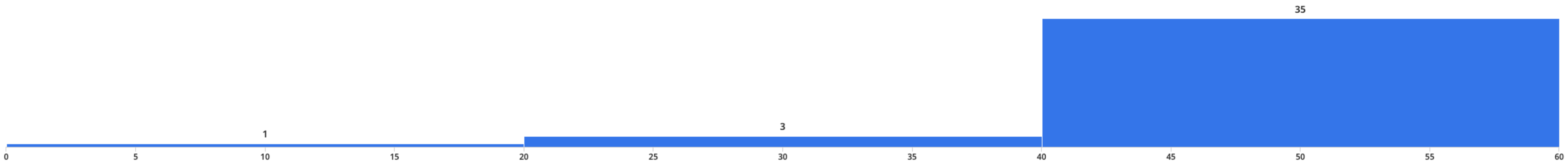
# HOMEWORK #1 GRADES

## Review Grades for Homework1-Code

● Grades Published




| Minimum | Median | Maximum | Mean | Std Dev |
| --- | --- | --- | --- | --- |
| **14.0** | **60.0** | **60.0** | **54.65** | **11.71** |

## Review Grades for Homework1-Written

● Regrade Requests Closed ● Grades Published


| Minimum | Median | Maximum | Mean | Std Dev |
| --- | --- | --- | --- | --- |
| **30.0** | **37.0** | **40.0** | **36.08** | **2.83** |

# HOMEWORK #3

- Due 10/13 @ 11:59 PM ET on Gradescope
- 4 questions
    - Feature selection
    - Closed form Linear Regression
    - SGD-based Linear Regression
    - Comparison of closed form and SGD

# REMINDER: PROJECT

- Proposal due 10/23: 1-2 pages of problem, dataset, what you plan to do
- Spotlight slides due 10/30
- Spotlight: 11/1 in class
- Presentation: 11/29 and 12/4
- Report and deliverable due 12/13

# LINEAR REGRESSION (PART IV)

CS 334: Machine Learning

Slides adapted from Joyce Ho, Lee Cooper, Joydeep Ghosh, Carlos Carvalho, and Ryan Tibshirani

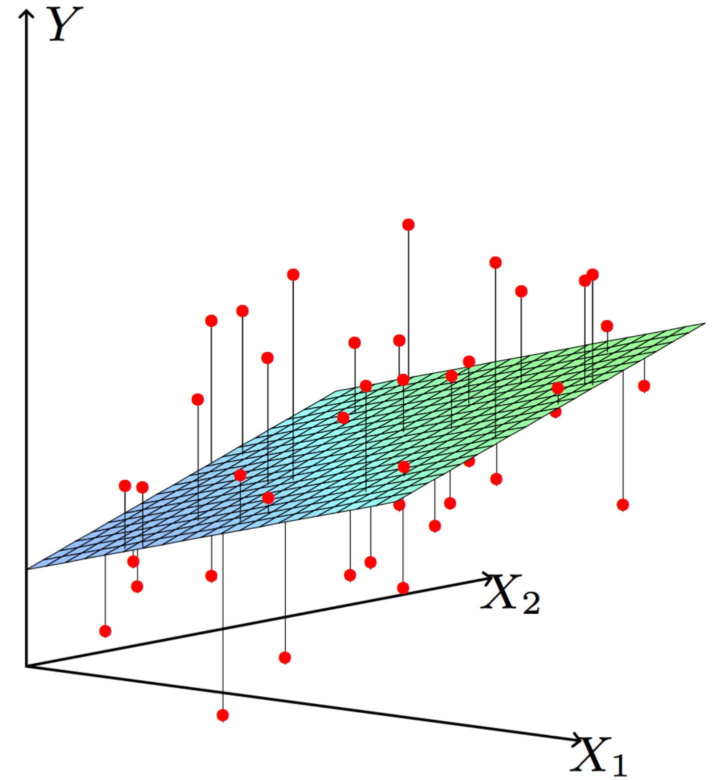
# LINEAR REGRESSION

- Closed form (direct solution)
- Iterative algorithms: Gradient descent (GD) and Stochastic gradient descent (SGD)
- Regularization: Ridge and Lasso
- Assessment

