CAB420: Overfitting and Linear Regression

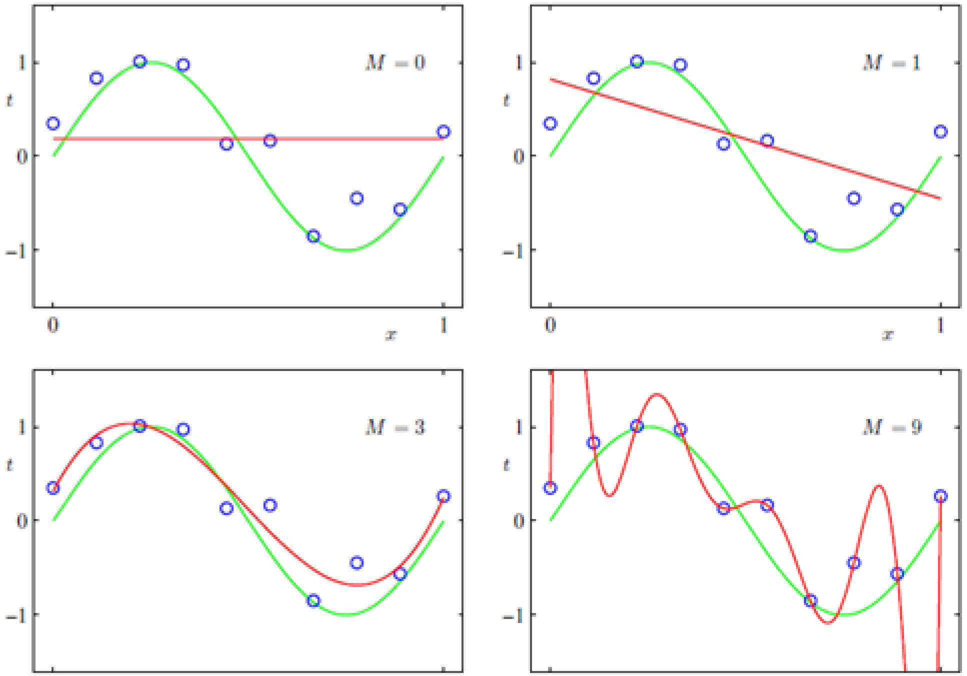
WHAT IS IT? AND WHY DO I CARE?

Overfitting and Regression

- Consider a multi-variate linear regression task
- We can (usually) make the model more accurate on the training set by adding more terms
  - Additional variables
  - Higher order terms
- For a time, it will also get more accurate on the test set
  - After a while though, it may go very wrong

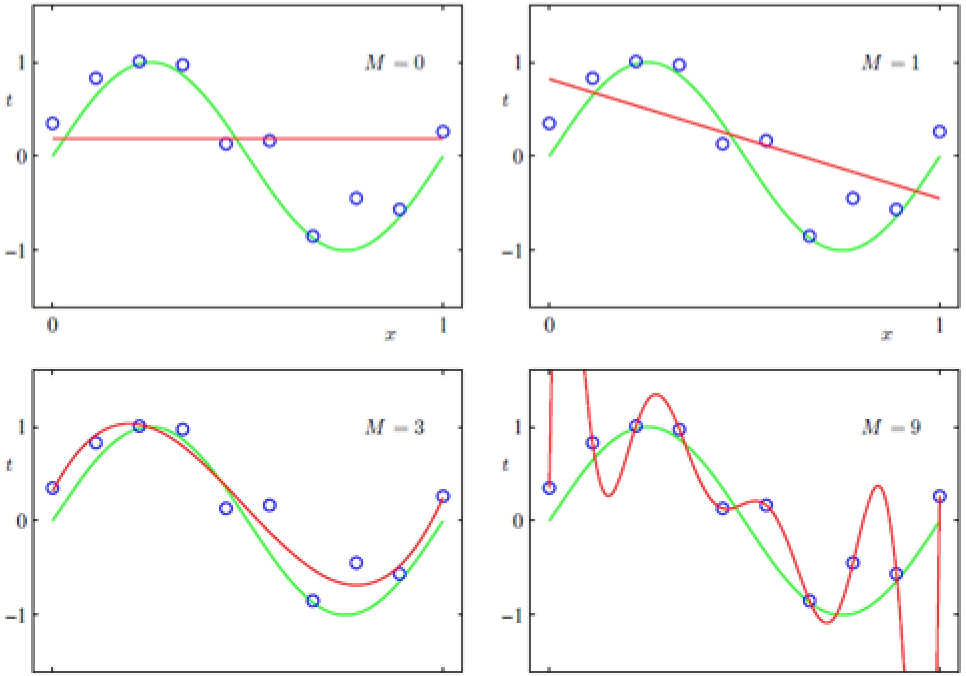
Overfitting and Regression

- 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 (9^{th} 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***

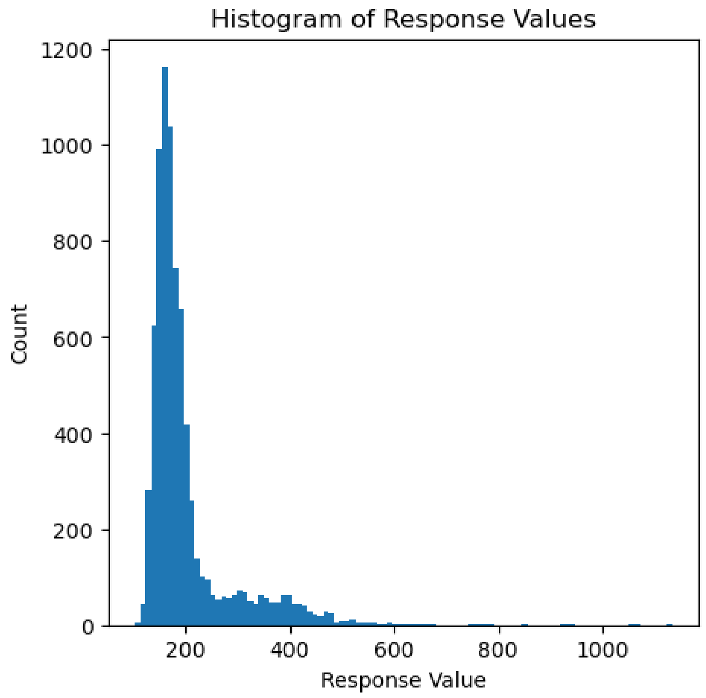
- Demo Overview
  - Load traffic data from Brisbane which contains average travel times between key points on the road network

  - We'll consider
    - The first 8 data series and time of day as predictors
    - The 33^{rd} data series as the response

  - Apply linear regression to data, increase complexity and observe results

Why the 33rd Data Series?

- We're predicting traffic travel times
- These are typically very non-Gaussian, and so don't get modelled very well
  - Lower bound on time, typically lots of values close to this
  - Very, very long tail with increasingly huge travel times due to traffic jams, etc
- The 33rd data series is simply not as bad as some of the others



Simple Linear Model

(linear terms with hour of day categorical term)

