms significant
  - 3 of our other predictors have poor p-values
    - Could investigate co-linearity here
    - May also be predictors that are unrelated to the response
  - Hour of day significant
    - Note that if one of the categorical terms is significant, we consider the whole model significant
- Residuals not normally distributed
- Predictions not great
- Higher accuracy on the test set
  - Not overfitting

Simple Linear Model: Is it any good?

- Sort of
  - No overfitting, simple model
  - Some poor terms, but most are meaningful
  - Predictive power is limited, but model seems to capture the main trends
  - End use needs to be kept in mind – is the model fit for purpose? How accurate does it need to be?

- Improving the model
  - Investigate higher order terms

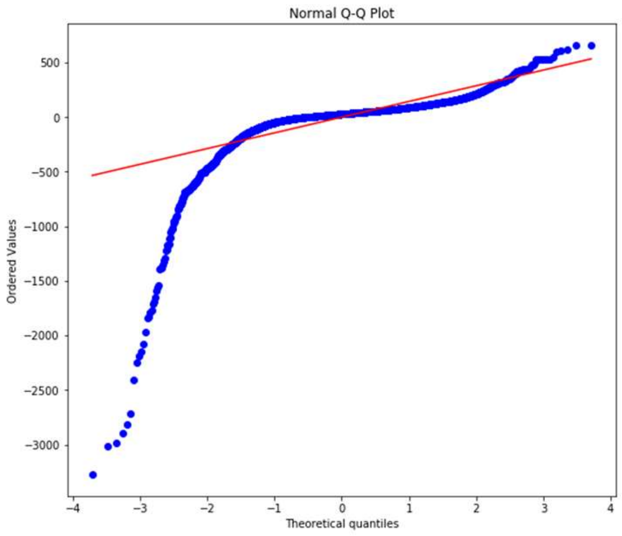


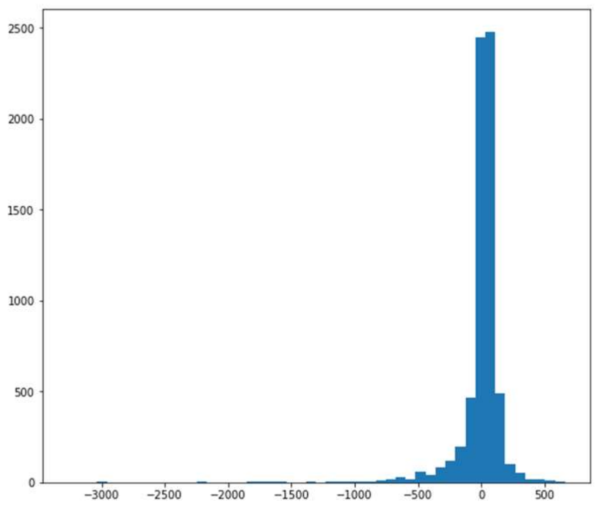
Cartoon from XKCD

A More Complex Model

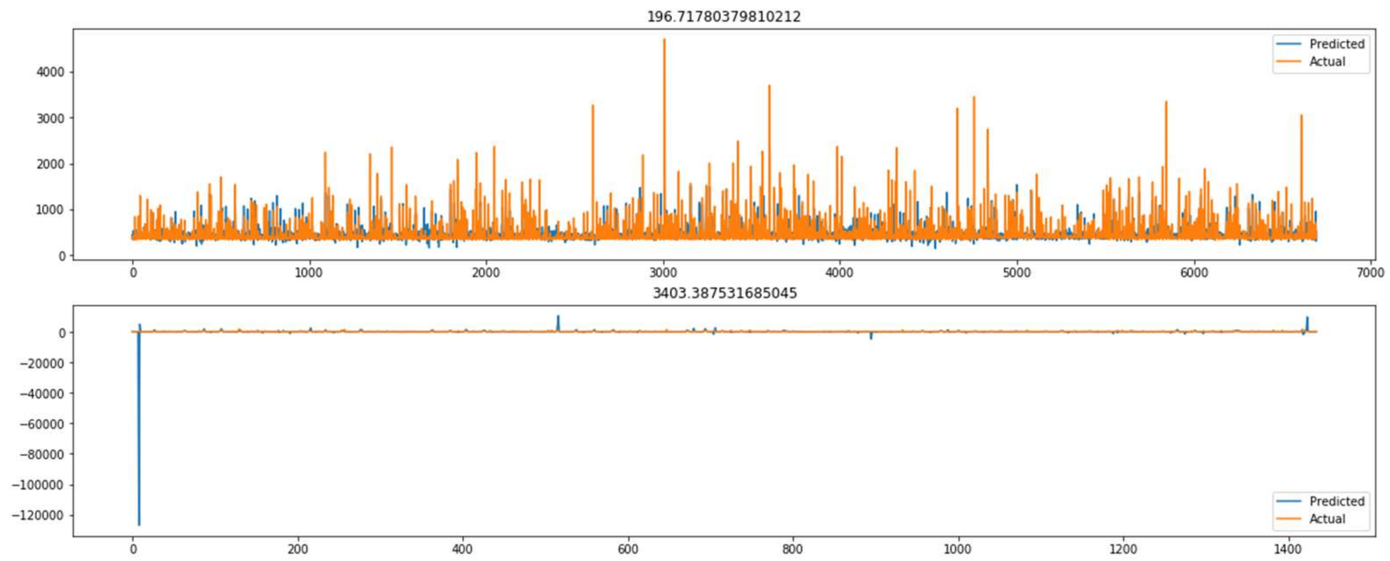
(quartic terms with interactions, and hour of day categorical term)

- ~500 model parameters
  - Too many terms to reasonably consider p-values, etc
  - R-squared of 0.412
    - A big improvement over what we had





A More Complex Model



A More Complex Model

- Improved R-squared (though with room for further improvement)
- Improved accuracy on training set
- Residuals not normally distributed
- Massive errors on the testing set
  - Model is overfitting

Complex Linear Model: Is it any good?

- Not really
  - Unpredictable performance on test data
  - Very high number of parameters
    - Difficult to inspect or tune due to size
    - Likely large amounts of redundancy, though difficult to assess due to model size

- Improving the model
  - Removing terms:
    - Reverting to lower order (i.e. quadratic rather than quartic) would reduce complexity, but may discard useful terms
    - Manual investigation is difficult given model size

CAB420: Regularisation

MAKING MODELS REGULAR?

Bias and Variance

- Bias and Variance are two factors in regression which we try to manipulate in order to find the "best" model.
- The **variance** of a model is the error from sensitivity to small changes in the training data. High variance can lead to overfitting.
  - Somewhat indicated by the R^2
- The **bias** of a model is the error from erroneous assumptions in the model. High bias can lead to underfitting.
  - Somewhat indicated by the RMSE
- As more terms are added to a model (i.e., it becomes more complex), the coefficients more accurately fit the given data (i.e., *bias decreases*).
- However, as more terms are added the model will become worse at predicting new data (i.e., *variance increases*) due to **over-fitting**

Bias and Variance

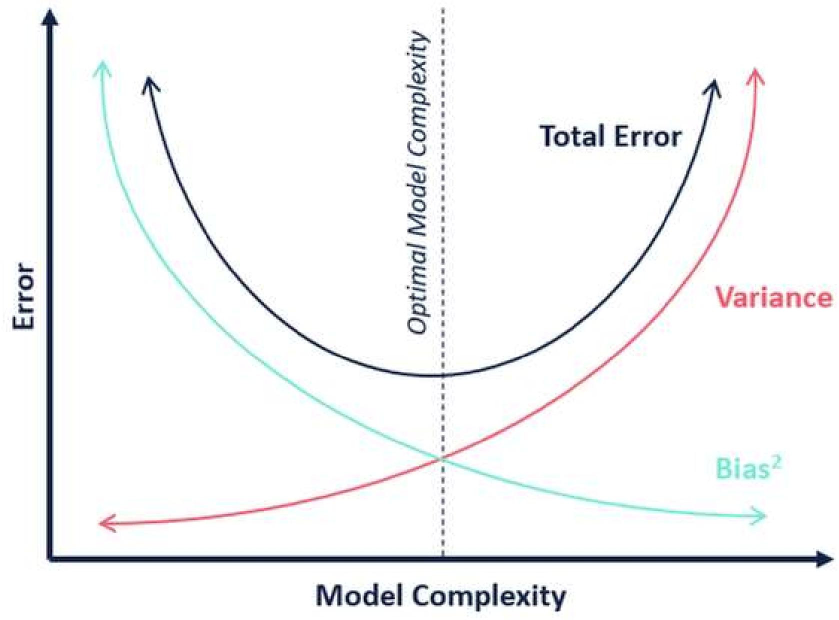


Image taken from blog on bias vs variance, found at:
https://community.alteryx.com/t5/Data-Science-Blog/Bias-Versus-Variance/ba-p/351862