# REVIEW: REGRESSION: LEAST SQUARES

- Find parameters that minimizes some cost function
- Residual: difference between actual Y and predicted Y
- Least squares: minimize residual sum of squares (RSS or SSR)

$$
\begin{aligned}
RSS(\boldsymbol{\beta}) &= \frac{1}{2}\sum_{i=1}^{N}(y_i - f(\mathbf{x}_i)^2) \\
&= \frac{1}{2}(\mathbf{y} - \mathbf{X}\boldsymbol{\beta})^\top(\mathbf{y} - \mathbf{X}\boldsymbol{\beta})
\end{aligned}
$$




$$f(\mathbf{x}) = \beta_0 + \sum_{i=1}^{p} x_i \beta_i$$

**How to find the solution?**

Figure 3.1 (Hastie et al.)

# REVIEW: MODEL REGULARIZATION

$$\min_{\beta} L(\mathbf{X}\boldsymbol{\beta}, \mathbf{y}) + \lambda \text{penalty}(\boldsymbol{\beta})$$

- Basic Idea: Add penalty term on model parameters to "shrink" the coefficients towards zero
- Called regularization
- Less prone to overfitting (prediction accuracy)
- Achieve a simpler model, get the "right" model complexity (interpretability)

# REVIEW: POPULAR PENALTIES

| Name | Penalty function |
| --- | --- |
| Ridge | $\|\|\beta\|\|_2$ |
| Lasso | $\|\|\beta\|\|_1$ |
| $L_0$ regularization | $\|\|\beta\|\|_0$ |
| Elastic net | $\alpha\|\|\beta\|\|_1 + (1-\alpha)\|\|\beta\|\|_2$ |

# REVIEW: EFFECT OF SELECTION ON COEFFICIENTS

| Term | LS | Best Subset | Ridge | Lasso |
|---|---|---|---|---|
| **Intercept** | 2.465 | 2.477 | 2.452 | 2.468 |
| **lcavol** | 0.680 | 0.740 | 0.420 | 0.533 |
| **lweight** | 0.263 | 0.316 | 0.238 | 0.169 |
| **age** | −0.141 | | −0.046 | |
| **lbph** | 0.210 | | 0.162 | 0.002 |
| **svi** | 0.305 | | 0.227 | 0.094 |
| **lcp** | −0.288 | | 0.000 | |
| **gleason** | −0.021 | | 0.040 | |
| **pgg45** | 0.267 | | 0.133 | |
| Test Error | 0.521 | 0.492 | 0.492 | 0.479 |
| Std Error | 0.179 | 0.143 | 0.165 | 0.164 |

How did this happen?