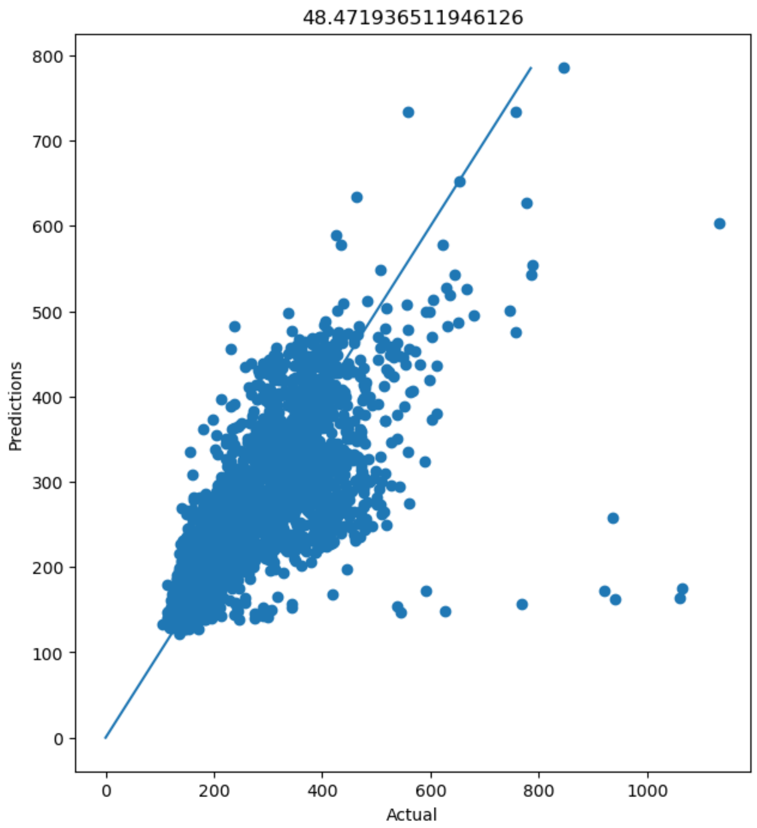
OLS Regression Results

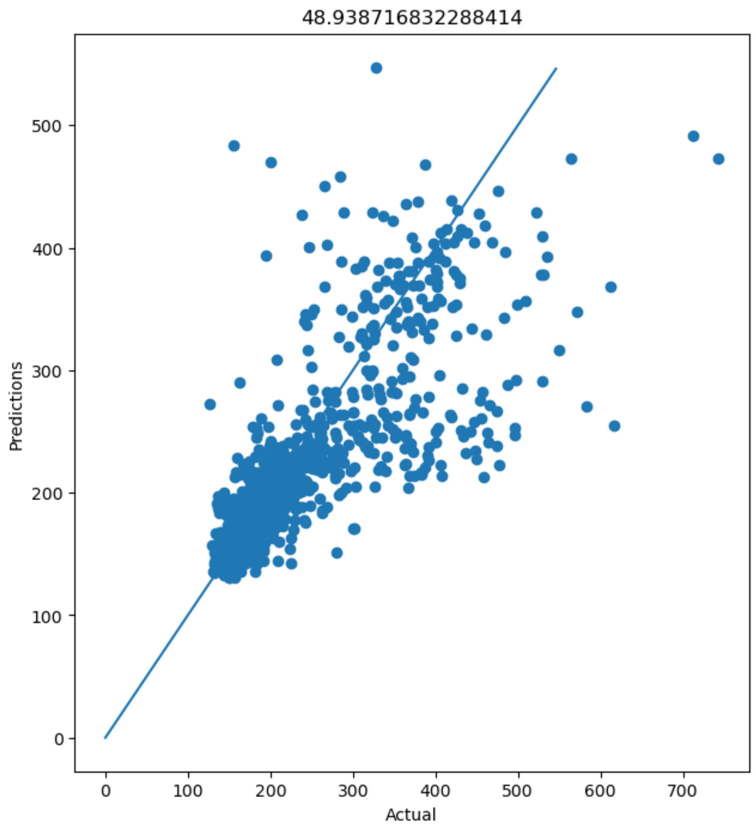
Dep. Variable:	x_1059__1060_	R-squared:	0.669
Model:	OLS	Adj. R-squared:	0.668
Method:	Least Squares	F-statistic:	504.2
Date:	Tue, 09 Jan 2024	Prob (F-statistic):	0.00
Time:	02:32:45	Log-Likelihood:	-41127.
No. Observations:	7760	AIC:	8.232e+04
Df Residuals:	7728	BIC:	8.254e+04
Df Model:	31		
Covariance Type:	nonrobust		

	coef	std err	t	P>|t|	[0.025	0.975]
const	15.5238	4.717	3.291	0.001	6.278	24.770
x_1098__1056_	0.3775	0.012	31.463	0.000	0.354	0.401
x_1058__1059_	0.6866	0.017	39.482	0.000	0.652	0.721
x_1057__1056_	0.0194	0.034	0.569	0.570	-0.047	0.086
x_1017__1007_	-0.0002	0.010	-0.022	0.982	-0.019	0.019
x_1115__1015_	-0.0976	0.034	-2.835	0.005	-0.165	-0.030
x_1015__1115_	0.7091	0.057	12.395	0.000	0.597	0.821
x_1103__1061_	0.2087	0.028	7.542	0.000	0.154	0.263
x_1135__1231_	0.0437	0.023	1.901	0.057	-0.001	0.089
1	-6.6110	5.373	-1.230	0.219	-17.143	3.921
2	0.6483	5.517	0.118	0.906	-10.167	11.463
3	5.6162	4.561	1.231	0.218	-3.324	14.556
4	7.6226	4.217	1.807	0.071	-0.645	15.890
5	1.6447	4.184	0.393	0.694	-6.557	9.846
6	7.1633	4.290	1.670	0.095	-1.246	15.573
7	5.7068	4.418	1.292	0.196	-2.954	14.367
8	-2.1201	4.431	-0.478	0.632	-10.806	6.566
9	-3.3471	4.238	-0.790	0.430	-11.655	4.961

Simple Linear Model

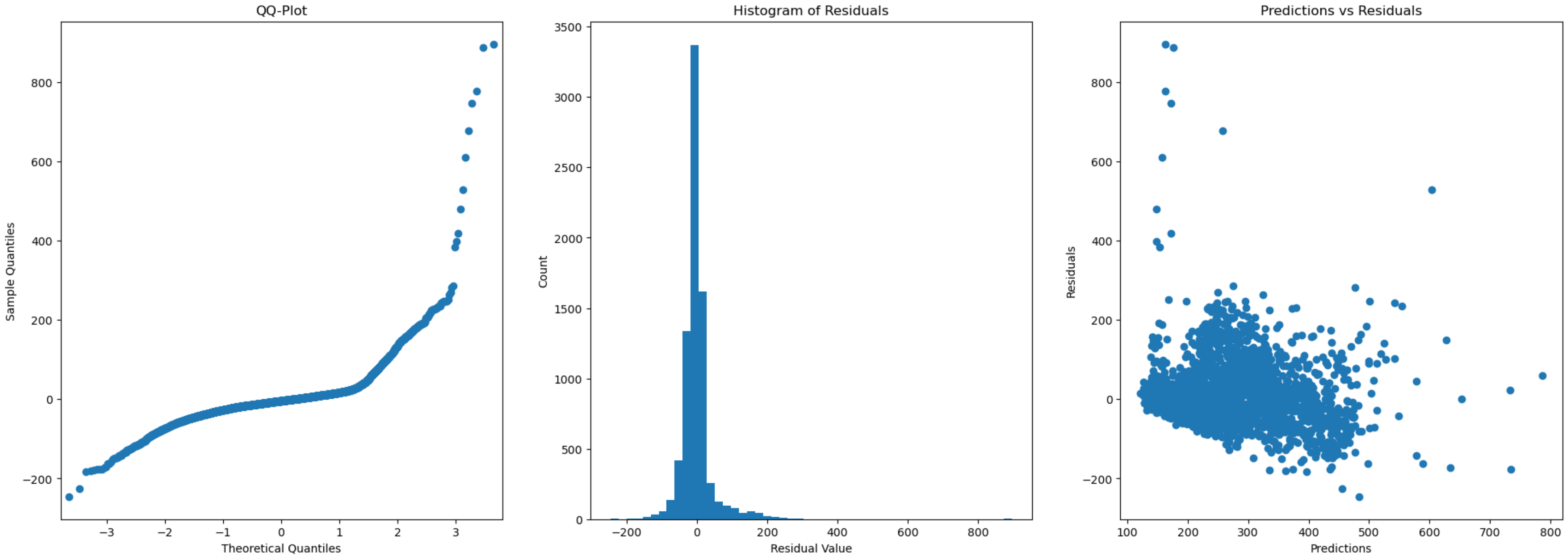
(linear terms with hour of day categorical term)





Simple Linear Model

(linear terms with hour of day categorical term)



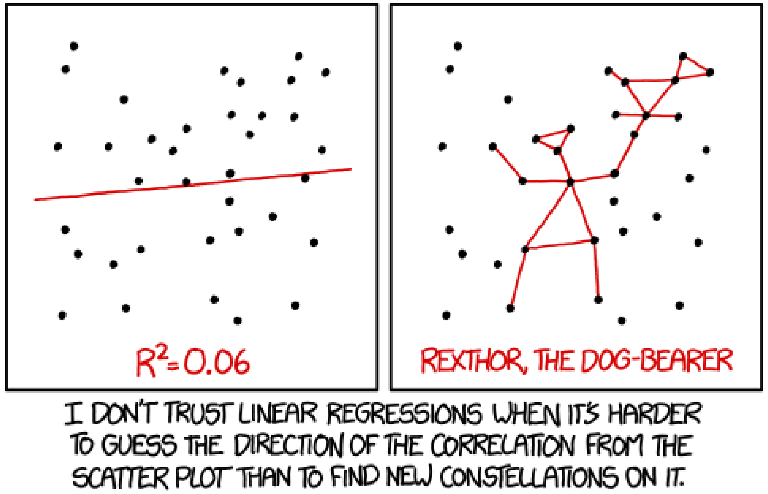
Simple Linear Model – In Summary

- R-squared not bad
- 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
- Predictions not too bad
  - Some outliers, likely caused by traffic events
  - Similar performance on training and testing sets
- Residuals not normally distributed
  - Very long tail, possibly caused by traffic events?
- Some evidence of heteroscedasticity

Simple Linear Model: Is it any good?