Regularises

- Reduce the **magnitude** and/or **number** of parameters in order to reduce model complexity.
- Reduction in model complexity → reduced variance and increased bias.
- Useful when applied to models with many parameters.
- Regularisation seeks to penalise complex models
  - We have an intuition that a small change in input value to a model should lead to a small change in output value
  - Model complexity often leads to overfitting, reducing parameters (complexity) makes overfitting less likely

Regularisation and Regression

- Regularises are applied by penalising slope terms, β.
- There are two types of regularization we look at in CAB420:
  - L1 regularisation (Lasso regression), and
  - L2 regularisation (ridge regression).
- Both L1 and L2 seek to
  - Penalise big coefficients
  - Favour models with small slopes for individual data points
- Why?
  - A large slope means a small change in the data gives a large change in the estimate
  - Seek to reduce the model's variance, and make estimates more stable

Regularisation and Regression

- With linear regression we aim to find values for β that minimises

$$\sum_{i=1}^{n}\left(y_i - \sum_j x_{ij}\beta_j\right)^2$$

- Regularisation applies a penalty term

$$\sum_{i=1}^{n}\left(y_i - \sum_j x_{ij}\beta_j\right)^2 + \lambda P$$

where λ is a weight that controls the influence of our penalty

Regularisation and Regression

- Adds extra term(s) to the objective function
  - Terms don't operate over data or errors, but rather the model parameters
  - Regularisation terms are usually weighted
    - We can control how strong the regularisation is
    - How do we select the weight?

- Regularisation can also help when we have more dimensions than samples
  - Though in such situations we need to use an optimisation algorithm to find parameters

CAB420: Ridge Regression

L2 REGULARISATION

Ridge Regression

Linear Regression with L2 regularisation

- Add to our loss term the sum of the coefficients squared

$$\sum_{i=1}^{n}\left(y_i - \sum_{j} x_{ij}\beta_j\right)^2 + \lambda\sum_{j=1}^{p}\left\|\beta_j\right\|_2$$

  - We don't add the intercept

- Very big slopes are penalised heavily
  - Favour smaller slopes for all terms
  - Weight the L2 term by a factor, lambda
    - The ridge term

Regression Formulation: Revision

- Recall that for OLS regression:
  - Sum of squared errors term:

$$SSE(\beta) = (\mathbf{y}'\mathbf{y} \; - 2\beta'\mathbf{x}'\mathbf{y} + \beta'\mathbf{x}'\mathbf{x}\beta)$$

  - Derivative of SSE with respect to β:

$$\nabla SSE(\beta) = 2(\mathbf{x}'\mathbf{x}\beta - \mathbf{x}'\mathbf{y})$$

  - Setting to 0 and solving for β gives the optimal vector, $\hat{\beta}$:

$$\hat{\beta} = (\mathbf{x}'\mathbf{x})^{-1}\mathbf{x}'\mathbf{y}$$

Ridge Regression Formulation

- We want to minimize

$$(\mathbf{y}'\mathbf{y} \ - 2\beta'\mathbf{x}'\mathbf{y} + \beta'\mathbf{x}'\mathbf{x}\beta) + \lambda\beta'\beta$$

- Derivative with respect to β:

$$2(\mathbf{x}'\mathbf{x}\beta - \mathbf{x}'\mathbf{y} + \lambda\beta)$$

- Setting to 0 and solving for β gives the optimal vector, $\hat{\beta}$:

$$0 = \beta(\mathbf{x}'\mathbf{x} + \lambda I) - \mathbf{x}'\mathbf{y}$$
$$\hat{\beta} = (\mathbf{x}'\mathbf{x} + \lambda I)^{-1}\mathbf{x}'\mathbf{y}$$

- Known as **ridge** regression because the slope penalty term is added along the diagonal of $\mathbf{x}'\mathbf{x}$ like a ridge.

Demo

- See ***CAB420_Regression_Example_2_Regularised_Regression.ipynb***
- Same setup as our overfitting example from before
- Fit to data using Ridge Regression

Using Ridge Regression

- Formula:

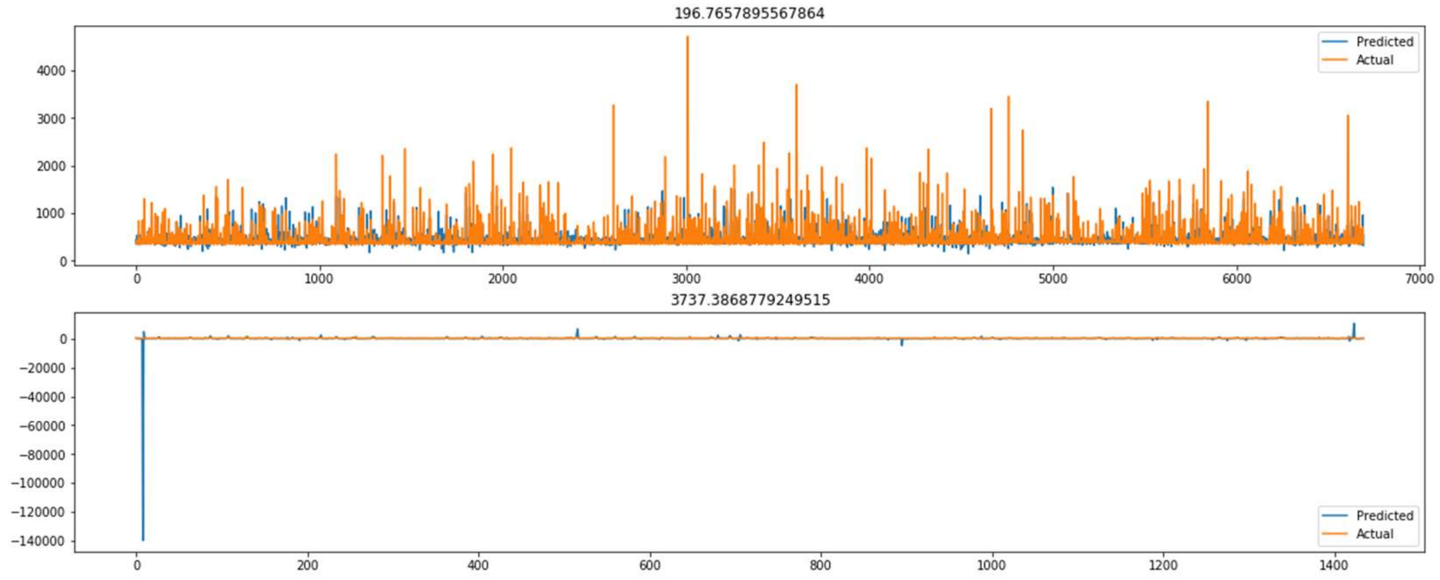
$$\sum_{i=1}^{n}\left(y_i - \sum_{j} x_{ij}\beta_j\right)^2 + \lambda\sum_{j=1}^{p}\|\beta_j\|_2$$