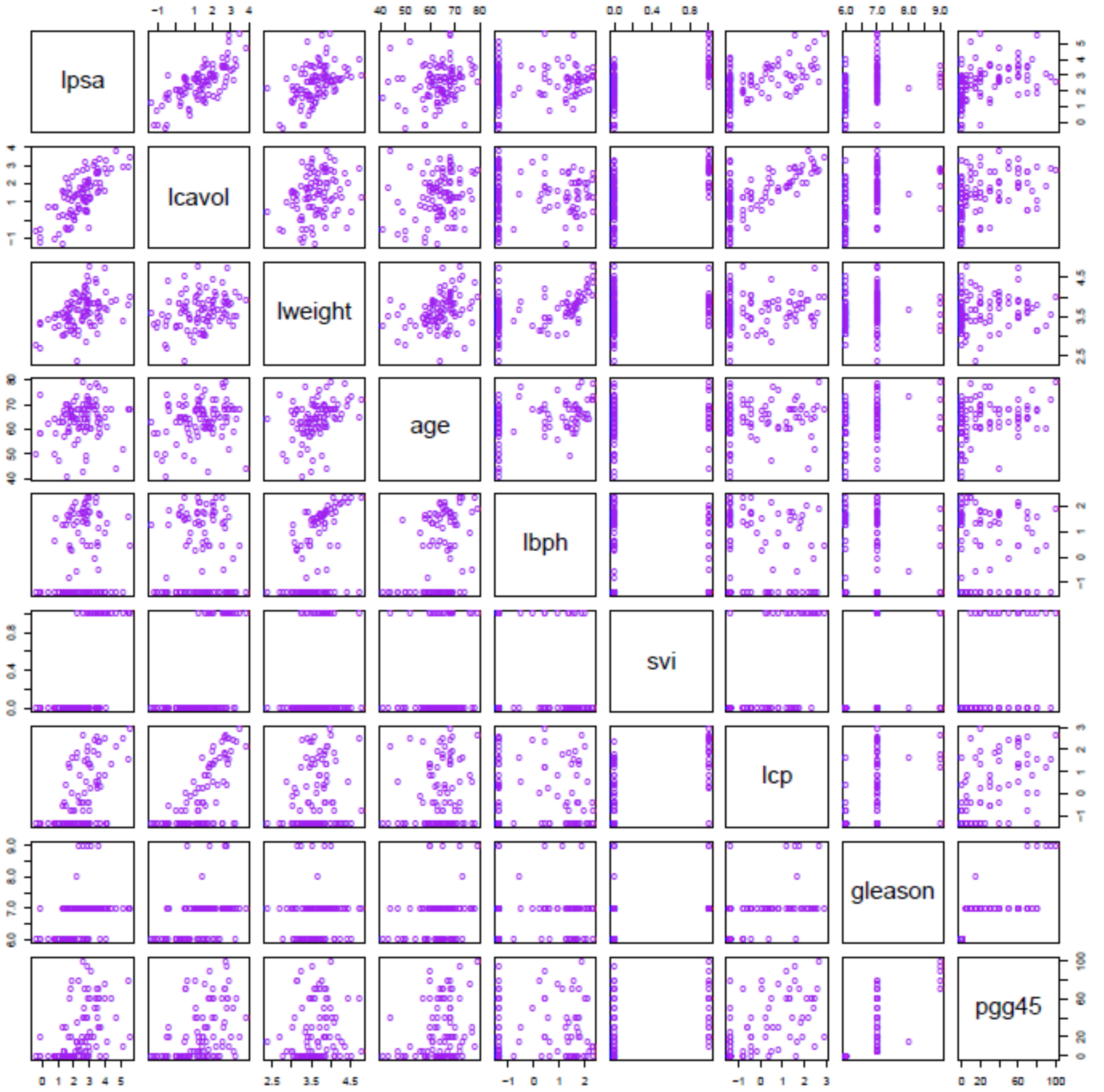
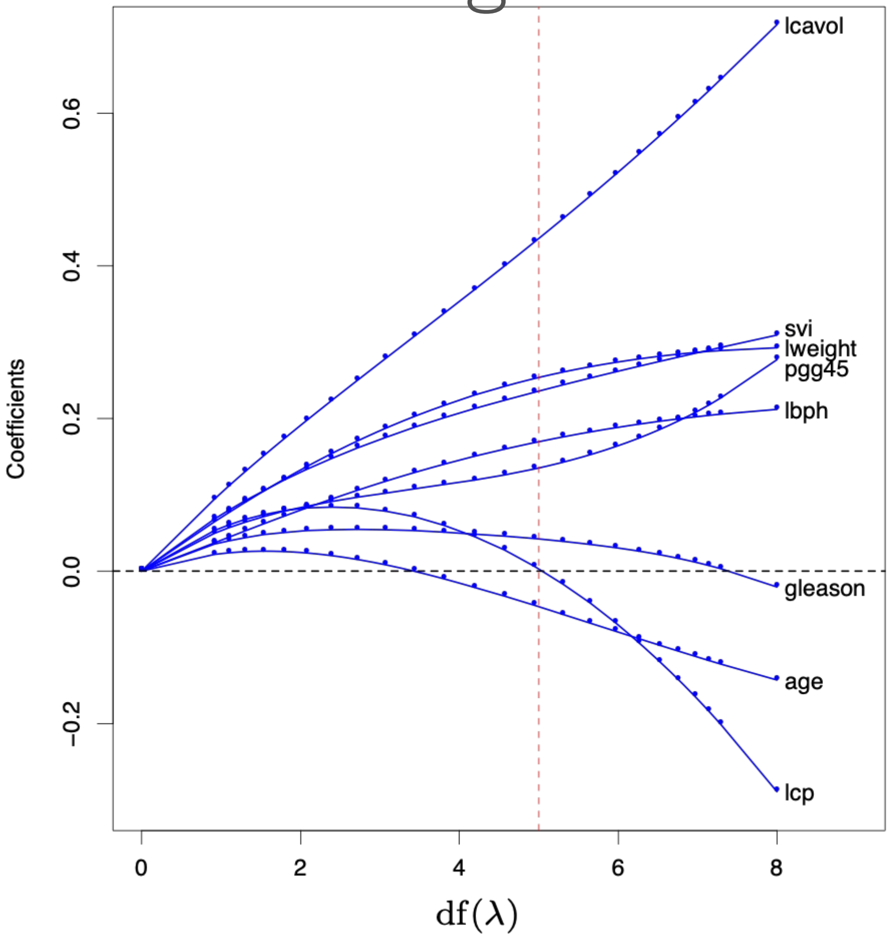



