Feature Engineering

Machine Learning II

Master in Business Analytics and Big Data

acastellanos@faculty.ie.edu

Data Scientist:

The Sexiest Job of the 21st Century

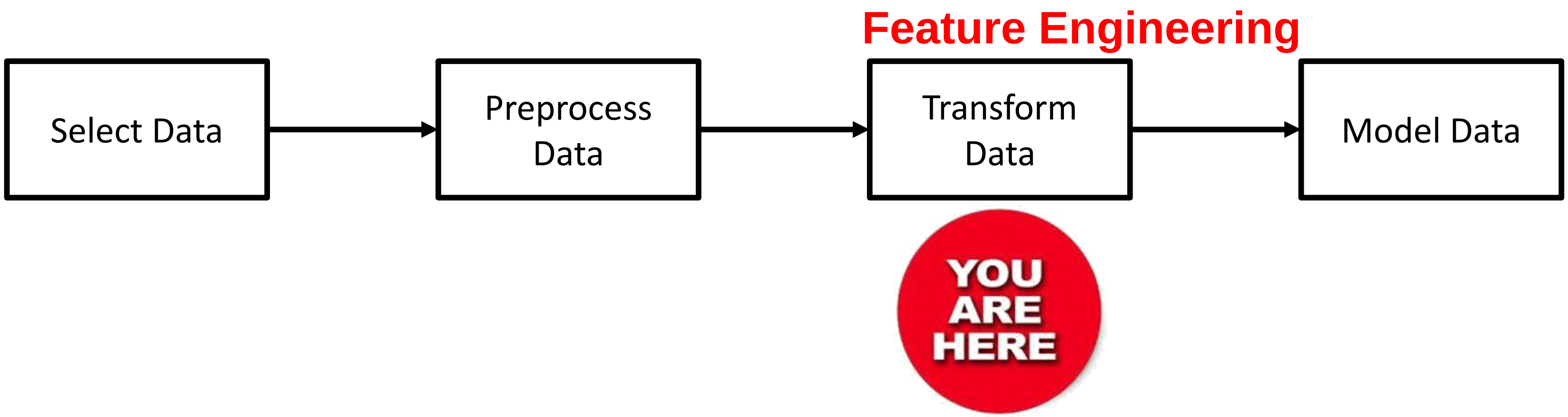
Meet the people who can coax treasure out of messy, unstructured data.

by Thomas H. Davenport and D.J. Patil

When Jonathan Goldman arrived for work in June 2006 at LinkedIn, the business networking site, the place still felt like a start-up. The company had just under 8 million accounts, and the number was growing quickly as existing members invited their friends and colleagues to join. But users weren't seeking out connections with the people who were already on the site at the rate executives had expected. Something was apparently missing in the social experience. As one LinkedIn manager put it, "It was like arriving at a conference reception and realizing you don't know anyone. So you just stand in the corner sipping your drink—and you probably leave early."

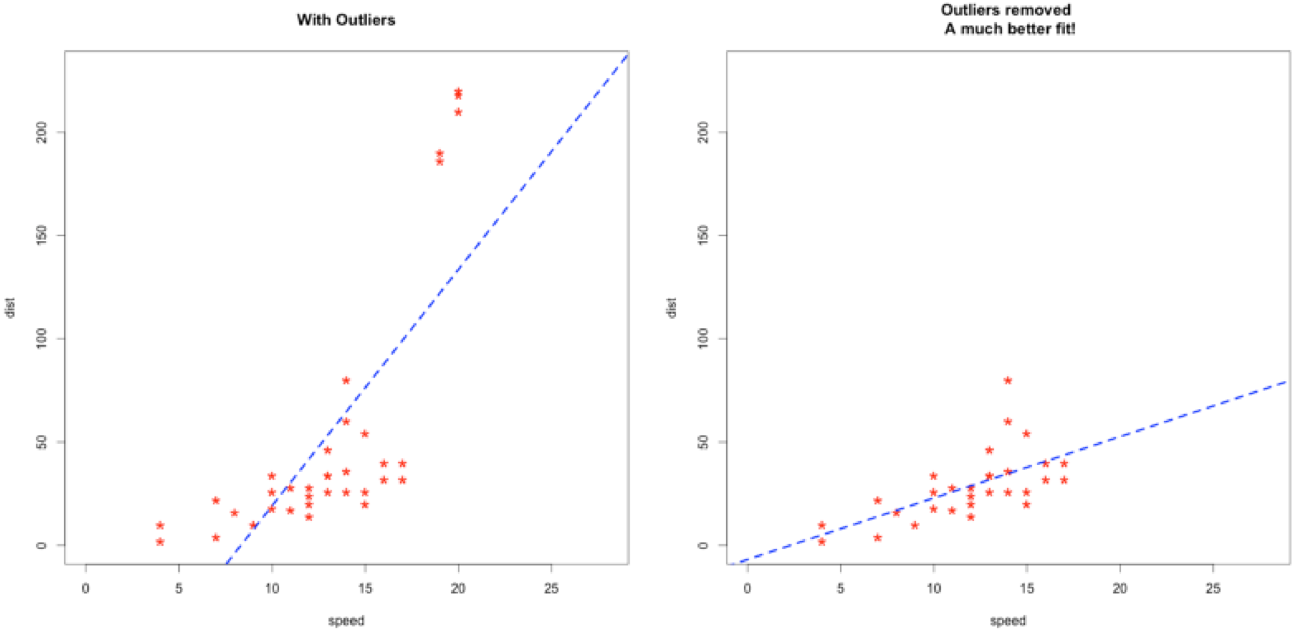
https://hbr.org/2012/10/data-scientist-the-sexiest-job-of-the-21st-century

The process of *machine learning*



Notes on Data Preprocessing

- **Outliers**
  - **Visualize** the numerical values
  - **Remove** based on some metric: Z-score