  - We need to choose λ

- What should λ be?
  - What happens if it's 0?
  - Let's try 1

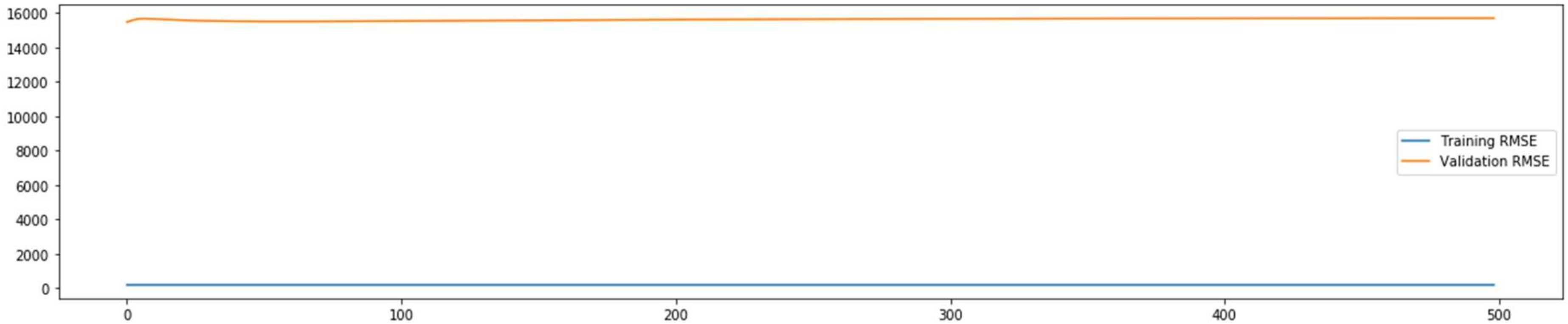
Ridge Regression: Results

- λ perhaps should not be 1
- Instead, try a range of values
  - 0, 2, 4, 6, …., 498, 500



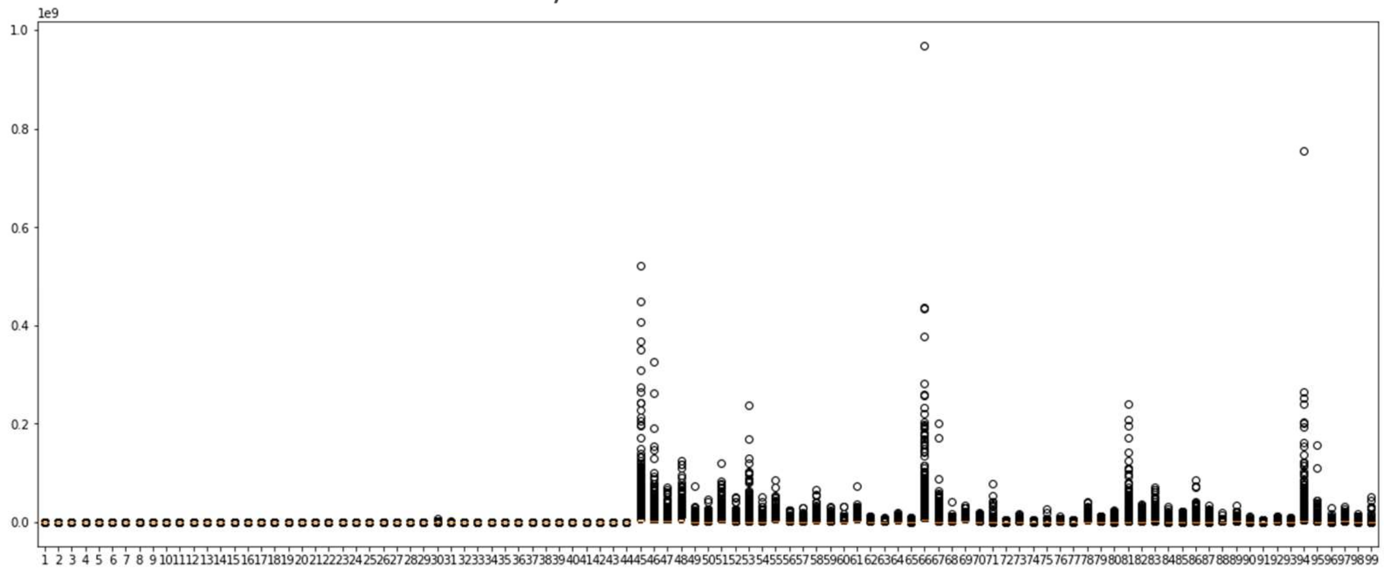
Ridge Regression: Results

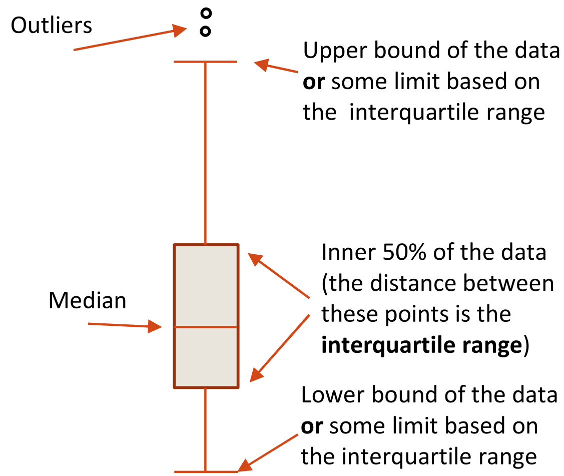
- Plotting RMSE as λ changes
- We see a very small change as λ increases
  - Clearly λ needs to be much bigger with the data as it is



An Aside: Standardisation

- Let's visualise our data using a box plot
- We can see that different variables have very different ranges
  - First 100 dimensions only shown





Standardisation – Why?

- For a given dataset, dimensions are usually in different scales
  - i.e. Dimension 1 may range from [0..1], Dimension 2 may range from [100…100000]
  - With a regularisation penalty, Dimension 1 may be penalised much more than Dimension 2 due to its scale

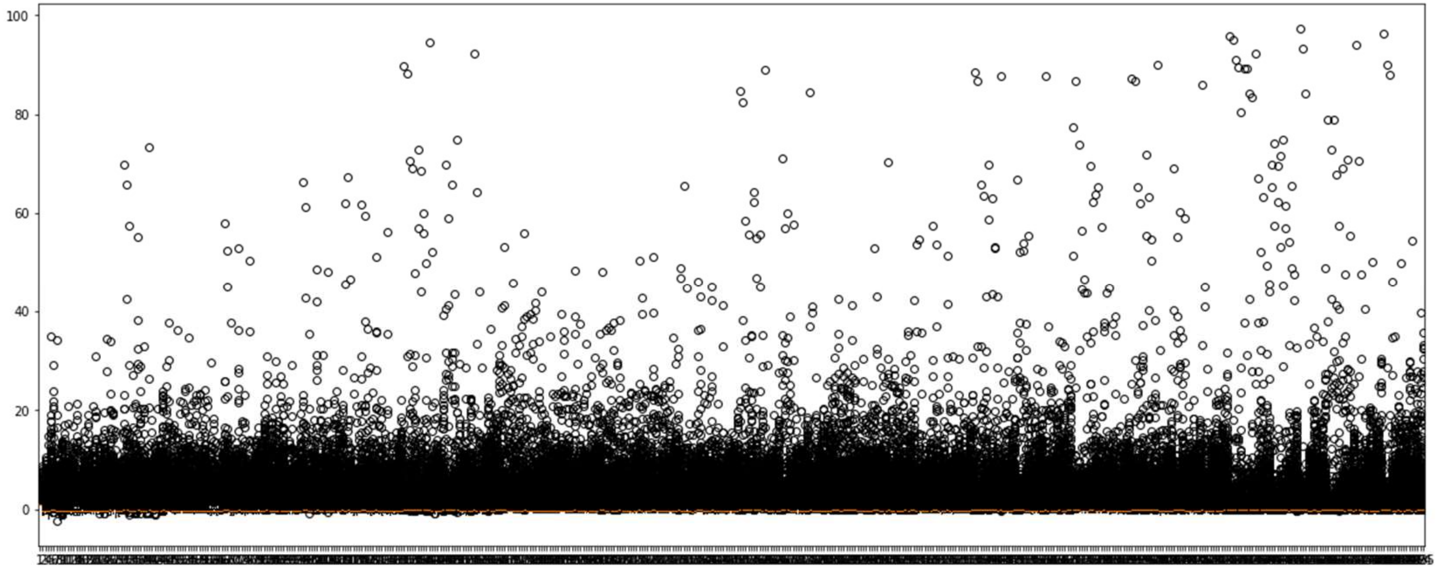
- We seek to scale all dimensions equally, so that they are all considered equally when fitting a model

Standardisation – What?