- Sort of
  - No overfitting, simple model
  - Some poor terms, but most are meaningful
  - Predictive power is not too bad, the model seems to capture the main trends
  - Residual distributions look problematic
    - Part of this is down to our very non-linear response
  - 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

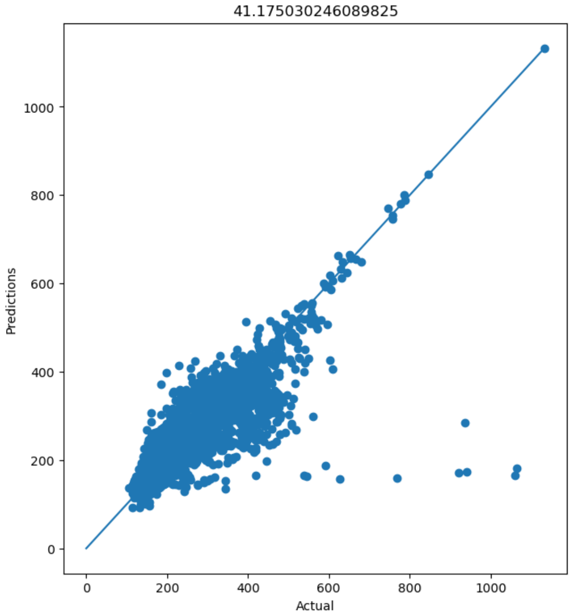


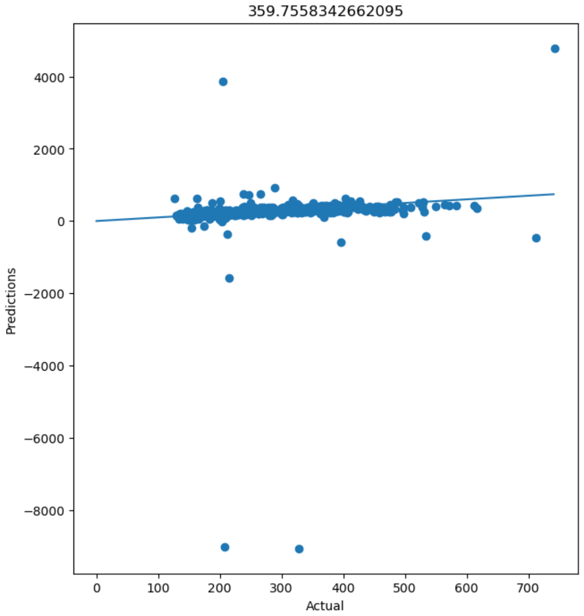
Cartoon from XKCD

A More Complex Model

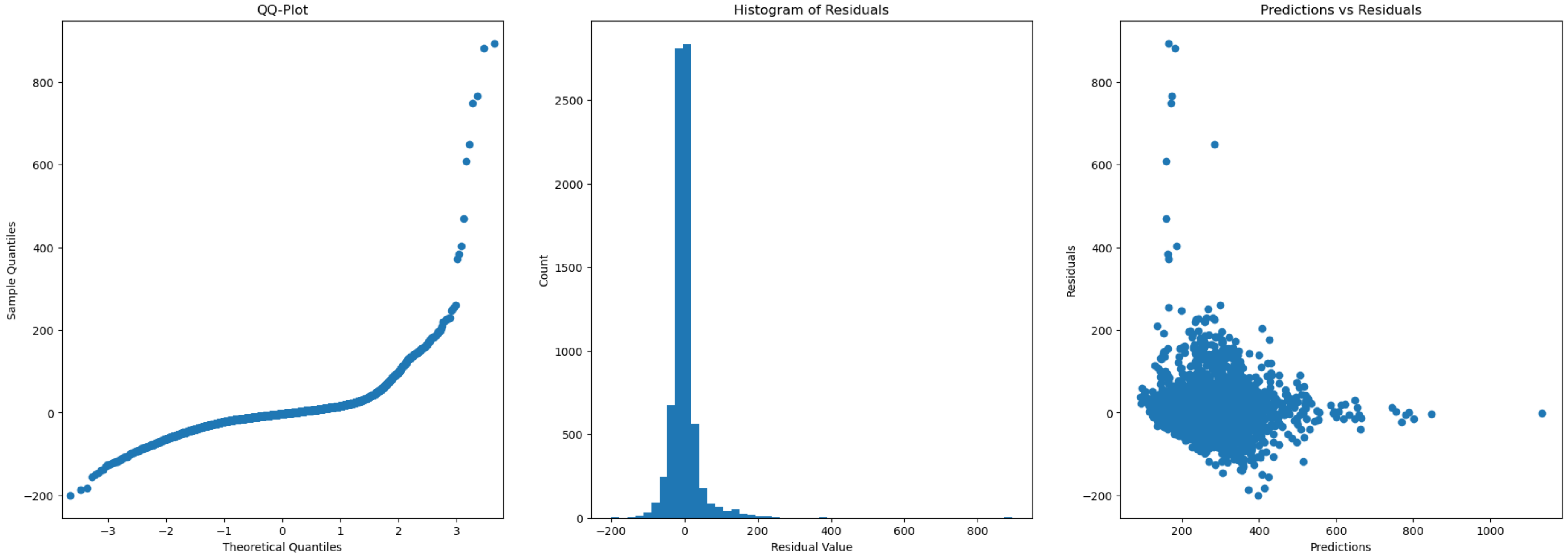
(quartic terms with interactions, and hour of day categorical term)

- ~500 model parameters
  - Too many terms to reasonably consider p-values, etc
  - R-squared of 0.761
    - It was 0.669, it's better, but that much better





A More Complex Model



A More Complex Model

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

Complex Linear Model: Is it any good?

- Probably not
  - 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
  - Regularisation!

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

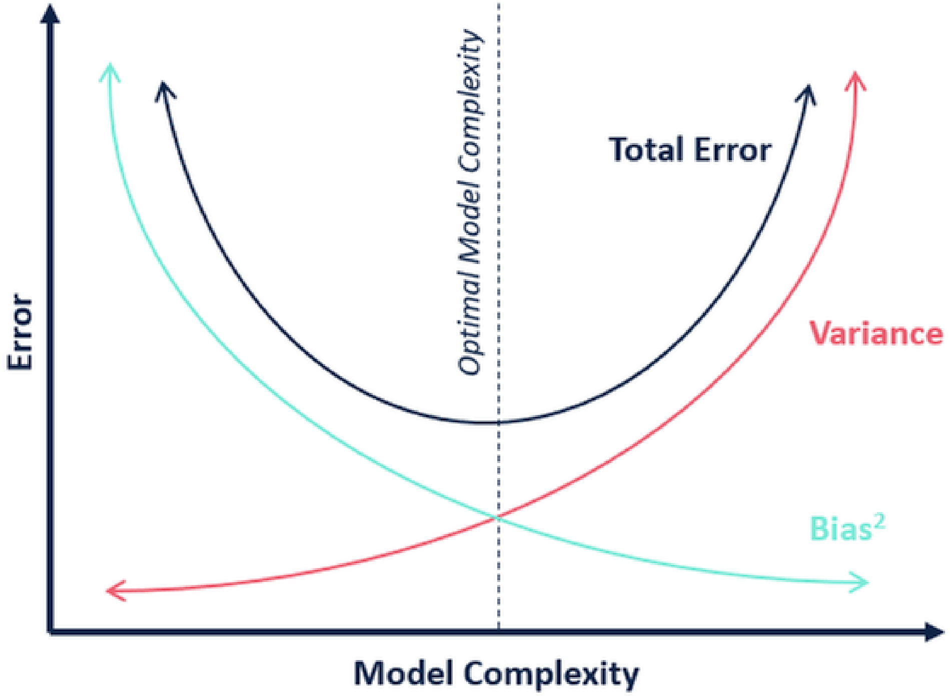


Image taken from blog on bias vs variance, found at:
https://community.alteryx.com/t5/Data-Science-Blog/Bias-Versus-Variance/ba-p/351862

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:

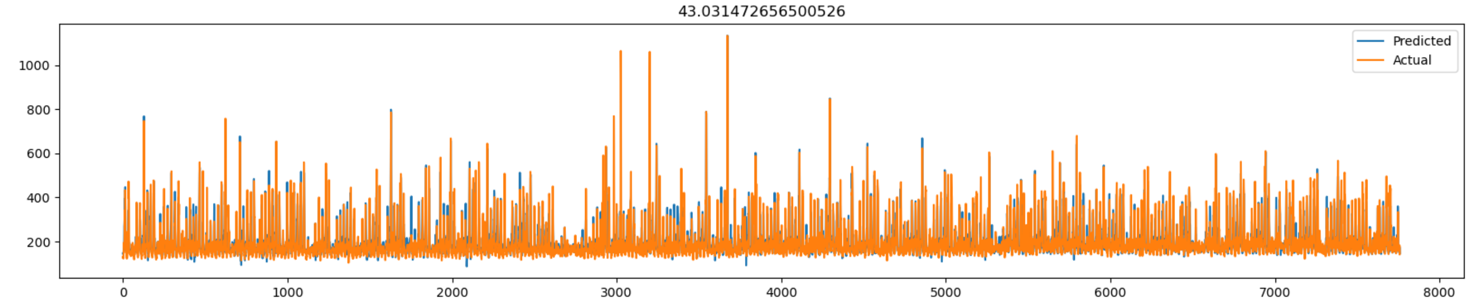
$$\sum_{i=1}^{n}\left(y_i - \sum_{j} x_{ij}\beta_j\right)^2 + \lambda\sum_{j=1}^{p}\left\|\beta_j\right\|_2$$