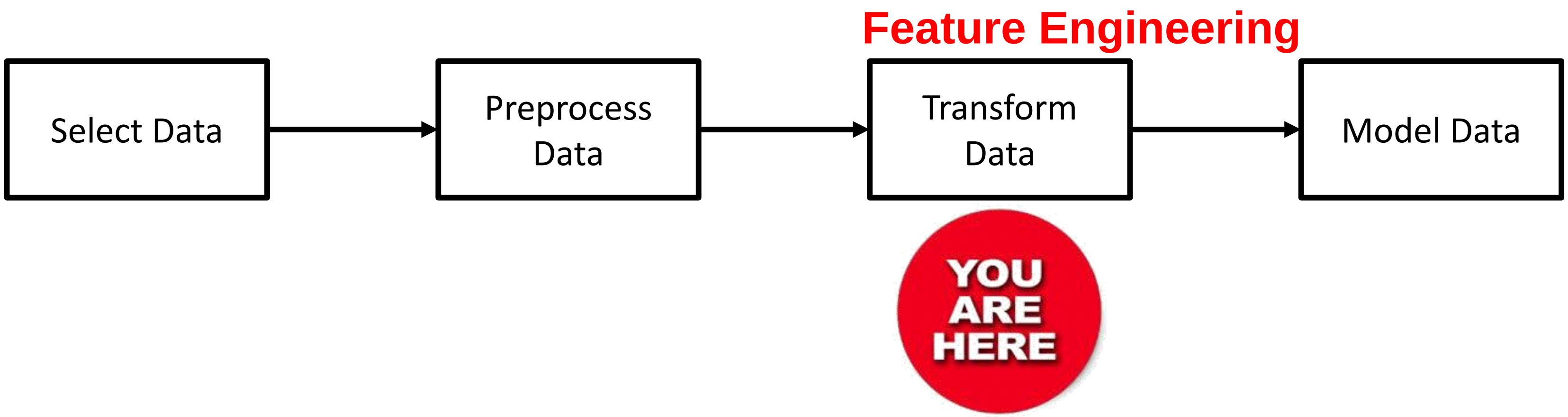


Source: https://heartbeat.fritz.ai/how-to-make-your-machine-learning-models-robust-to-outliers-44d404067d07?gi=2f654fe2301d

Notes on Data Preprocessing

- **Outliers**
  - **Visualize** the numerical values
  - **Remove** based on some metric: Z-score
- **Null Values**
  - If enough information → **Impute**
    - Numerical: Column Stats
    - Categorical: Most Common vs Closest
    - Learn the imputation: Train a model to impute the values
  - If not → **Remove**
    - Better nothing than wrong information
- **Skewness in the Target Variable**
- **Bucketization**
  - [0-10] → Low, Medium, High

The process of *machine learning*



What is a *feature*

Informative

Discriminating

Independent

Individual measurable property of a phenomenon being observed.

What is *feature engineering*?

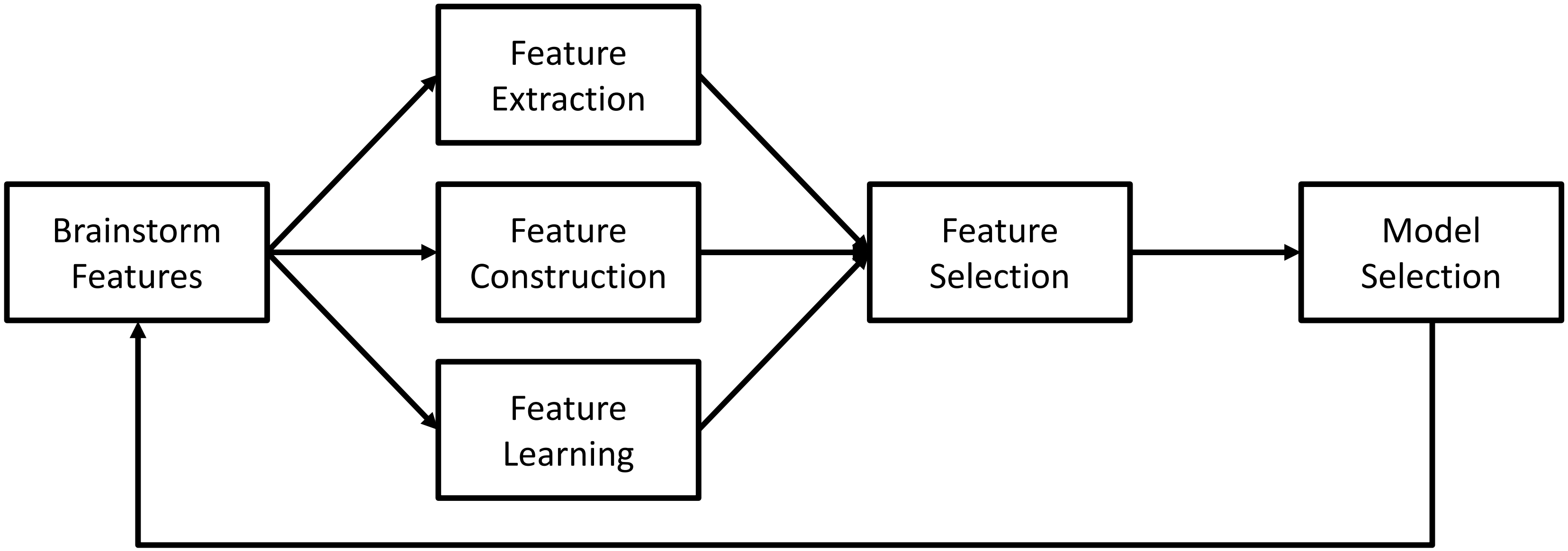
- **Get the most out of the data** for your algorithms to work with to **get the best posible results from a predictive model** on unseen data.
- **Raw data → features**
  - Better represent the underlying problem to the predictive models.

Feature Engineering Importance

- **Simpler & more Flexible**
  - Less complex models, easier to understand and maintain.
  - Working closer to the underlying problem and better representation of all the available data.