- For each dimension
  - Get the mean and standard deviation
  - For that dimension, subtract the mean, divide by the standard deviation
- End result:
  - All dimensions have mean 0, standard deviation 1
  - i.e. they are all scaled to the same range
  - Outliers are preserved
    - A point that is 10 standard deviations away in the original set, is still 10 standard deviations away
- Also
  - It usually makes the model easier to visualise

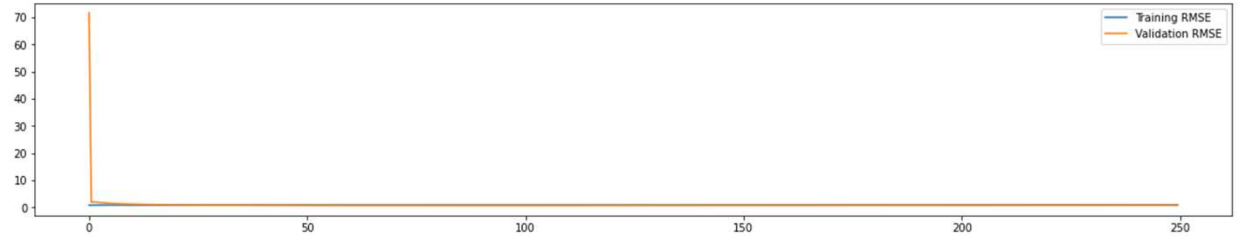
Standardised Data

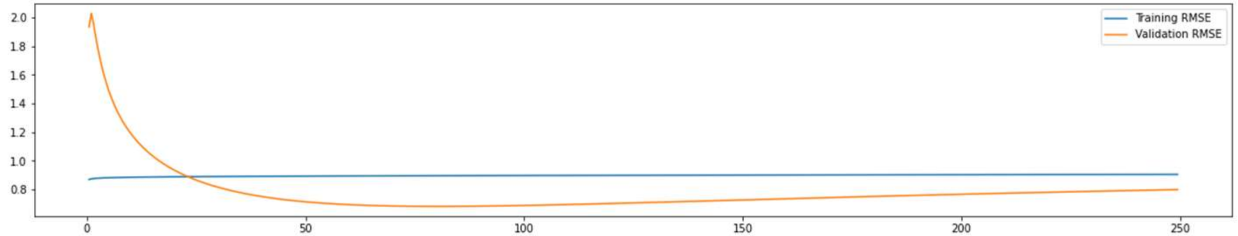
- All data now has a similar range
  - First 100 dimensions shown
  - Lots of outliers still visible



Ridge Regression with Standardised Data

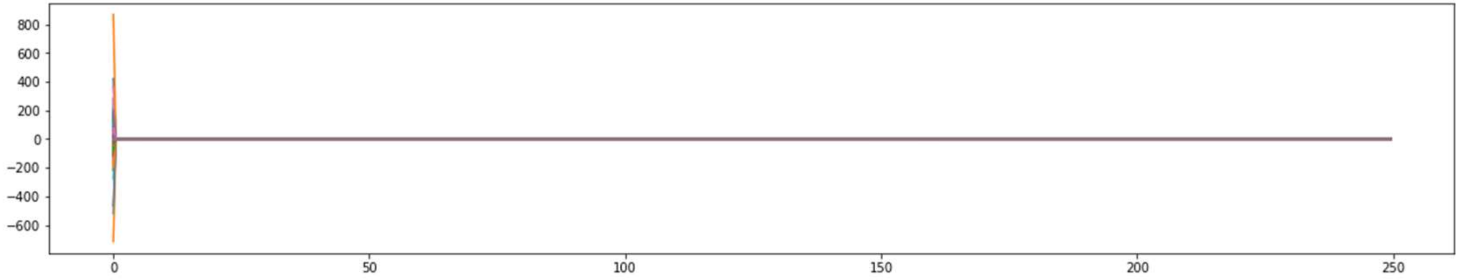
- RMSE vs λ
  - We see an immediate drop as we increase λ
  - Remember, λ = 0 is least squared regression
  - Value which minimises the Validation RMSE is our best λ
    - For us, this is 79.5
  - Training RMSE will gradually increase with λ
    - Variance vs Bias

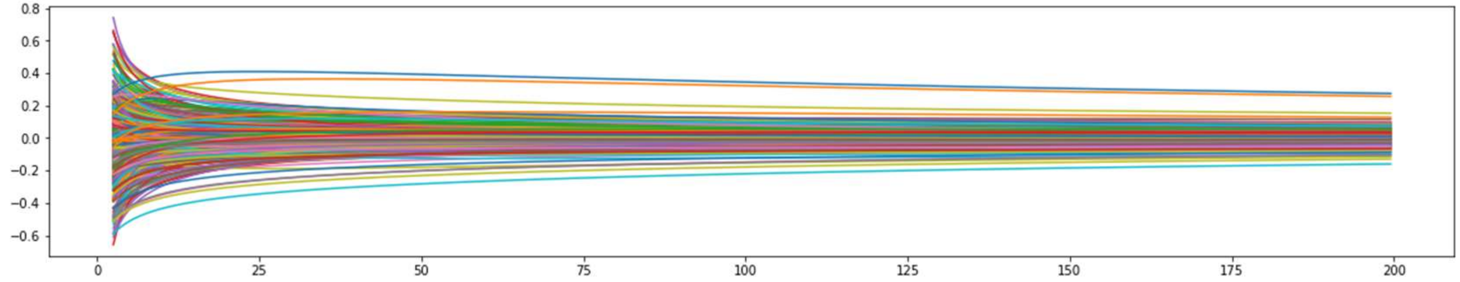




Ridge Trace Plot

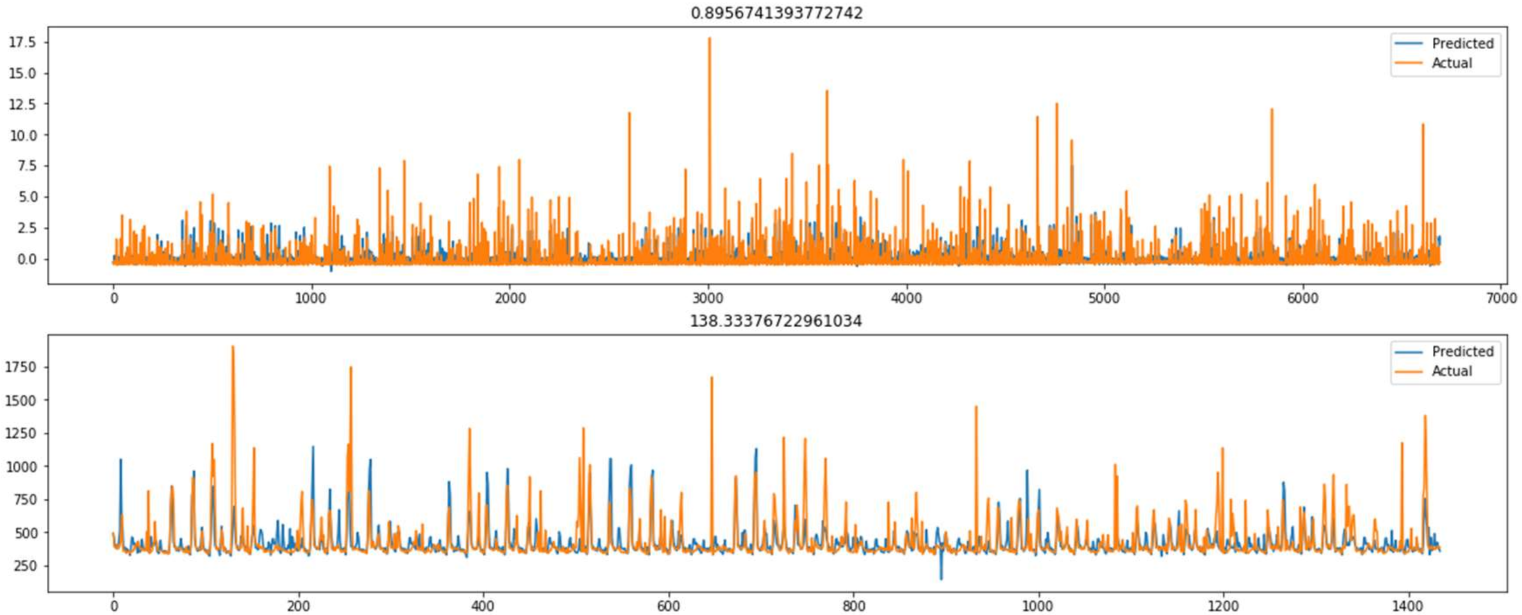
- Individual Coefficients vs λ
  - Increases in λ lead to smaller coefficients overall
    - Note the distorted scale when $\lambda = 0$ is inlcuded
  - Coefficients gradually decrease and slowly approach 0