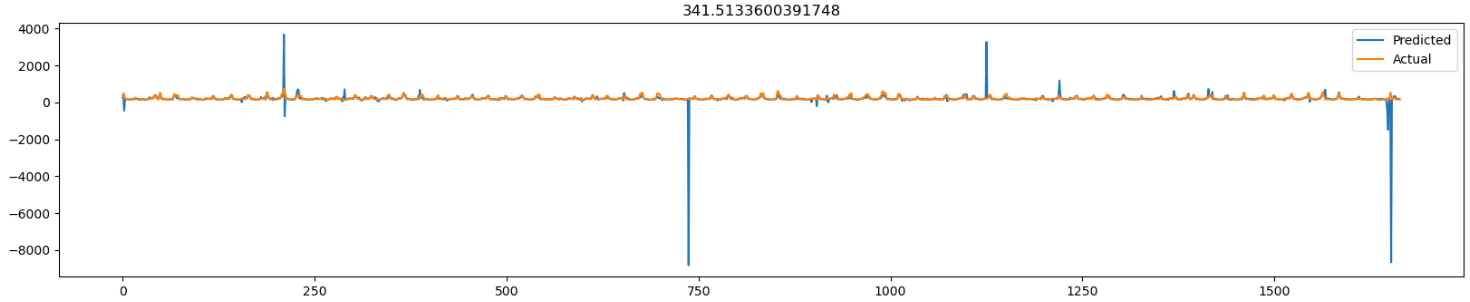
  - We need to choose λ

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

Ridge Regression: Results

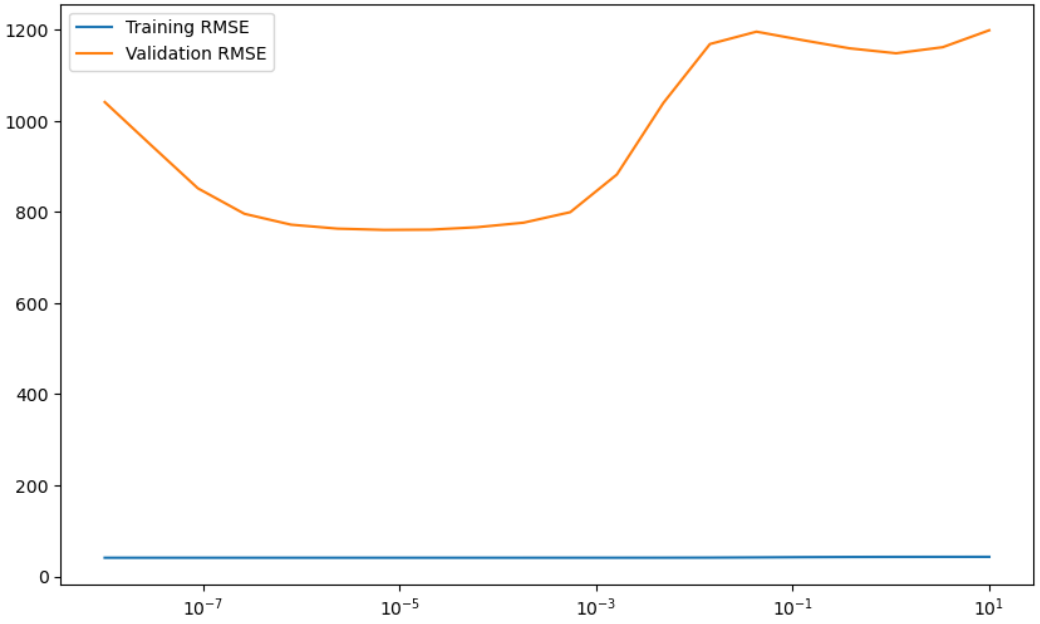
- λ perhaps should not be 1
- Instead, try a range of values
  - We'll use a log scale to produce a list of values to search as this is a bit more efficient





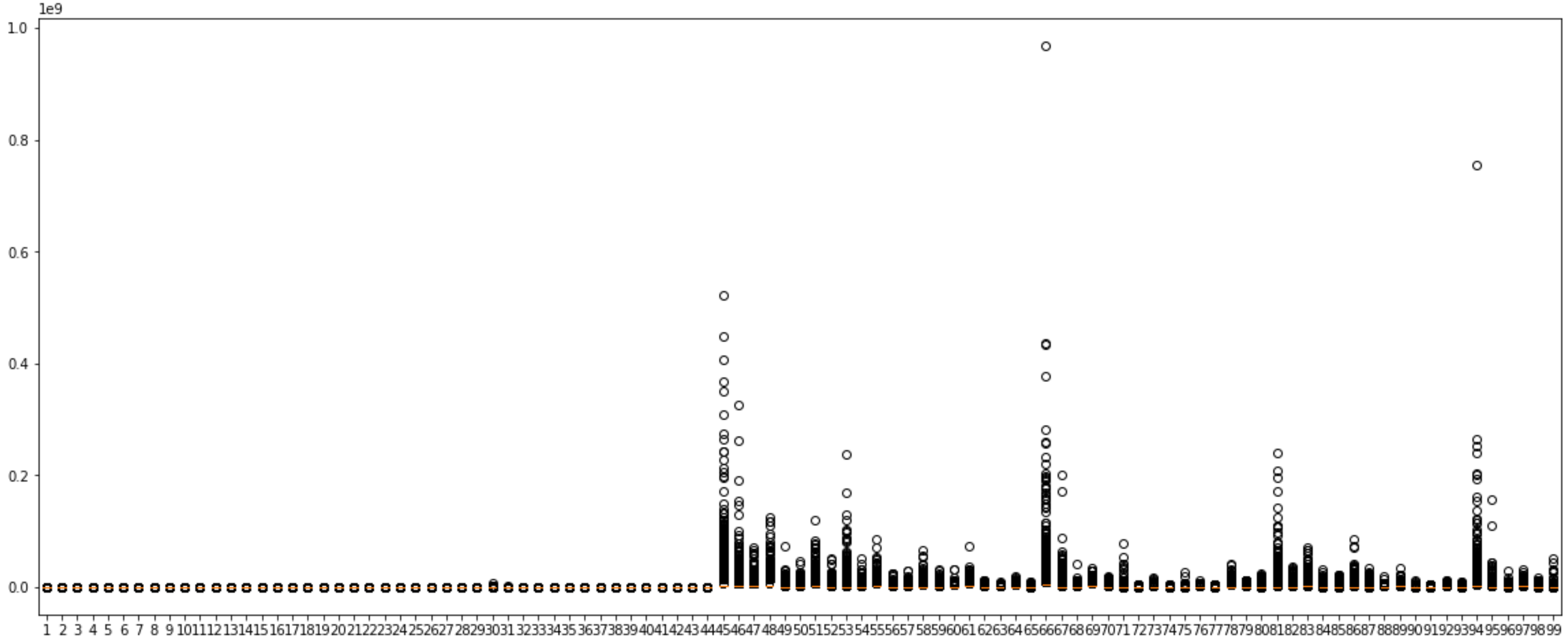
Ridge Regression: Results

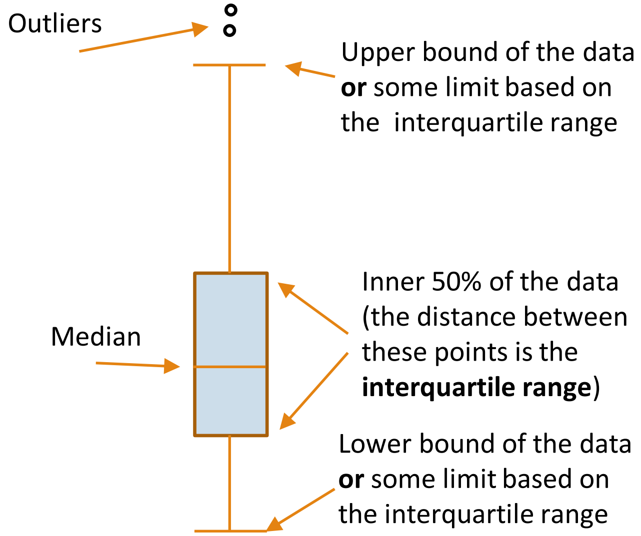
- Plotting RMSE as λ changes
- We see a drop as λ increases up to some minimum, but then the RMSE goes up again
  - After some point, we are over-regularizing



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

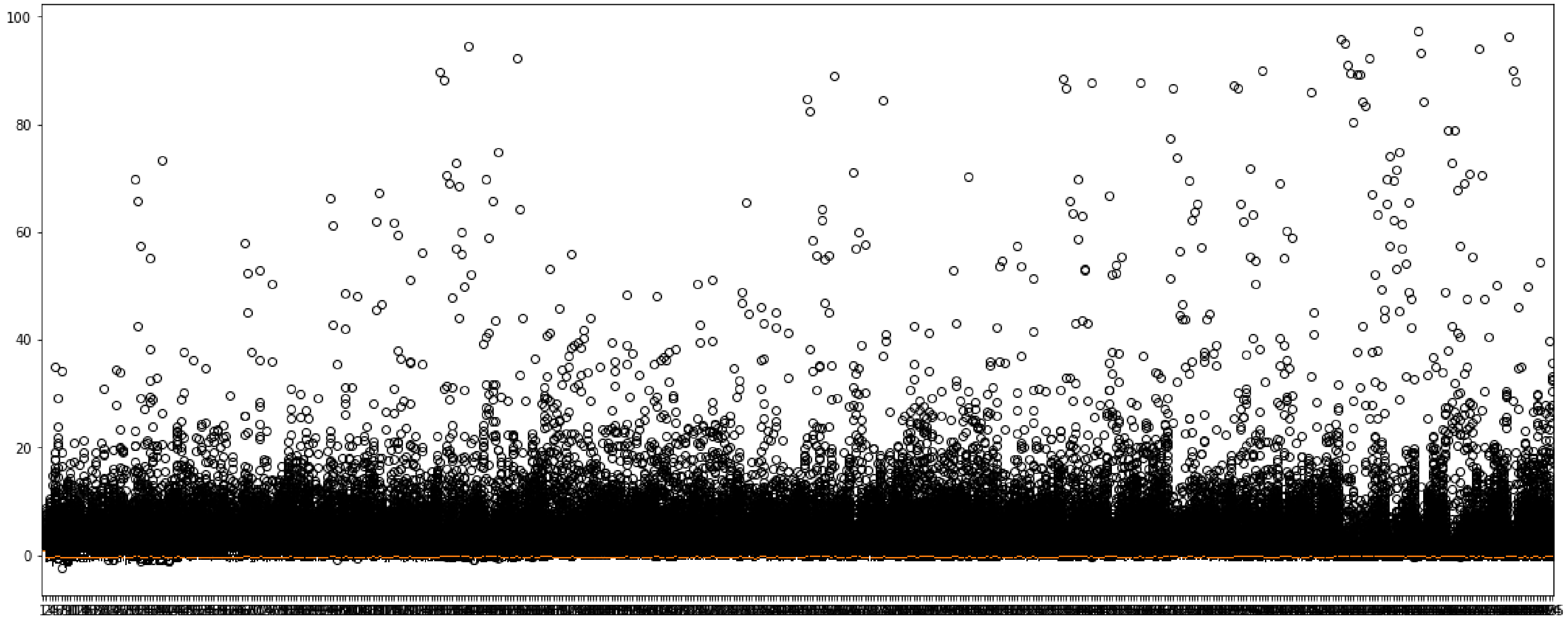
- We seek to scale all dimensions equally, so that they are all considered equally when fitting a model