- **Better Results**
  - **Prof. Andrew Ng:** *"Coming up with features is difficult, time-consuming, requires expert knowledge. 'Applied machine learning' is basically feature engineering."*
  - **Winner of the Flights Quest challenge in Kaggle: "***The algorithms we used are very standard for Kagglers. We spent most of our efforts in feature engineering."*

Iterative process of feature engineering



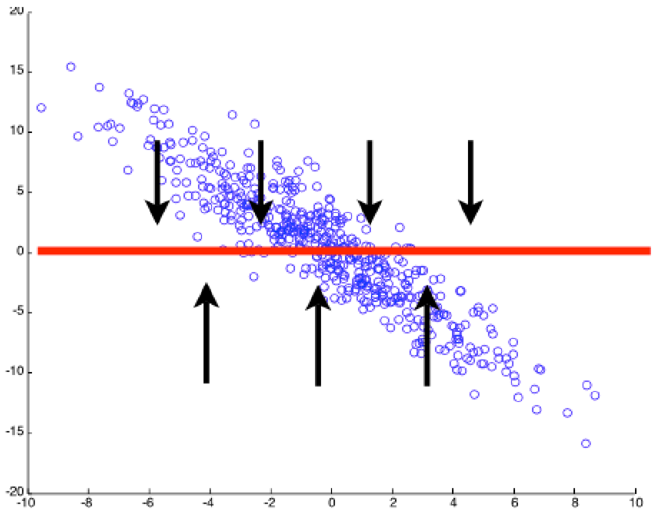
Feature Creation

- **Creating derived variables**
  - Ratios: Credit card Sales / Marketing spend
  - Time Differences
  - More Creative variables: Infer age from treatment (Ms., Mrs.)
  - https://towardsdatascience.com/automated-feature-engineering-in-python-99baf11cc219
- **Creating dummy variables**
  - Categorical variable into numerical variables
- **Variable Transformation**
  - Change the scale
  - Complex non-linear relationships into linear relationships
  - Symmetric distribution

Feature selection vs. Dimensionality reduction

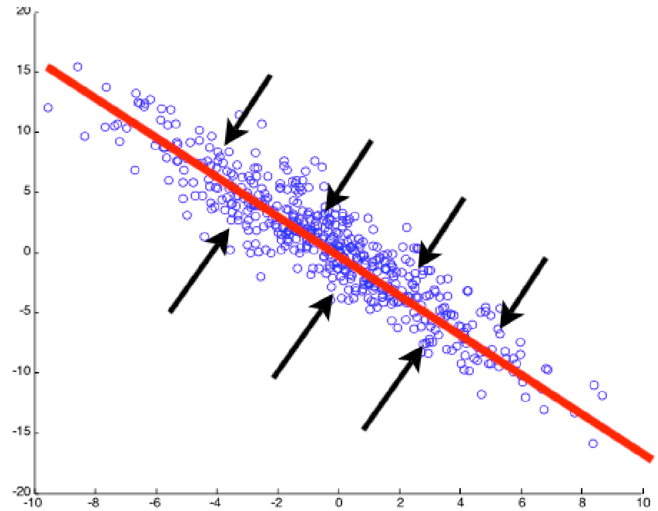
Feature Selection

Remove features by **projecting data onto lower-dimensional subspace**



Dimensionality Reduction

Remove features by creating **new combinations of attributes**.



Methods for feature (and model) selection

- **Filter Methods**
- **Wrapper Methods**
- **Embedded Methods (Regularization)**

Methods for feature (and model) selection

- **Filter Methods**
- Wrapper Methods
- Embedded Methods (Regularization)

Feature Importance (Filtering)

- **Estimate the usefulness of features**.
  - Features are ranked by a **'score'**. Features with highest scores are selected and the rest are ignored.
  - The scores can also provide useful information to extract or construct new features.
- **A feature may be important if it is highly correlated with the variable we want to predict.**
- More **complex predictive modeling algorithms perform feature importance and selection internally** while constructing their model.
  - Random Forest, MARS (multivariate adaptive regression splines), Gradient Boosted Machines.

Chi Squared test

When?

Pairs of **categorical** variables from a single population (simple random sampling) to determine whether there is a **significant association** between the two variables (knowing one of them help me to predict the other one).

How to do it?

- Select a **significance level**
- **Degrees of freedom** $DF = (r-1)*(c-1)$
- **Calculate $\chi2$**
- Calculate **p-value:** the probability of observing a $\chi2$ value with DF of under the null hypothesis (Variable A and B are **independent)**
- Comparing the p-value to the **significance level** set, and rejecting the null hypothesis (Variable A and B are **independent)** when the **p-value < significance level**.

$$\chi^2 = \sum_{r,c} \frac{\left(O_{r,c} - E_{r,c}\right)^2}{E_{r,c}}$$

Chi Squared test

Observed

	favor	indifferent	opposed	total
democrat	138	83	64	285
republican	64	67	84	215
total	202	150	148	500

Expected

	favor	indifferent	opposed	total
democrat	285·202/500 = 115.14	285·150/500 = 85.5	285·148/500 = 84.36	285
republican	215·202/500 = 86.86	215·150/500 = 64.5	215·148/500 = 63.64	215
total	202	150	148	500

$= (138-115.14)^2/138 + (83-85.5)^2/83 + (64-84.36)^2/64 + (64-86.86)^2/64 + (67-64.5)^2/67 + (84-63.64)^2/84 = \mathbf{22.2}$

= 22.2, df = 2, p-value = 0.000015

Source: http://stattrek.com/chi-square-test/independence.aspx?Tutorial=AP

Example (in R)

- Set the significance level
- Set the null hypothesis
- Analyze the data
- Interpret results

Pros and Cons

Pros:

- Very fast
- Simple to apply

Disadvantages:

- **Doesn’t take into account interactions between features:** Apparently useless features can be useful when grouped with others

Information Gain

- How **important the inclusion or exclusion of a particular feature** is in predicting the **correct class.**

- It estimates the **'amount of information'** that is shared between two random variables say X and Y:
  - Knowing **X**, would reduce the uncertainty of **Y**?
  - Knowing the **feature** would reduce the uncertainty of the **class**?

- **Entropy:** How **unpredictable** is an attribute.

Information Gain

How to compute it, for an attribute with *k* different categorical values:

Sex
Female
Male
Male
Male
Male
Female
Male
Female
Male
Male

$$H(class) = -\sum_{i=1}^{k} P(e_i) log_2(P(e_i))$$

$$\boldsymbol{H(Sex)} = 3/10 \cdot \log 2(3/10) + (7/10) \cdot \log 2(7/10) = \mathbf{0.8812909}$$

Information Gain (cont.)

To compute the IG of each variable, we **compute the entropy associated to the class we want to predict (Sex)**.