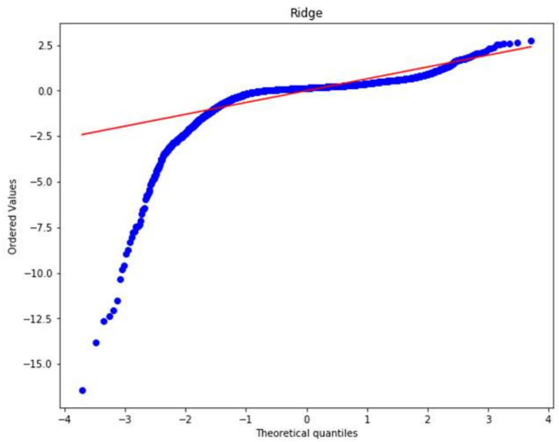
Ridge Results

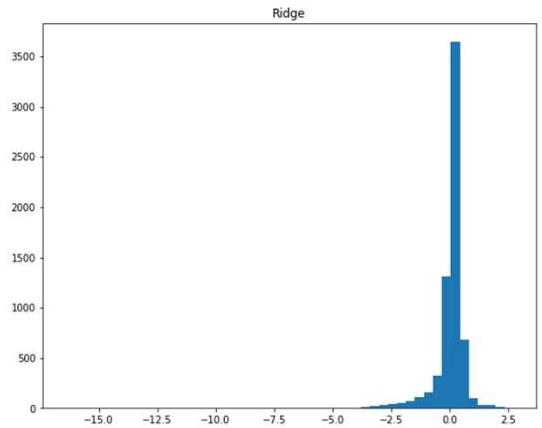
- Final Model, $\lambda = 79.5$
  - Similar performance to original Linear model



Ridge Results

- Final Model, $\lambda = 79.5$
- $R^2 = 0.244$
  - Much lower R^2 than our higher order linear model, yet better performance on validation data
  - Variance vs Bias
- Similar looking residual plots to previously





CAB420: LASSO Regression

L1 REGULARISATION

LASSO Regression

Linear Regression with L1 regularisation

- Add to our loss the sum of absolute values of coefficients

$$\sum_{i=1}^{n}\left(y_i - \sum_{j} x_{ij}\beta_j\right)^2 + \lambda \sum_{j=1}^{p}\|\beta_j\|_1$$

  - Again, we don't add the intercept
- Compared to Ridge Regression

$$\sum_{i=1}^{n}\left(y_i - \sum_j \; x_{ij}\beta_j\right)^2 + \lambda \sum_{j=1}^{p}\|\beta_j\|_2 \text{ vs } \sum_{i=1}^{n}\left(y_i - \sum_j \; x_{ij}\beta_j\right)^2 + \lambda \sum_{j=1}^{p}\|\beta_j\|_1$$

  - Only difference is the type of norm being used
    - L1 (LASSO) vs L2 (Ridge)
  - Big coefficients aren't penalised quite as badly
  - Coefficients can go to 0
    - We can eliminate poor terms
  - L1 norm still controlled by a scaling factor

Lasso Regression Formulation

- We want to minimize

$$(\mathbf{y}'\mathbf{y} \; - 2\beta'\mathbf{x}'\mathbf{y} + \beta'\mathbf{x}'\mathbf{x}\beta) + \lambda\beta$$

- The following is the derivative with respect to β:

$$\mathbf{2x}'\mathbf{x}\beta - \mathbf{2x}'\mathbf{y} + \lambda I$$

- Setting to 0 and solving for β gives the optimal vector, $\hat{\beta}$:

$$\hat{\beta} = (2\mathbf{x}'\mathbf{x})^{-1}(\mathbf{2x}'\mathbf{y} - \lambda I)$$

- Where does the name come from?
  - Acronym: **L**east **A**bsolute **S**election and **S**hrinkage **O**perator
- Not completely straight-forward, as the term in the first line should be $\lambda|\beta|$
  - This actually makes it a lot more complex

Demo

- See ***CAB420_Regression_Example_2_Regularised_Regression.ipynb***

- Same setup as our overfitting and ridge regression

- Fit to data using LASSO Regression

Using Lasso Regression

- Formula:

$$\sum_{i=1}^{n}\left(y_i - \sum_{j} x_{ij}\beta_j\right)^2 + \lambda \sum_{j=1}^{p}\left\|\beta_j\right\|_1$$

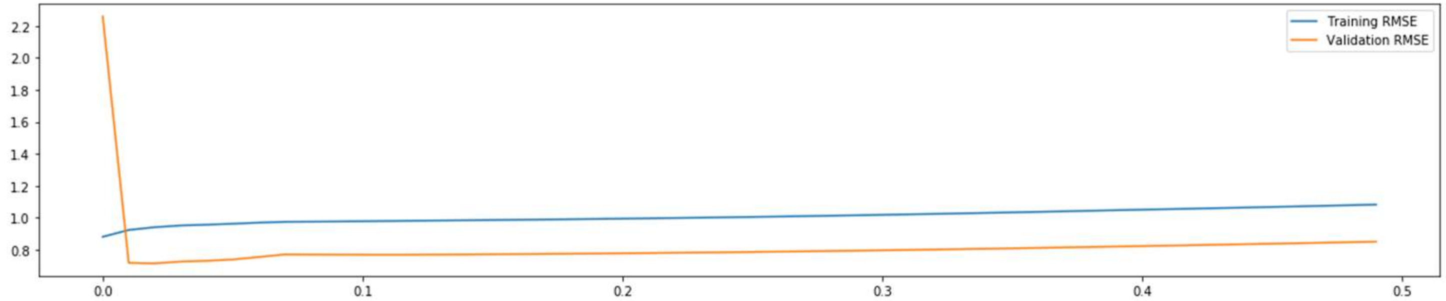
  - We need to choose λ

- As per Ridge, we'll use a range
  - 0 to 0.5 in steps of 0.01
  - Lasso typically uses a smaller λ than ridge