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

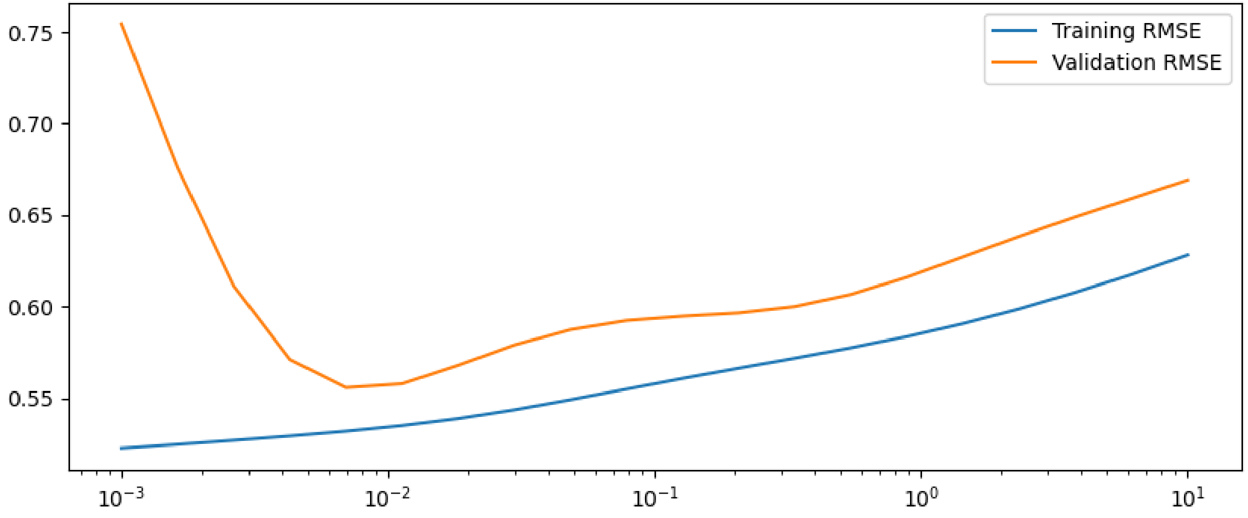
Standardised Data

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



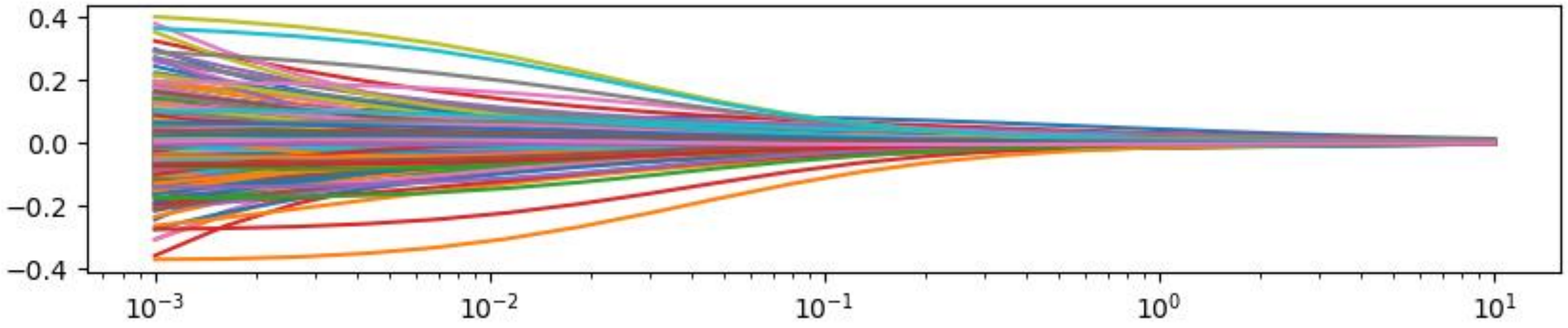
Ridge Regression with Standardised Data

- RMSE vs λ
  - We see an immediate drop as we increase λ
  - Remember, λ = 0 is least squares regression
  - Value which minimises the Validation RMSE is our best λ
    - For us, this is 0.00695
  - Training RMSE will gradually increase with λ
    - Variance vs Bias



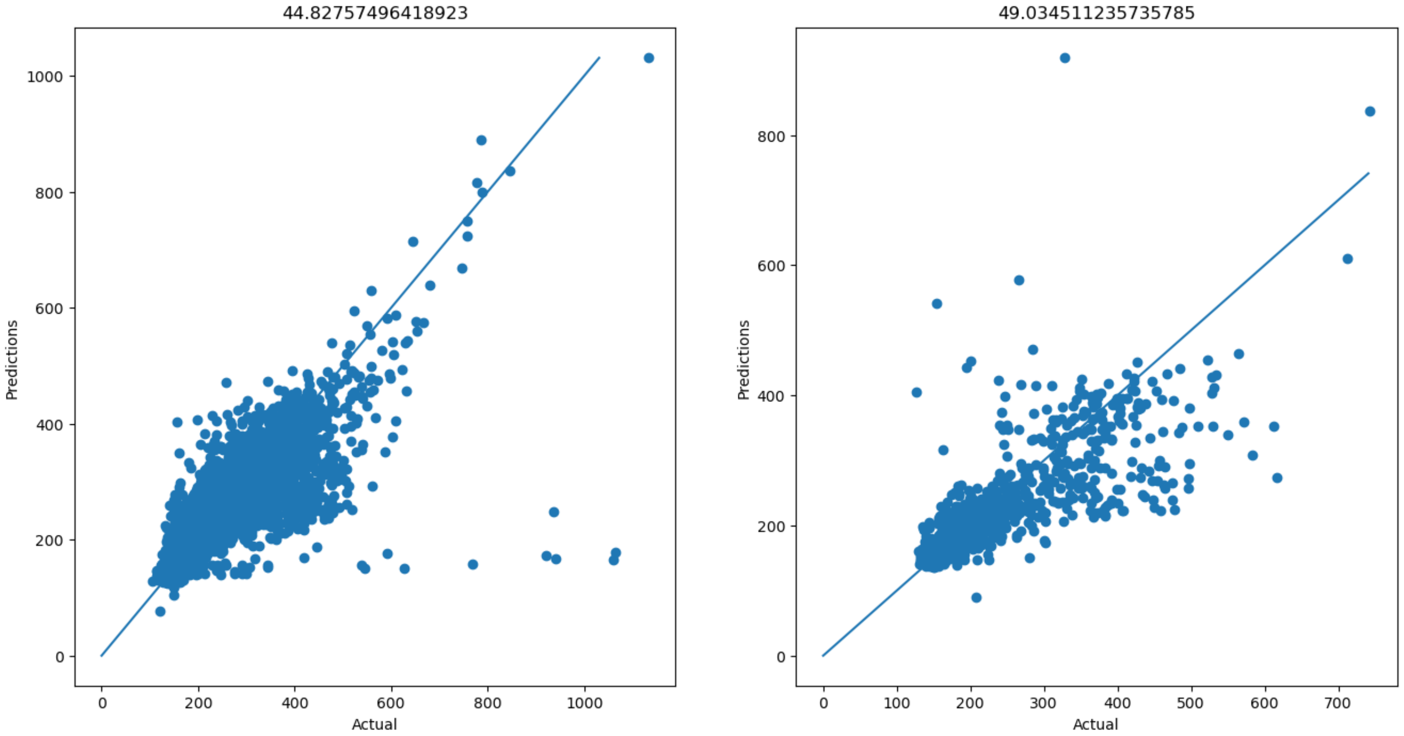
Ridge Trace Plot

- 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



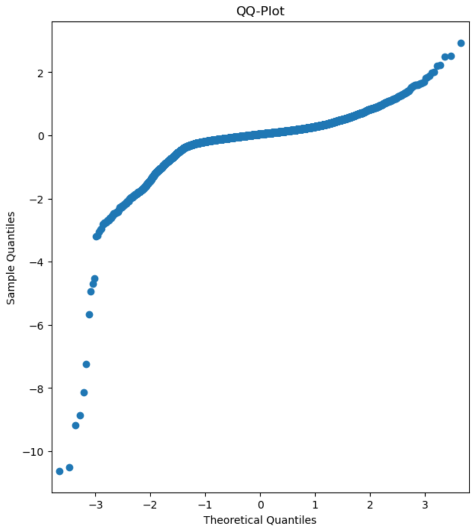
Ridge Results

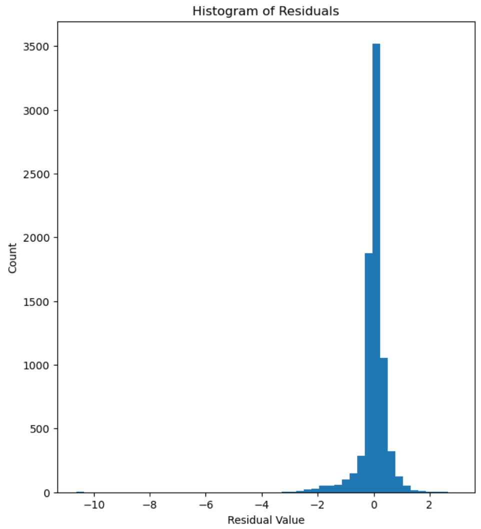
- Final Model, $\lambda = 0.00695$
  - Similar performance to original Linear model