For simplicity, lets bin pulse a bit more (<25, 25-50, 50-75, >75)

Sex	npulse
Female	100
Male	25
Male	100
Male	25
Male	50
Female	75
Male	100
Female	75
Male	75
Male	100

$$H(pulse = 25) \ = \ (2/2) \cdot \log2(2/2) \ + \ (0/2) \cdot \log(0/2) \ = \ 0$$

$$H(pulse = 50) \ = \ (1/1) \cdot \log2(1/1) \ + \ (0/1) \cdot \log(0/1) \ = \ 0$$

$$H(pulse = 75) \ = \ (1/3) \cdot \log2(1/3) \ + \ (2/3) \cdot \log(2/3) \ = \ 0.918$$

$$H(pulse = 100) \ = \ (3/4) \cdot \log2(3/4) \ + \ (1/4) \cdot \log(1/4) \ = \ 0.811$$

$$H(pulse) = (2/10)H_{25} \ + \ (1/10)H_{50} \ + \ (3/10)H_{75} \ + \ (4/10)H_{100} = 0.599$$

Information Gain (cont.)

- What Information Gain is obtained by the variable ‘pulse’, to predict Sex?

$$IG = H_{SEX} - H_{PULSE} = \mathbf{0.8812909 - 0.599 = 0.282}$$

- Use this gain to compare how this variable helps the model to make the prediction, and **filter out those with lower information gain.**

Methods for feature (and model) selection

- Filter Methods
- **Wrapper Methods**
- Embedded Methods (Regularization)

Model Selection (Wrapper)

- **Search problem**: different combinations are prepared, evaluated and compared to other combinations.
  - **Evaluate each combination** of features and assign a score based on model accuracy.
  - **Methodical:** best-first search
  - **Stochastic:** random hill-climbing algorithm
  - **Heuristics:** forward and backward passes to add and remove features.
- **Examples:** Recursive feature elimination algorithm

Example: Best Subset Selection

Intuition: Fit a least squares regression for each possible combination of the predictors (p).

Algorithm:

for (k = 1 … p):

1. Fit all possible $\binom{p}{k}$ models with *k* predictors.
2. Select the one ($\mathcal{M}_k$) with lower RSS or higher R^2
3. Select the best model using *cross validated prediction error*.

Problems

- Computationally **expensive** when p is large
- **Overfitting** when p is large: ***the larger the search space, the higher the chance of finding models that look good on the training data***, even though they might not have any predictive power on future data.

For both of these reasons, **stepwise methods**, which explore a far more restricted set of models, are **attractive alternatives to best subset selection**.

Forward Stepwise Selection

Begins with no predictors, and adds the predictors that adds greatest additional improvement to the model, one-at-a-time, until all of the predictors are in the model.

Algorithm:

1. $\mathcal{M}_0$ be the null model with no predictors
2. for $(\mathrm{k} = 0, \dots, \mathrm{p} - 1)$:
    1. Consider all p-k models in $\mathcal{M}_k$ that add a single different predictor
    2. Select the one ($\mathcal{M}_{k+1}$) with lower RSS or higher R^2
3. Select the best model using **cross validated prediction error.**

Pros and Cons

Pro: Computationally more efficient

- Each iteration ($\mathrm{k} = 0 \ldots \mathrm{p} - 1$) evaluates $p - k$ models:

$$1 + \sum_{k=0}^{p-1}(p-k) = 1 + p(p+1)/2 \text{ models.}$$

Con: Best possible model not guaranteed.

- Let p=3, if best possible model with 1 predictor contains X1, and best possible 2 predictors model contains X2 and X3, Forward Stepwise selection will fail.

Backward stepwise selection

- **Same pros/cons that in Forward Stepwise Selection**

Algorithm:

1. $\mathcal{M}_p$ be the full model with all predictors
2. for (k = p … 1):
   1. Consider all k models in $\mathcal{M}_k$ that contain all but one of the predictors in $\mathcal{M}_k$
   2. Select the one ($\mathcal{M}_{k-1}$) with lower RSS or higher R^2
3. **Select the best model using *cross validated prediction error*.**

Feature Engineering

How?

- How do we **“Select the best model”?**
  - **R^2 and RSS are measuring training error.**
  - We want to choose a model with **lower test error** (**R^2 and RSS are not valid**)
  - **Two possible alternatives**
    - **Indirectly Estimate test error**
    - **Directly estimate test error**

Indirect estimate

- **The idea is:**
  - MSE (RSS/n) underestimates the test error
  - So, we adjust the training error to add how much bias the overfitting adds, as a penalty.

- **Evaluate the variance of the error (σ^2), to measure how much our observations deviate from the fitted surface.**
  - But is unknown, so must estimate

$$\hat{\sigma}^2 = \frac{RSS}{n-p}$$

Indirect techniques

Penalty increases as the **number of predictors** (p) increases.

As estimators for MSE, **the lower the better.**

BIC places heavier penalty on models with many variables

$$\begin{cases} C_p = \frac{1}{n}(RSS + 2p\hat{\sigma}^2) \\ AIC = \frac{1}{n\hat{\sigma}^2}(RSS + 2p\hat{\sigma}^2) \\ BIC = \frac{1}{n}(RSS + \log(n)\,p\hat{\sigma}^2) \end{cases}$$

The larger the better.