- We'll use standarised data from the start

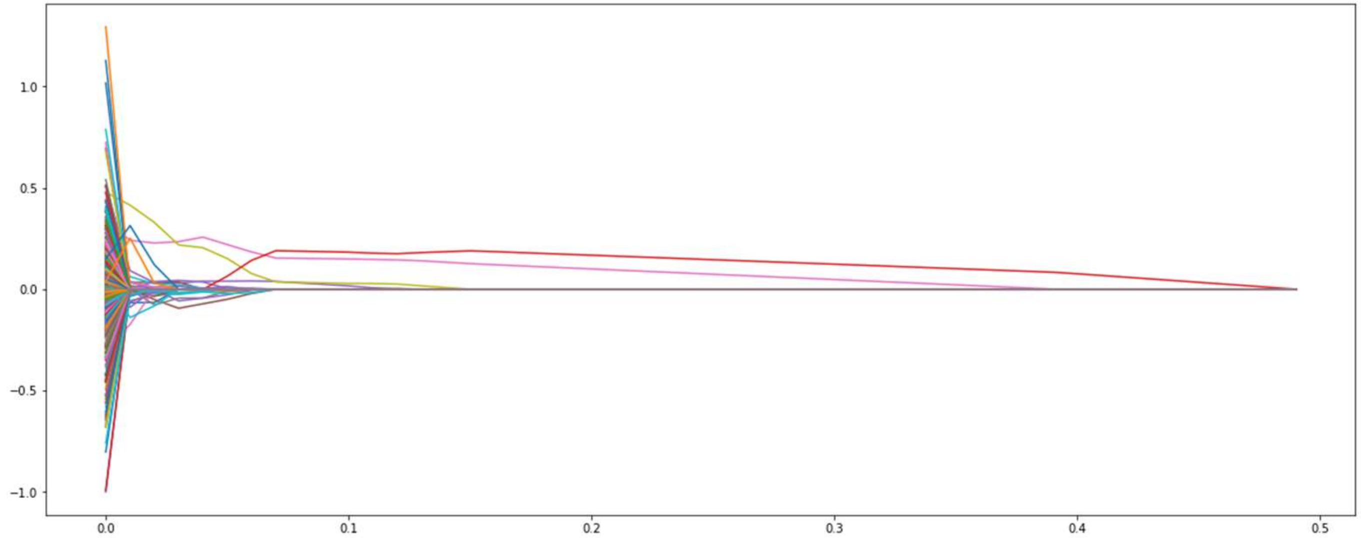
Lasso: Selecting Lambda

- Best $\lambda = 0.02$
- Same trend as ridge
  - Training data always increases with λ
  - Validation data decreases to a minimum, then increases



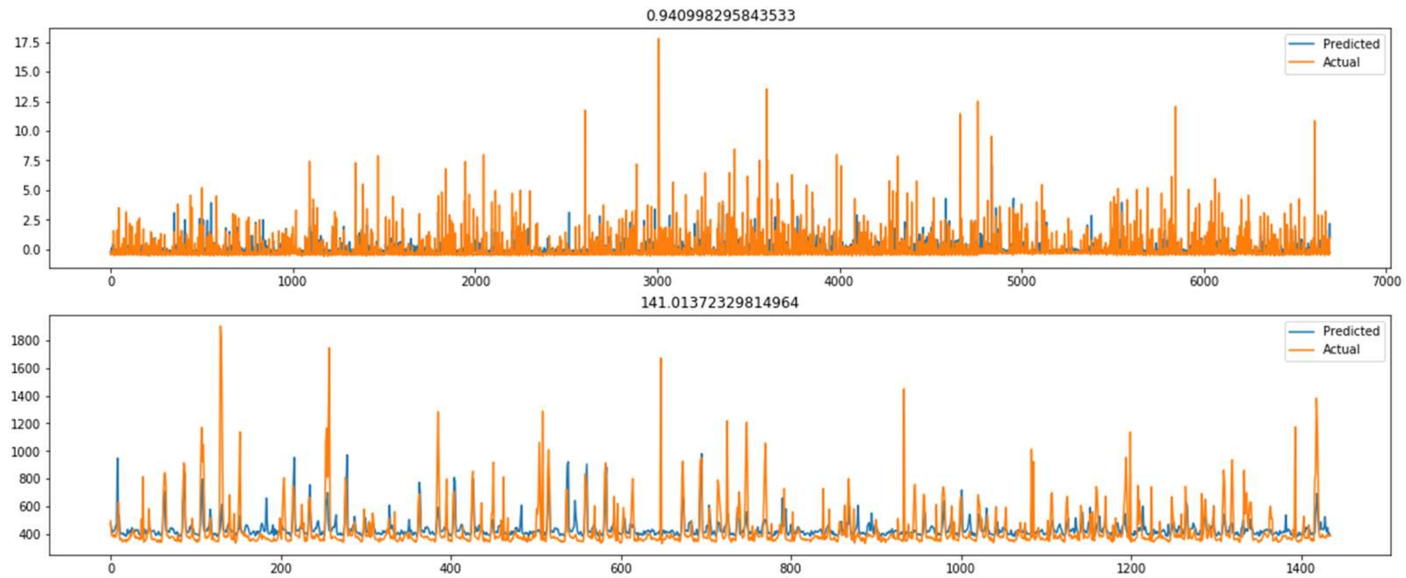
Lasso Trace Plot

- Terms decrease in value as λ increases
  - Terms can go to 0 and be eliminated
  - At the far end of the plot, all terms are 0 (constant model)



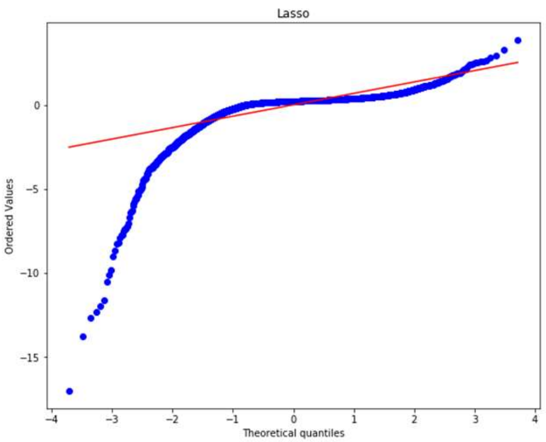
Lasso Results

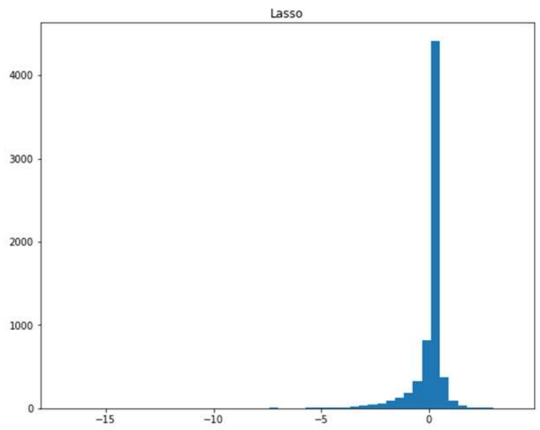
- Final Model, $\lambda = 0.02$
  - Similar to Ridge and Linear Model
  - Final model contains 26 terms (all others are 0)



Lasso Results

- Final Model, $\lambda = 0.02$
- $R^2 = 0.315$
  - Between higher order linear model and ridge model
  - Model less accurate than ridge on training data, more accurate than higher order linear model, Variance vs Bias again
  - Similar looking residual plots to previously





ElasticNet Regression

- Bonus Regression Method!
- StatsModels regression implementation also does ElasticNet Regression
  - L1 and L2 terms added to the least squares loss
  - By default the function does pure Lasso
- Does this mean it’s twice as good?
  - Not really, though it’s not bad either
  - It does mean that we now have another hyper-parameter to tune
    - We need to select the relative weight of the two terms

A Note on Comapring Models

- We're only comparing our data on
  - Training data: which the model is trained on
  - Validation data: which is used to select lambda
- Ideally, we want a third dataset
  - Testing data: totally unseen, used to confirm that our model generalises to unseen data

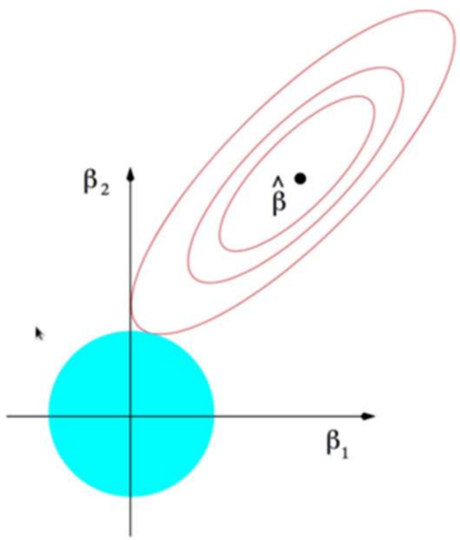
CAB420: Ridge vs LASSO

WHICH ONE?