**FIGURE 1.1.** *Scatterplot matrix of the prostate cancer data. The first row shows the response against each of the predictors in turn. Two of the predictors,* **svi** *and* **gleason***, are categorical.*

# REVIEW: RIDGE VS. LASSO: COEFFICIENT PATHS

$$\min_{\beta} L(\mathbf{X}\boldsymbol{\beta}, \mathbf{y}) + \lambda ||\boldsymbol{\beta}||_2$$

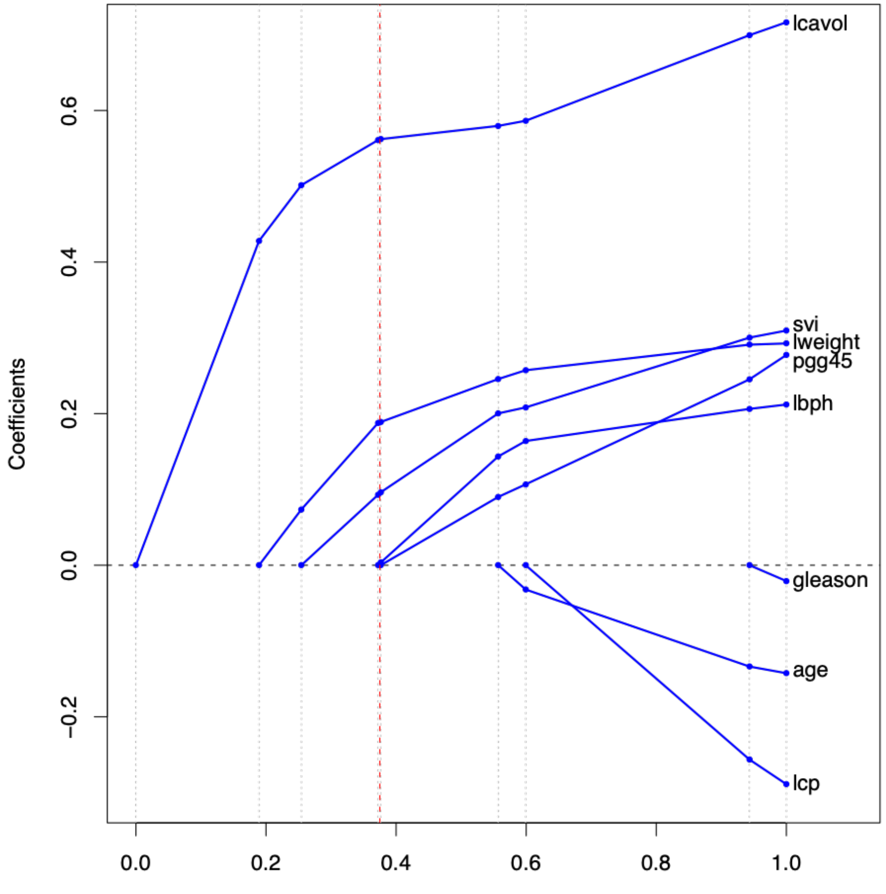
Ridge




effective degrees of freedom (inverse to lamda)

$$\min_{\beta} L(\mathbf{X}\boldsymbol{\beta}, \mathbf{y}) + \lambda ||\boldsymbol{\beta}||_1$$