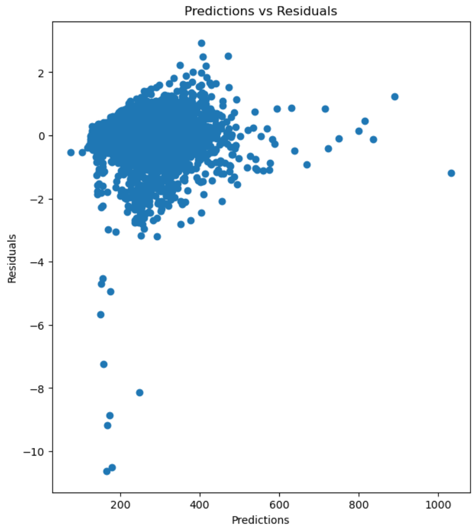


Ridge Results

- Final Model, $\lambda = 0.00695$
- $R^2 = 0.717$
  - 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

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

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

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

  - 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:

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

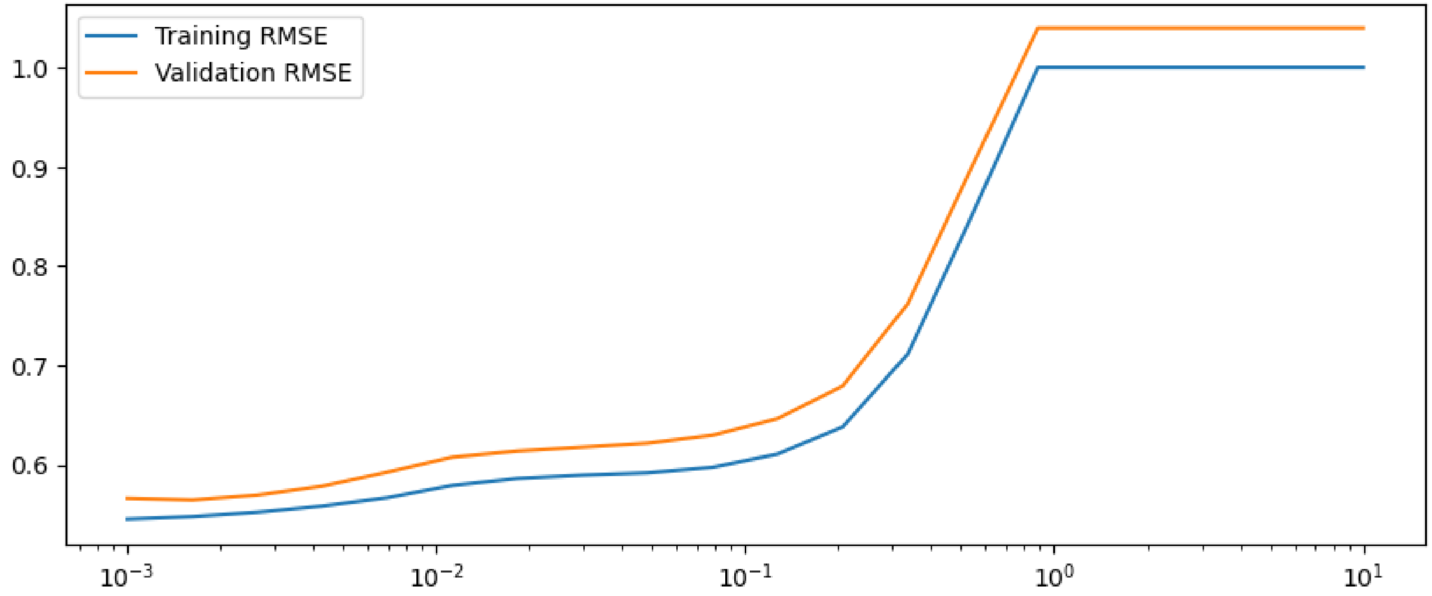
  - We need to choose λ

- As per Ridge, we'll use a range
  - Again, we’ll use a log-scale
  - Lasso typically uses a smaller λ than ridge

- We'll use standarised data from the start

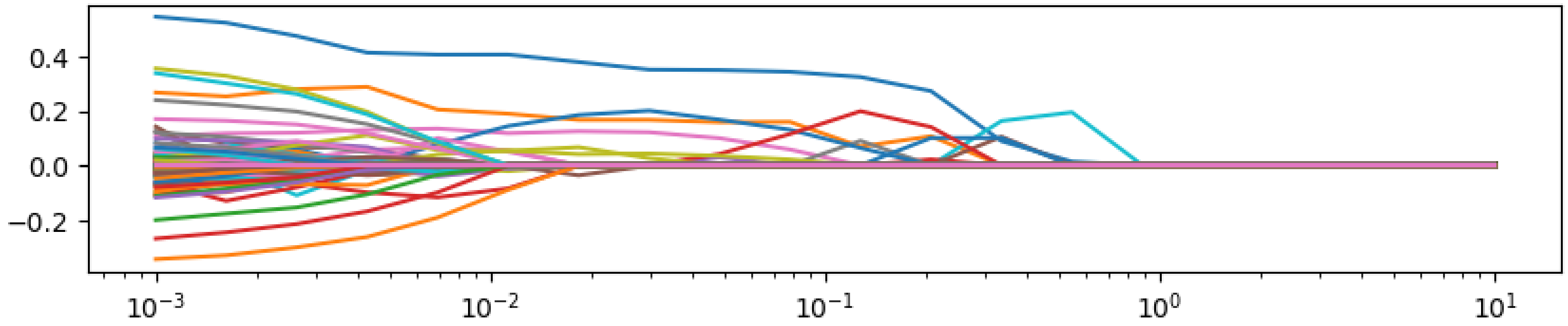
LASSO : Selecting Lambda

- Best $\lambda = 0.00162$
- Same trend as ridge
  - Training data always increases with λ
  - Validation data decreases to a minimum, then increases
    - There is a tiny drop at first there



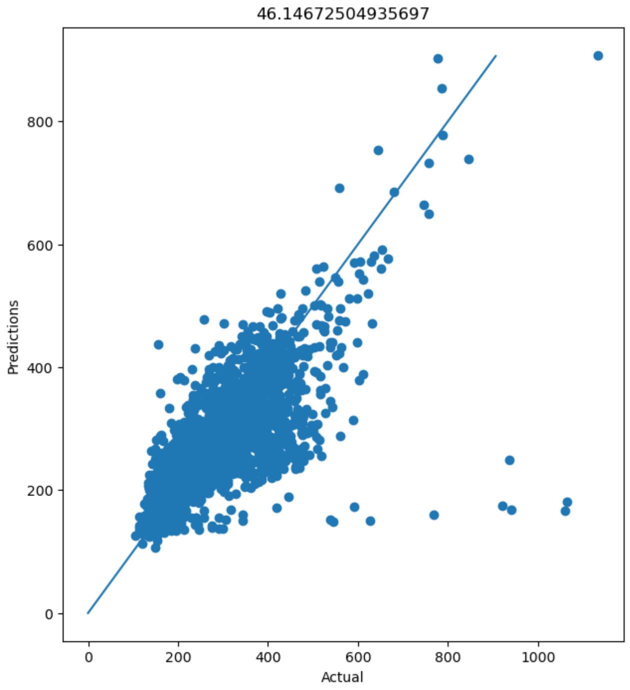
LASSO Trace Plot

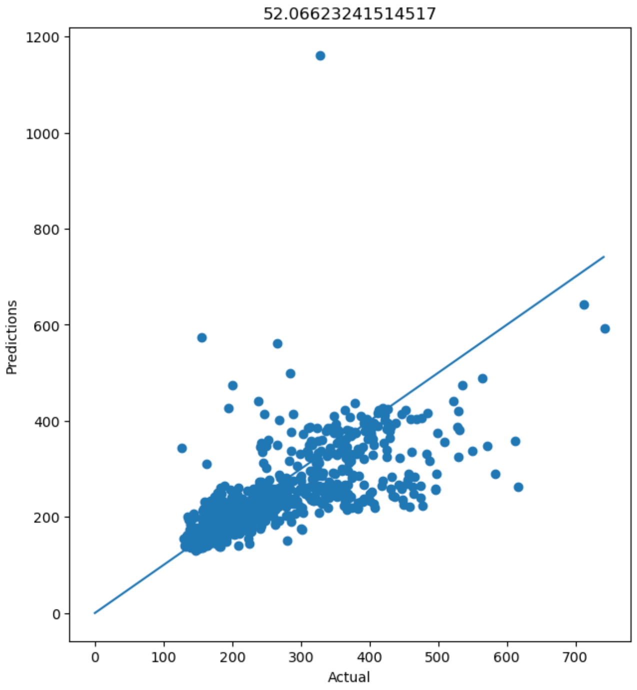
- Terms decrease in value as λ increases
  - Terms can go to 0 and be eliminated
  - At the far end of the plot, all terms are 0 (constant model)