Ridge vs Lasso

$$\sum_{i=1}^{n}\left(y_i - \sum_{j} x_{ij}\beta_j\right)^2 + \lambda \sum_{j=1}^{p} \|\beta_j\|_2$$

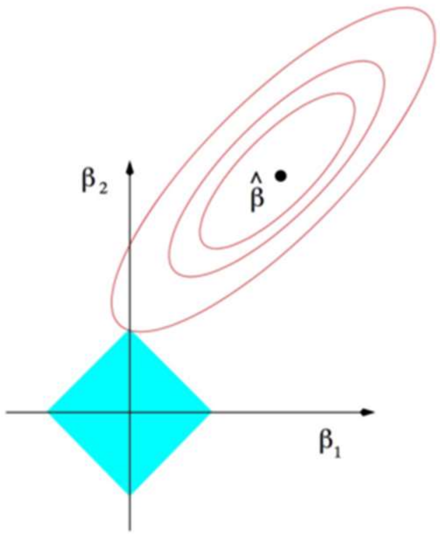
- We have a two coefficients
  - The "best solution" according to least squares is $\hat{\beta}$
  - The blue area is the constraint region for a given λ
- Ridge uses an L_2norm
  - Circular constraint region
  - Closest point on the constraint region to $\hat{\beta}$ is our ridge solution



Ridge vs Lasso

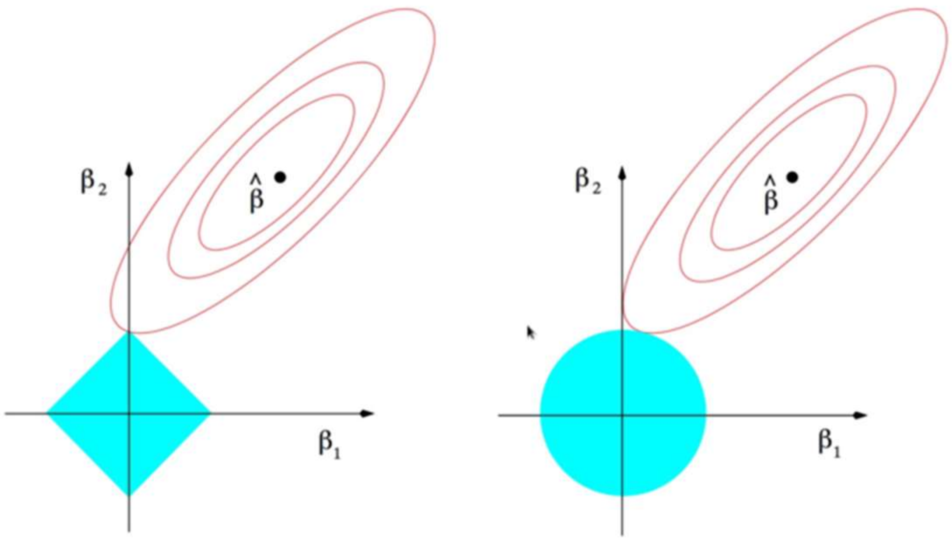
$$\sum_{i=1}^{n}\left(y_i - \sum_{j} x_{ij}\beta_j\right)^2 + \lambda \sum_{j=1}^{p} \left\|\beta_j\right\|_1$$

- We have a two coefficients
  - The "best solution" according to least squares is $\hat{\beta}$
  - The blue area is the constraint region for a given λ
- Lasso uses an L_1norm
  - Diamond shaped constraint region
  - Closest point on the constraint region to $\hat{\beta}$ is our ridge solution



Ridge vs Lasso

- Due to the shape of the constraint region
  - Lasso can pull terms to 0
  - Ridge can make terms very small, but not 0



Impact of λ

ANOTHER LOOK AT WHAT IT DOES

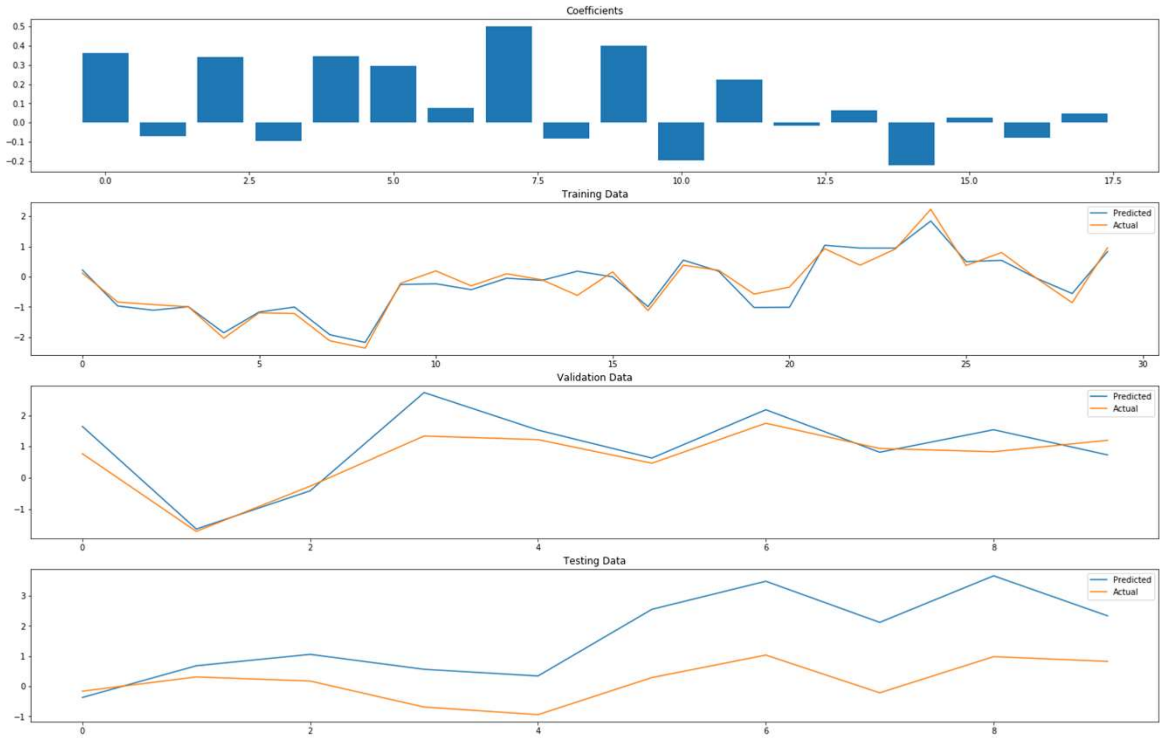
A Simple Example

- See ***CAB420_Regression_Additional_Example_Regularisation_Impact.ipynb***

- Predict traffic times again
  - Standardised data
  - 18 predictors
  - Linear, Ridge and Lasso models
  - Training, validation and testing set all taken from different time periods
    - Split in chronological order