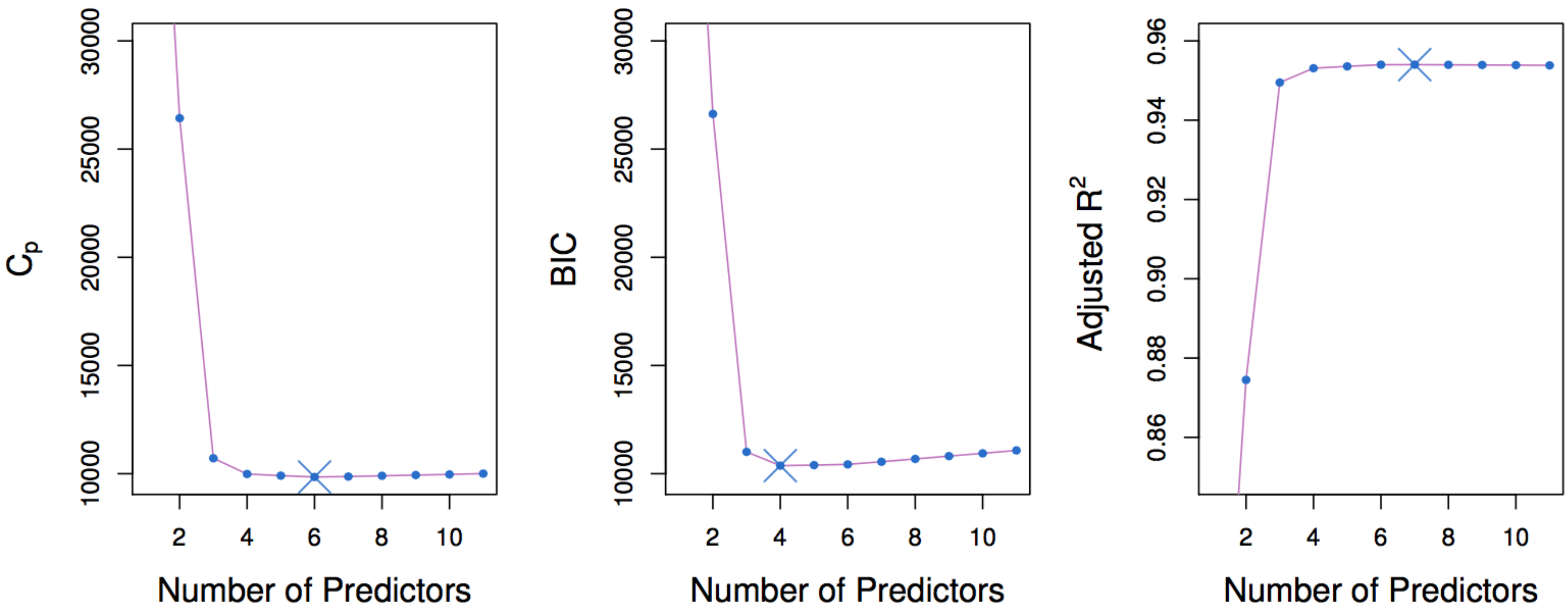
Minimizing numerator

The model with largest will have only correct variables and no noise variables.

Increases only if the **new term improves the model more than would be expected by chance**.

$$Adj.R^2 = 1 - \frac{RSS/(n-p-1)}{TSS/(n-1)}$$

Credit card data sample



Direct techniques (Wrapper)

- Directly measuring the estimate error over **a cross validation** data set (cross validation error) is **always better than using indirect techniques**
  - We don't have to estimate the error variance.
  - Makes fewer assumptions over the underlying model
  - **Computational resources are no longer a problem** to perform cross validation.

- **Cross validation k-fold (k=10) with 60%-20% split between training and validation sets.**

Methods for feature (and model) selection

- Filter Methods
- Wrapper Methods
- **Embedded Methods (Regularization)**

Embedded: Regularization

- Learn which features best contribute to the accuracy of the model **while the model is being created.**
- Adds a ‘**penalty**’ as they introduce **additional constraints into the optimization** that **bias the model toward lower complexity** (fewer coefficients).
- Examples of regularization algorithms are the LASSO, Elastic Net and Ridge Regression.

Shrinkage methods (Regularization)

With least squares and subset selection, we tried to find the optimal subset of predictors that minimize the RSS.

$$RSS = \sum_{i=1}^{n}\left(y_i - \beta_0 - \sum_{j=1}^{p}\beta_j x_{ij}\right)^2$$

Shrinkage methods try to **minimize RSS** by adding a penalty to the equation above, that rewards coefficients that are close to zero.

$$\sum_{i=1}^{n}\left(y_i - \beta_0 - \sum_{j=1}^{p}\beta_j x_{ij}\right)^2 + \lambda\sum_{j=1}^{p}\beta_j{}^2$$

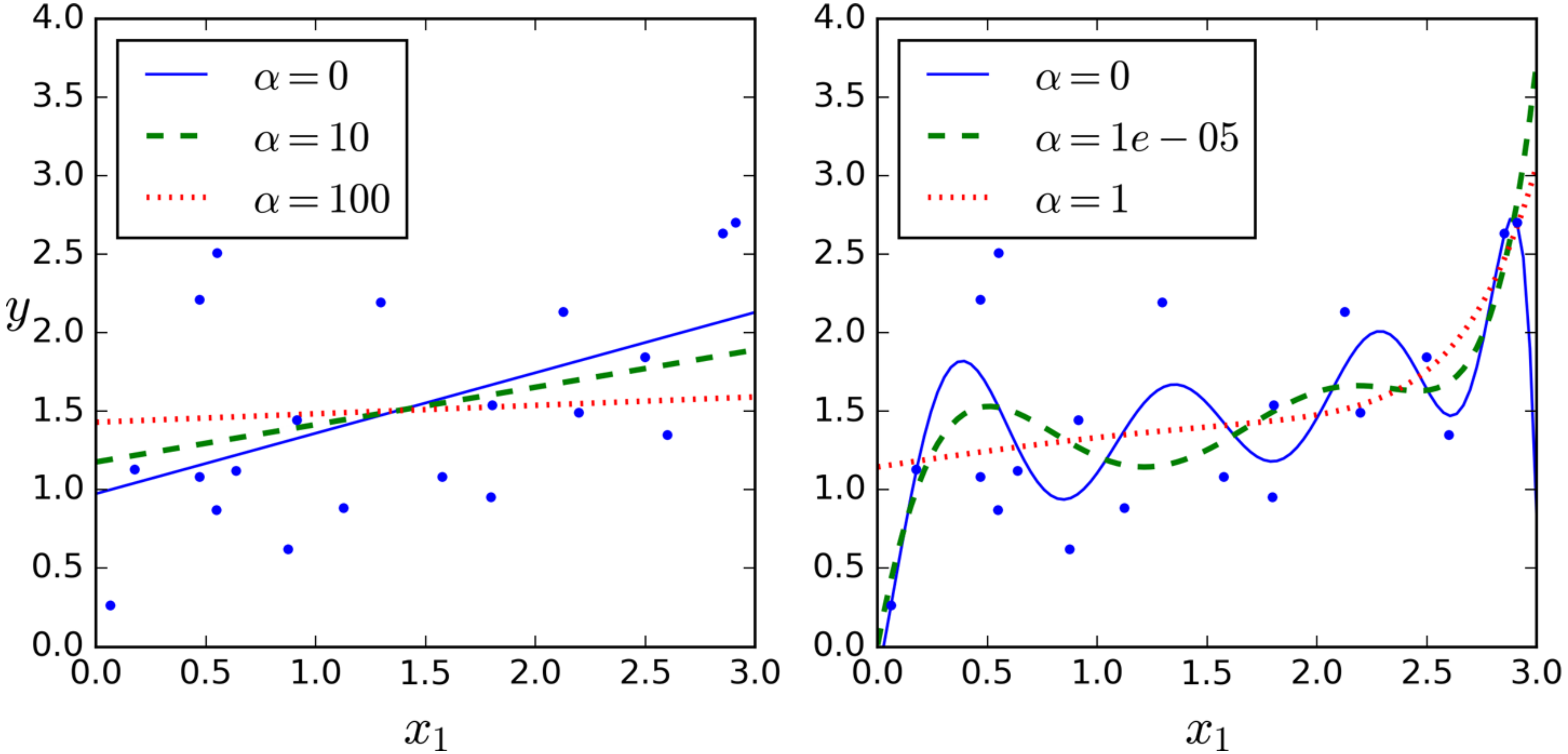
AKA: Regularization, in Machine Learning dialect.