LASSO Results

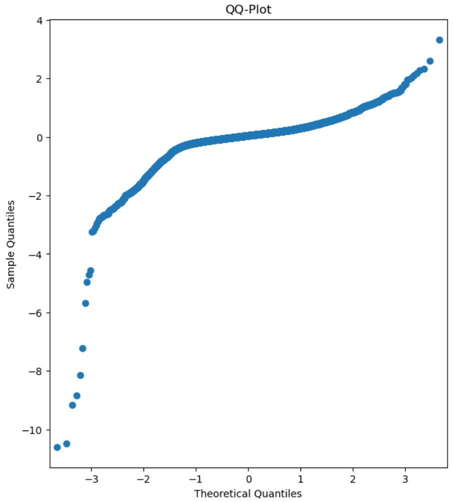
- Final Model, $\lambda = 0.00162$
  - Similar to Ridge and Linear Model
  - Final model contains 93 terms (the other 400+ are all 0)

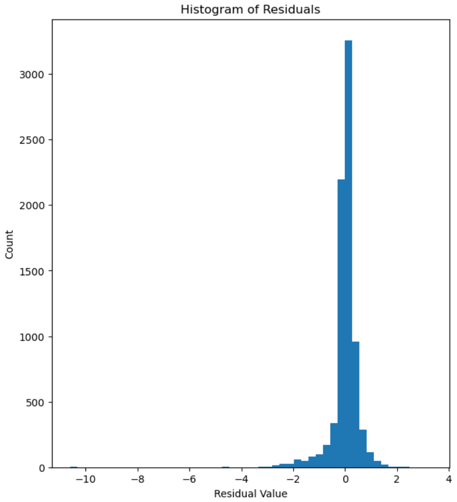


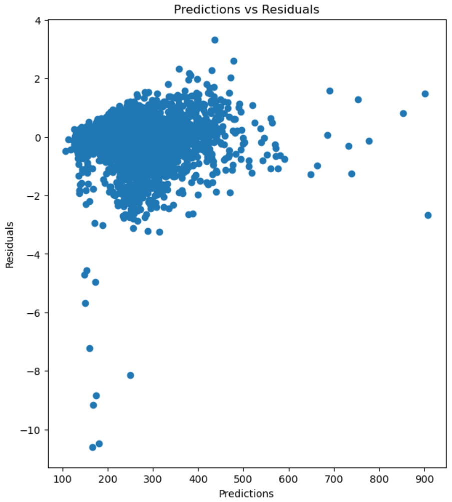


LASSO Results

- Final Model, $\lambda = 0.00162$
- $R^2 = 0.700$
- Less accurate, and a worse fit, than the ridge model
- 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
- 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 Comparing Models

- We have three datasets
  - Training
    - We're training all our models on this data
    - All residual plots are using this data
  - Validation
    - We're using this to evaluate our regularized models at each value of λ
    - We're plotting validation RMSE to select λ
  - Test
    - We're looking at our prediction accuracy (RMSE) on the test data
- Ridge regression wins here
  - Best RMSE on the test set

Is Ridge Best?

- It is in this case. Maybe.
  - We could do a more fine-grained search for λ that might change the result. Or it might not.
- In general, Ridge is not better than LASSO or vice versa. They have their own pros and cons
  - Ridge is typically faster to train
  - LASSO allows you to eliminate terms and thus simplify models
    - Faster at test time as you have fewer terms
    - But this doesn't always make it more accurate
- Unless you have other design considerations that make one impractical, try both.

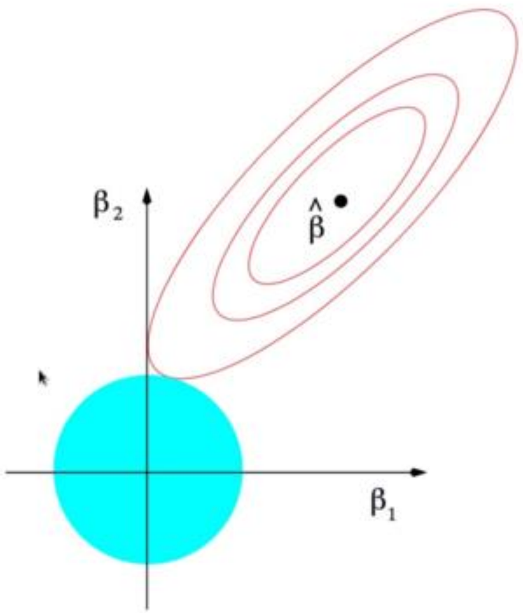
CAB420: Ridge vs LASSO

WHICH ONE?

Ridge vs Lasso

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

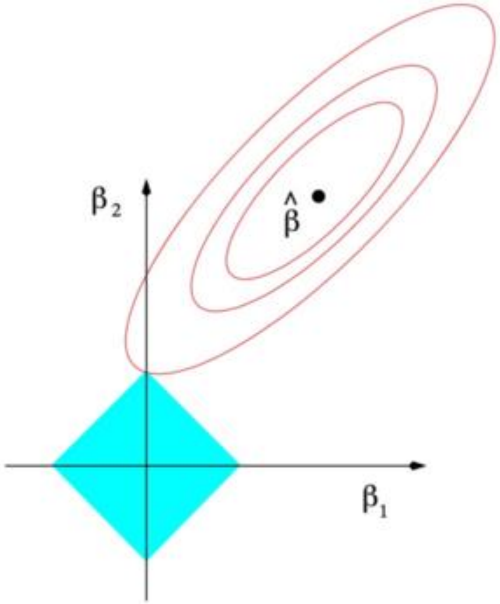
- We have a two coefficients
  - The “best solution” according to least squares is $\hat{\beta}$
  - The blue area is the constraint region for a given λ
- Ridge uses an L_2norm
  - Circular constraint region
  - Closest point on the constraint region to $\hat{\beta}$ is our ridge solution



Ridge vs **Lasso**

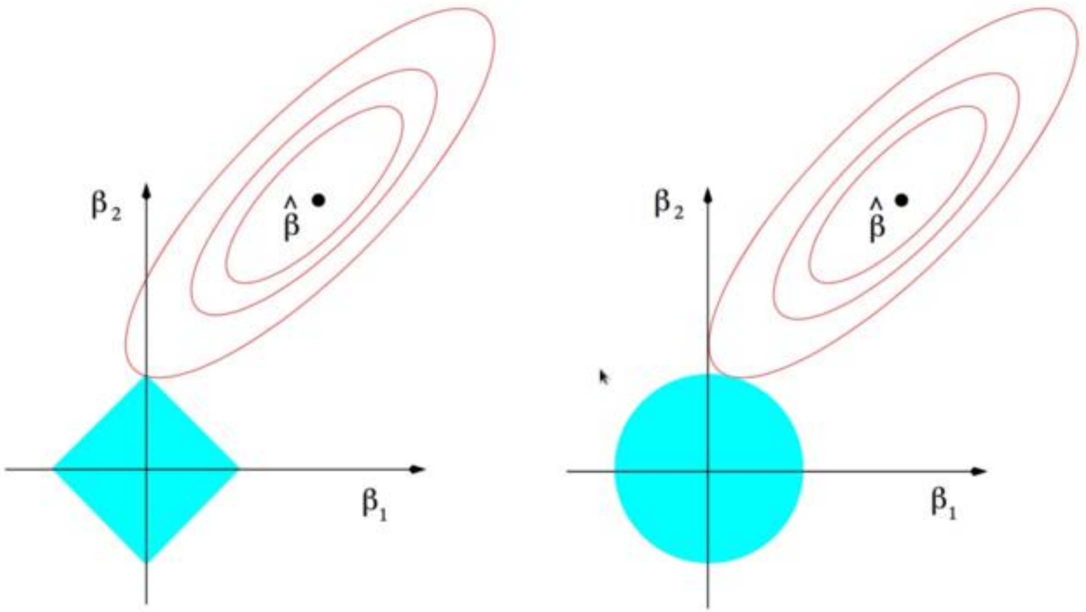
$$\sum_{i=1}^{n}\left(y_i - \sum_{j} x_{ij}\beta_j\right)^2 + \lambda \sum_{j=1}^{p}\left\|\beta_j\right\|_1$$

- 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_1 norm
  - 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***