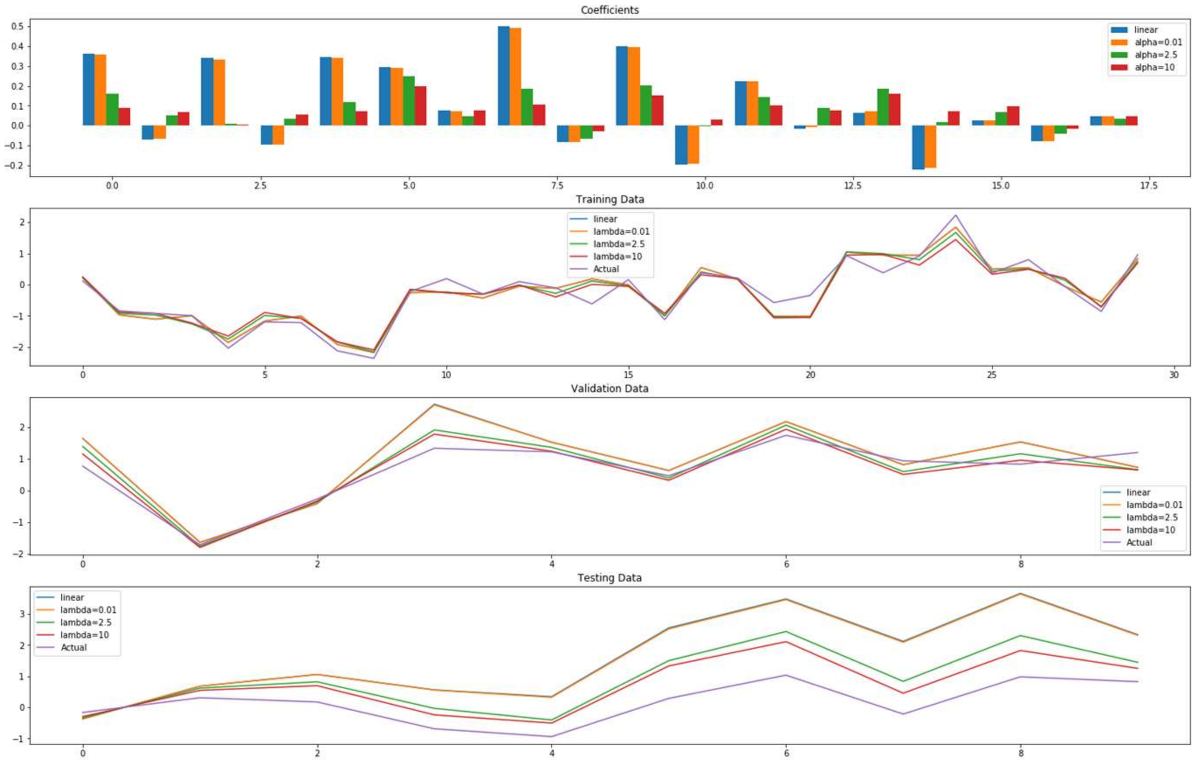
Linear Model

- Excellent fit to training data
- Fit gets worse for validation and testing data
- Coefficients vary in value



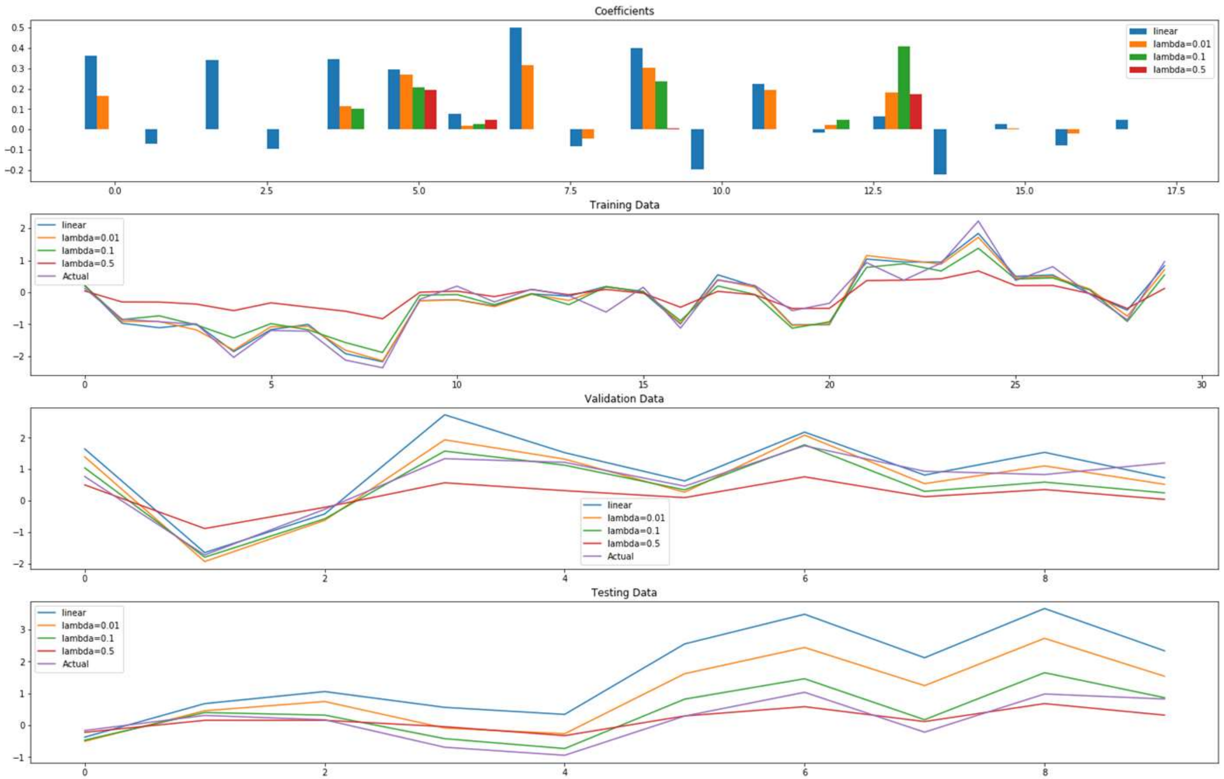
Ridge Model

- Larger λ leads to
  - Smaller coefficients
  - Flatter prediction curves
  - Coefficents can change sign
- Largest λ is least accurate on training data, most accurate on testing data



Lasso Model

- Larger λ leads to
  - Smaller coefficients
  - Flatter prediction curves
  - Coefficents can change sign
- Coefficients can go to 0
  - Can happen at very small lambda
- Large λ will push all coefficients to 0

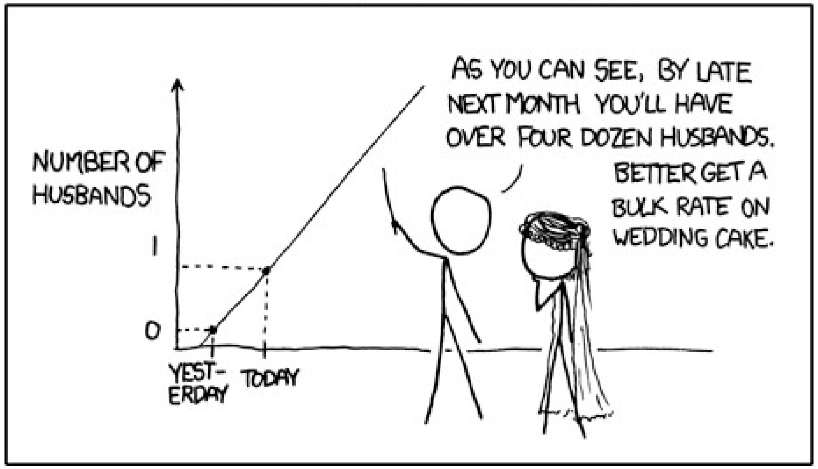


Regularised Regression and Small Datasets

Regression Data Requirements

- Usually, we would like to have more data points than parameters
- If we don't have this, direct solutions to fit a regression function will fail
- However, gradient descent can be used to find a solution
  - Allows us to fit high dimensional models to small datasets
  - Increases the danger of overfitting
- In general, extrapolation with linear regression can be risky

MY HOBBY: EXTRAPOLATING



Cartoon from XKCD

Demo

- See ***CAB420_Regression_Example_3_Regression_with_Less_Data.ipynb***

- Traffic time prediction again, but with very limited data
  - 50 samples total
    - 30 training, 10 validation, 10 testing
  - ~150 variables
- Linear model will overfit
- Lasso and Ridge can be used to get a better fit to the data
- Review this example in your own time
  - Covered in more detail in the interactive session