Ridge Regression

$$\sum_{i=1}^{n}\left(y_i-\beta_0-\sum_{j=1}^{p}\beta_j x_{ij}\right)^2+\lambda\sum_{j=1}^{p}\beta_j{}^2$$

Second term is the shrinkage penalty.
λ (tuning parameter) controls the impact.

Ridge Regression



Source: https://www.oreilly.com/library/view/hands-on-machine-learning/9781491962282/ch04.html

Ridge Regression example

Computer machines dataset

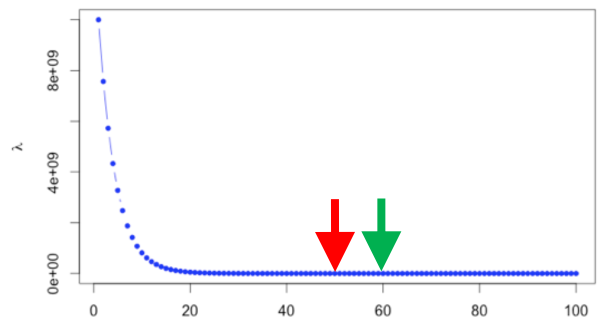
- A simple regression model can be used to predict ERP.
- Let’s apply ridge regression
- Determine the optimal lambda
- Use cross validation to predict and tune Ridge Regression.

1	Vendor Name	Many vendor names
2	Model Name:	many unique symbols
3	MYCT:	machine cycle time in nanoseconds (integer)
4	MMIN	minimum main memory in kilobytes (integer)
5	MMAX:	maximum main memory in kilobytes (integer)
6	CACH:	cache memory in kilobytes (integer)
7	CHMIN:	minimum channels in units (integer)
8	CHMAX:	maximum channels in units (integer)
9	PRP:	published relative performance (integer)
10	ERP:	estimated relative performance from the original article (integer)

```
myurl="http://archive.ics.uci.edu/ml/machine-learning-databases/cpu-performance/machine.data"
machine <- read.csv(url(myurl), header=FALSE)
attach(machine)
x <- model.matrix(V10 ~ V3+V4+V5+V6+V7+V8)
y <- machine$V10
library(glmnet)
```

```
grid = 10^seq(10,-2,length=100)
ridge.mod = glmnet(x, y, alpha=0, lambda=grid)
dim(coef(ridge.mod))

## [1]   8 100
```

```
ridge.mod$lambda[50]

## [1] 11497.57
```




```
coef(ridge.mod)[,50]

##  (Intercept)   V3       V4       V5       V6       V7       V8
## 94.288600246  -0.00215 0.00042  0.00015  0.03165  0.17702  0.04526
```

```
coef(ridge.mod)[,60]

##  (Intercept)   V3       V4       V5       V6       V7       V8
## 47.34178      -0.01488 0.00443  0.00162  0.29936  1.58925  0.44299
```

Split the dataset into 50% of random samples for training, and the other 50% for tests.

```
set.seed(1)
train=sample(1:nrow(x), nrow(x)/2)
test=(-train)
y.test=y[test]
```

Fit the model with a training set, and measure the MSE with test set

```
ridge.mod=glmnet(x[train,], y[train], alpha=0, lambda=grid, thresh=1e-12)
ridge.pred=predict(ridge.mod, s=4, newx=x[test,])
mean((ridge.pred - y.test)^2)

## 4633.87
```

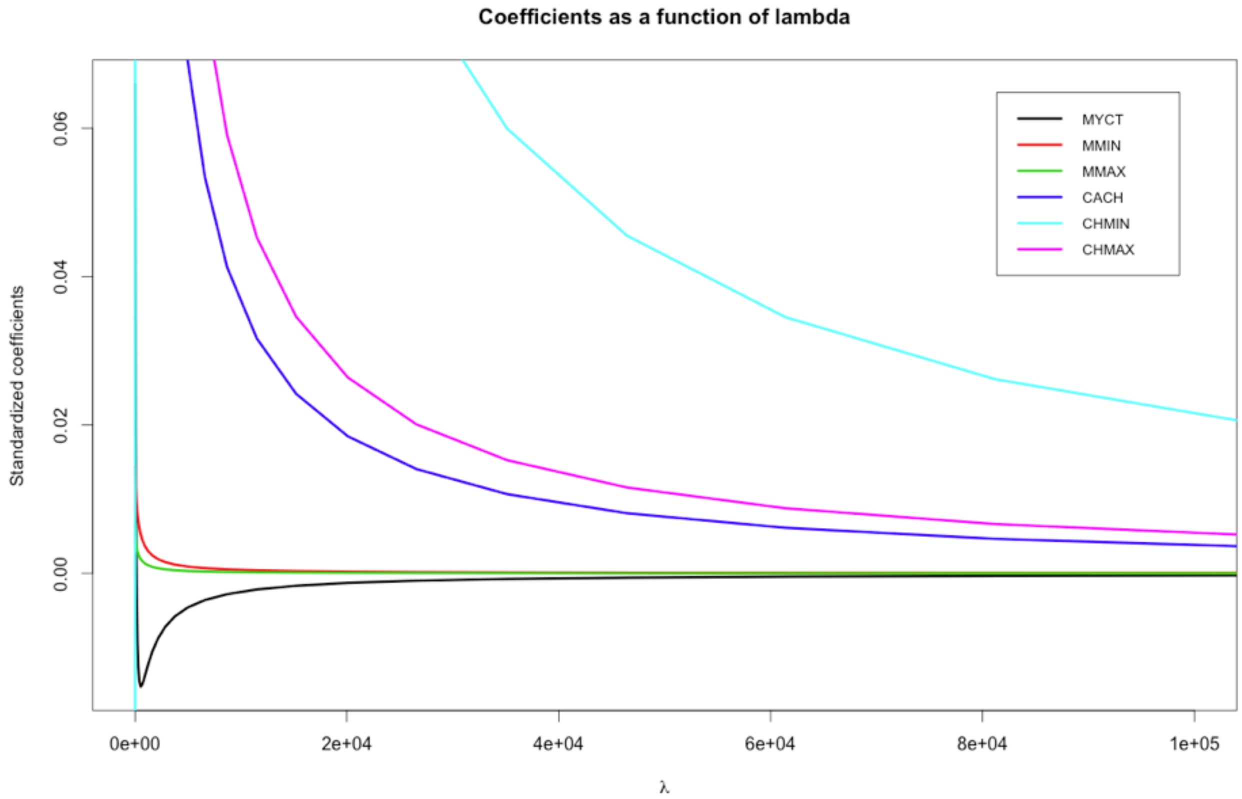
How to choose the value of lambda using cross validation?

```
set.seed(1)
cv.Out <- cv.glmnet(x[train,], y[train], alpha=0)     # cv.glmnet uses k-fold
plot(cv.out)
bestlam <- cv.out$lambda.min
bestlam

## 16.87241

# Apply the best lambda to predict over test set.
ridge.pred <- predict(ridge.mod, s=bestlam, newx=x[test,])
mean((ridge.pred - y.test)^2)

## 2837.874
```