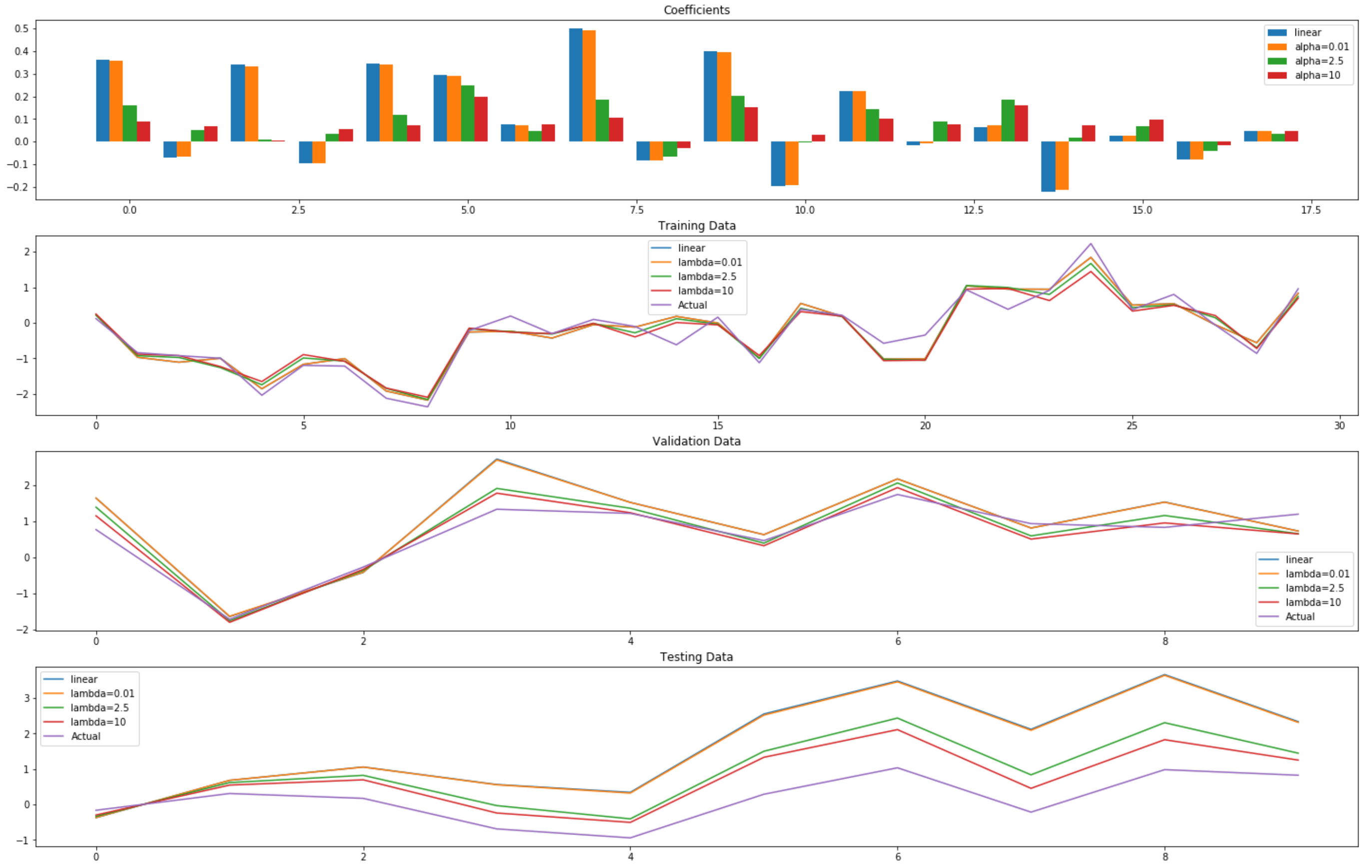
- Predict traffic times again
  - Standardised data
  - 18 predictors
  - Linear, Ridge and Lasso models
  - Training, validation and testing set all taken from different time periods
    - Split in chronological order

Linear Model

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

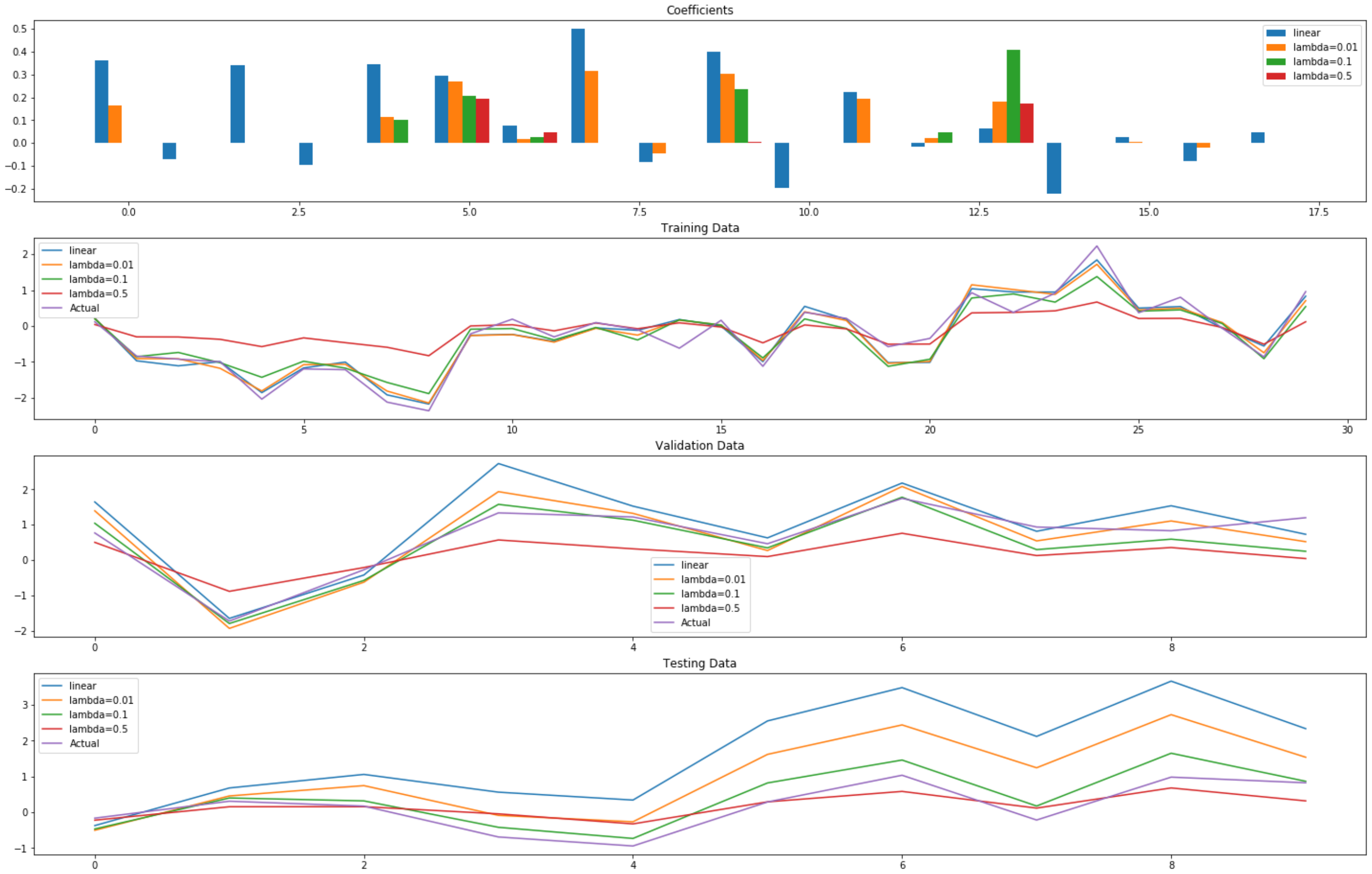
Ridge Model

- 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

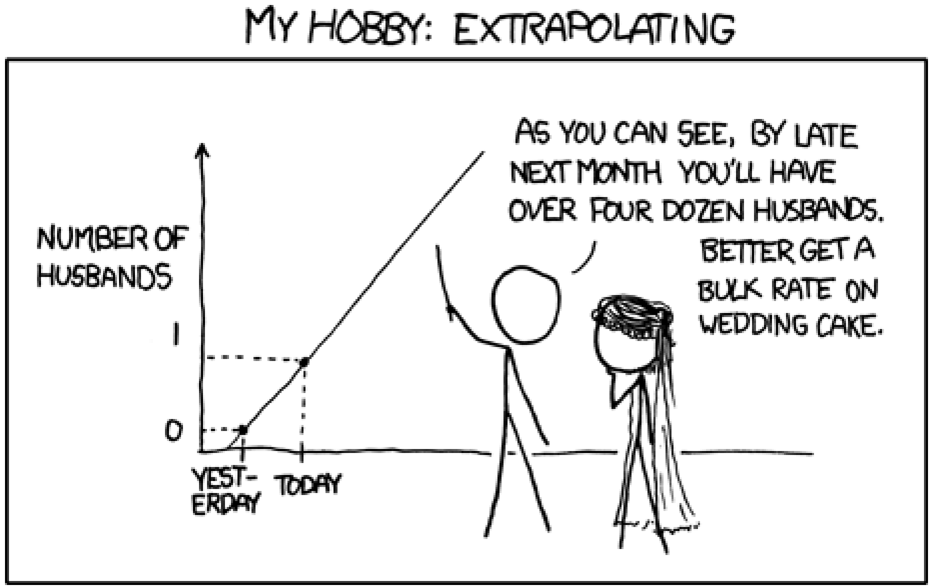
- Larger λ leads to
  - Smaller coefficients
  - Flatter prediction curves
  - Coefficents can change sign
- Coefficients can go to 0
  - Can happen at very small lambda
- Large λ will push all coefficients to 0



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



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