This result is better than the MSE=4633.87 that was obtained from selecting a random value of lambda = 4.

Individual coefficients behavior

- For lambda close to zero (left), ridge regression is equivalent to least squares.
- For large values of lambda, coefficients tend to zero, which is equivalent to null model.
- Some coefficients tend to increase as lambda increases (MYCT).



Bias vs. Variance in Ridge Reg.

As lambda increases, the flexibility of the model __________,

leading to __________ variance, but ___________ bias.

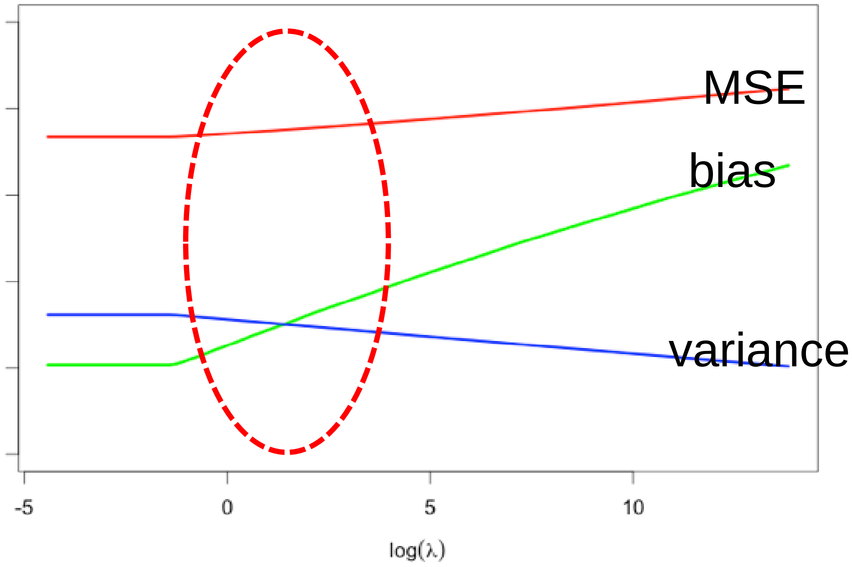
$$\sum_{i=1}^{n}\left(y_i - \beta_0 - \sum_{j=1}^{p}\beta_j x_{ij}\right)^2 + \lambda\sum_{j=1}^{p}{\beta_j}^2$$

[increase, decrease]

Bias vs. Variance in Ridge Reg.

As lambda increases, the flexibility of the model **decreases**,

leading to **decreased** variance, but **increased** bias.



Lasso

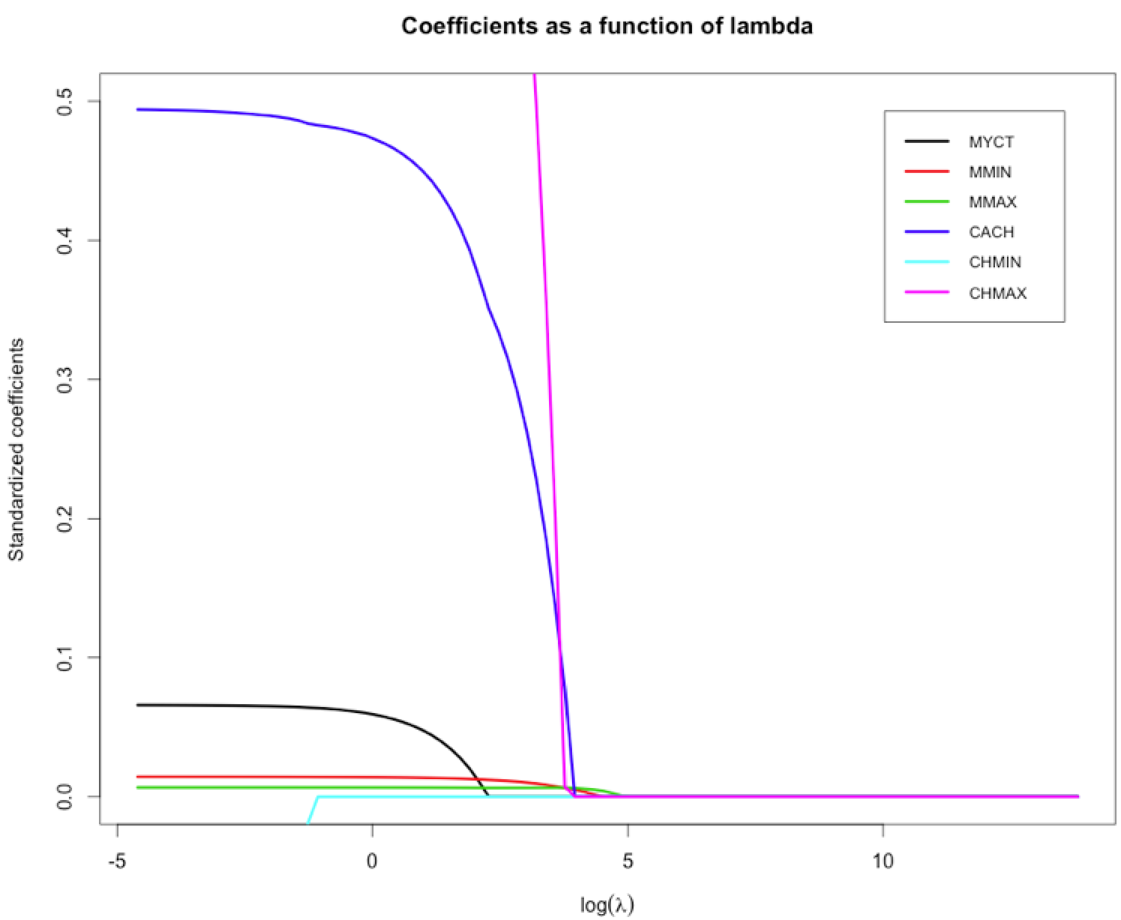
- Ridge regression will **include all p predictors** in the final model. **Does NOT perform subset selection.**
- **Lasso provides that functionality.**

*Forces some of the coefficient estimates to be **zero** when the tuning parameter λ is sufficiently large*

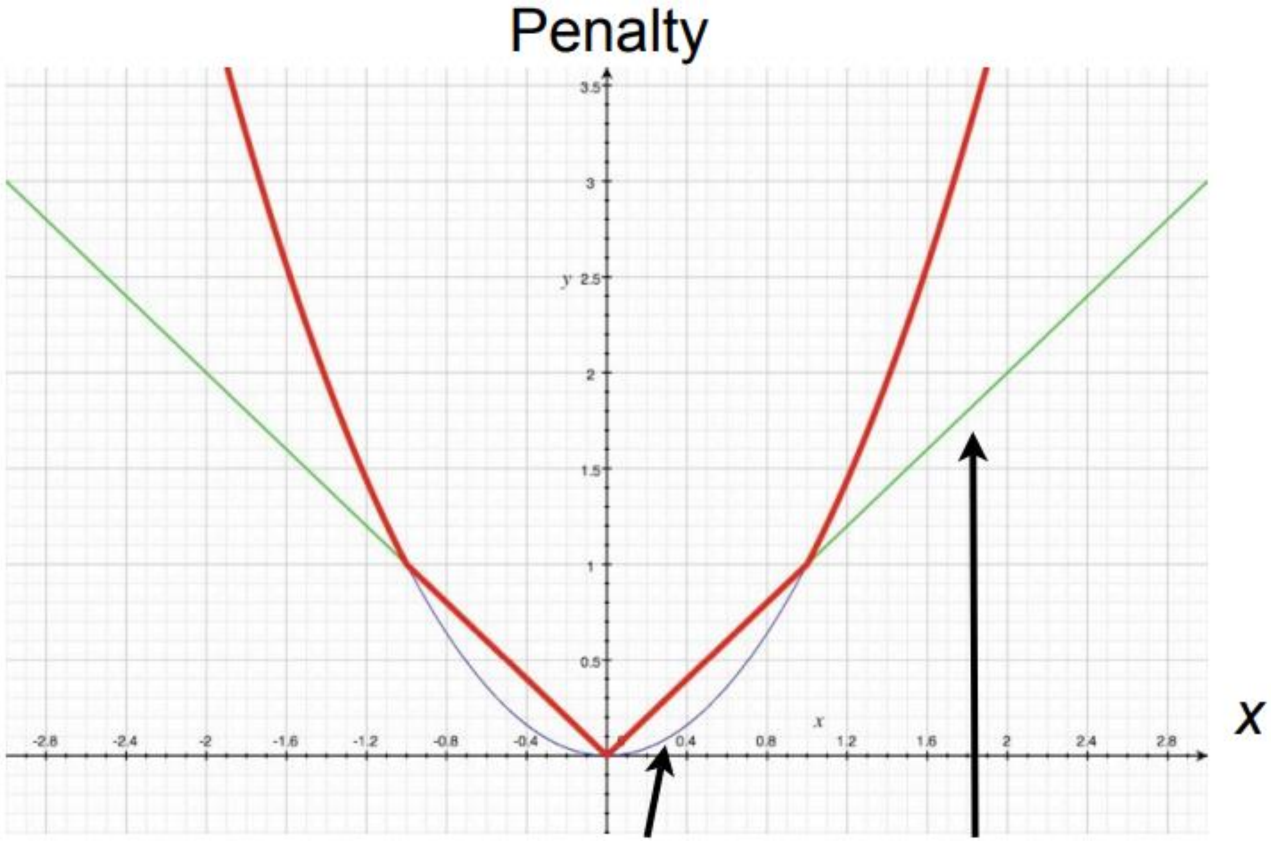
$$\sum_{i=1}^{n}\left(y_i - \beta_0 - \sum_{j=1}^{p}\beta_j x_{ij}\right)^2 + \lambda\sum_{j=1}^{p}|\beta_j|$$

With ℓ_1 norm, instead of ℓ_2 norm. $\|\beta\|_1 = \sum|\beta_j|$

Effect on coefficients



Lasso vs. Ridge



Lasso penalizes more when x is small (use this for **sparsity**)

Ridge penalizes more when x is big (use this for **robustness**)

Lasso vs. Ridge

- **Lasso models are generally much easier to interpret**
- **Lasso models perform better when the response is a function of only a relatively small number of predictors**.
- Cross-validation can be used to determine which approach is better, per case basis.

Summary

- Feature engineering is **IMPORTANT** especially when accuracy is not the only driver.

- **Strategy:** filter lightly, select models by overshooting and regularize
  - Better to include irrelevant features than to miss important ones
  - Use regularization or feature selection to prevent overfitting

- **Evaluate feature engineering on DEV set**. Then, when the feature set is frozen, evaluate on TEST to get a final evaluation.

Further readings

- Check the 'Linear Model Selection and Regularization' chapter in An Introduction to Statistical Learning
- Guyon, I., & Elisseeff, A. (2003). An introduction to variable and feature selection. *Journal of machine learning research*, *3*(Mar), 1157-1182.
- Nate Silver. The Signal and the Noise: Why So Many Predictions Fail--but Some Don't.