LASSO



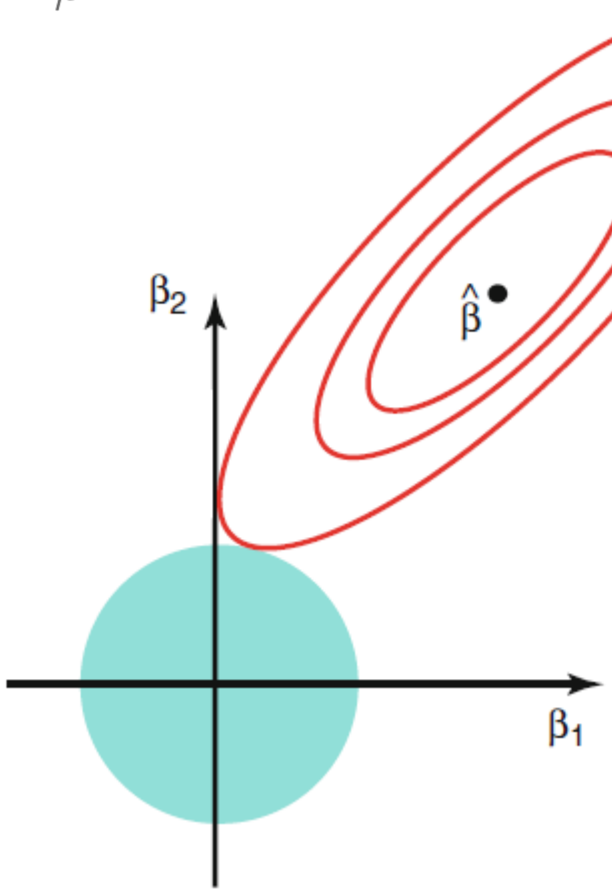

Why did this happen?

Figure 3.8 and 3.10 (Hastie et al.)

# RIDGE VS. LASSO: OPTIMIZATION

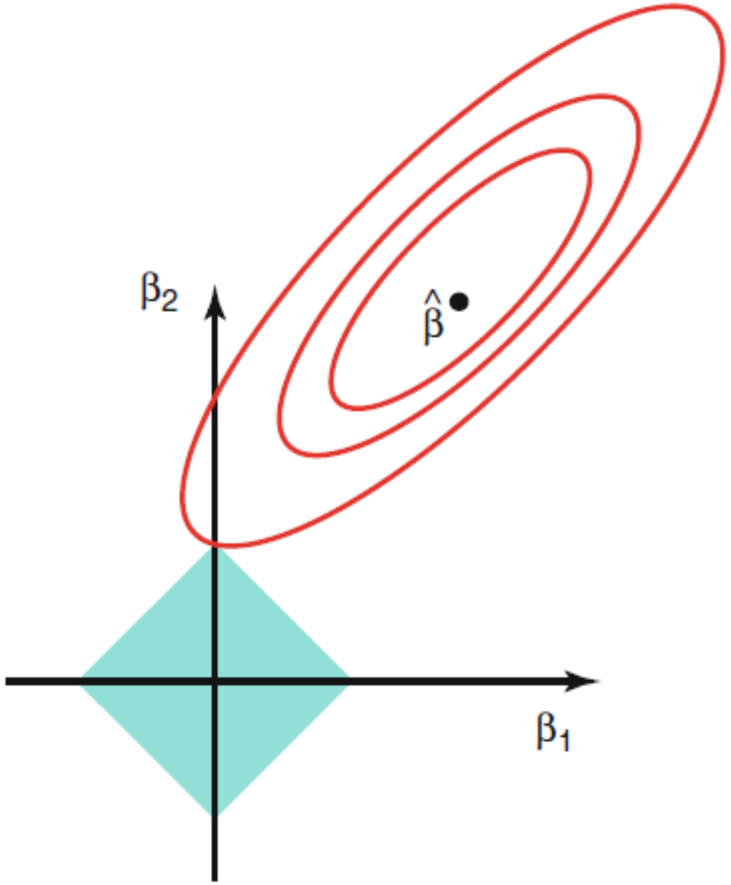
Ridge

$$\min_{\beta} L(\mathbf{X}\boldsymbol{\beta}, \mathbf{y}) + \lambda||\boldsymbol{\beta}||_2$$




LASSO

$$\min_{\beta} L(\mathbf{X}\boldsymbol{\beta}, \mathbf{y}) + \lambda||\boldsymbol{\beta}||_1$$




ure 6.7 (James et al.)

# ELASTIC NET REGULARIZATION

- Compromise between ridge and lasso

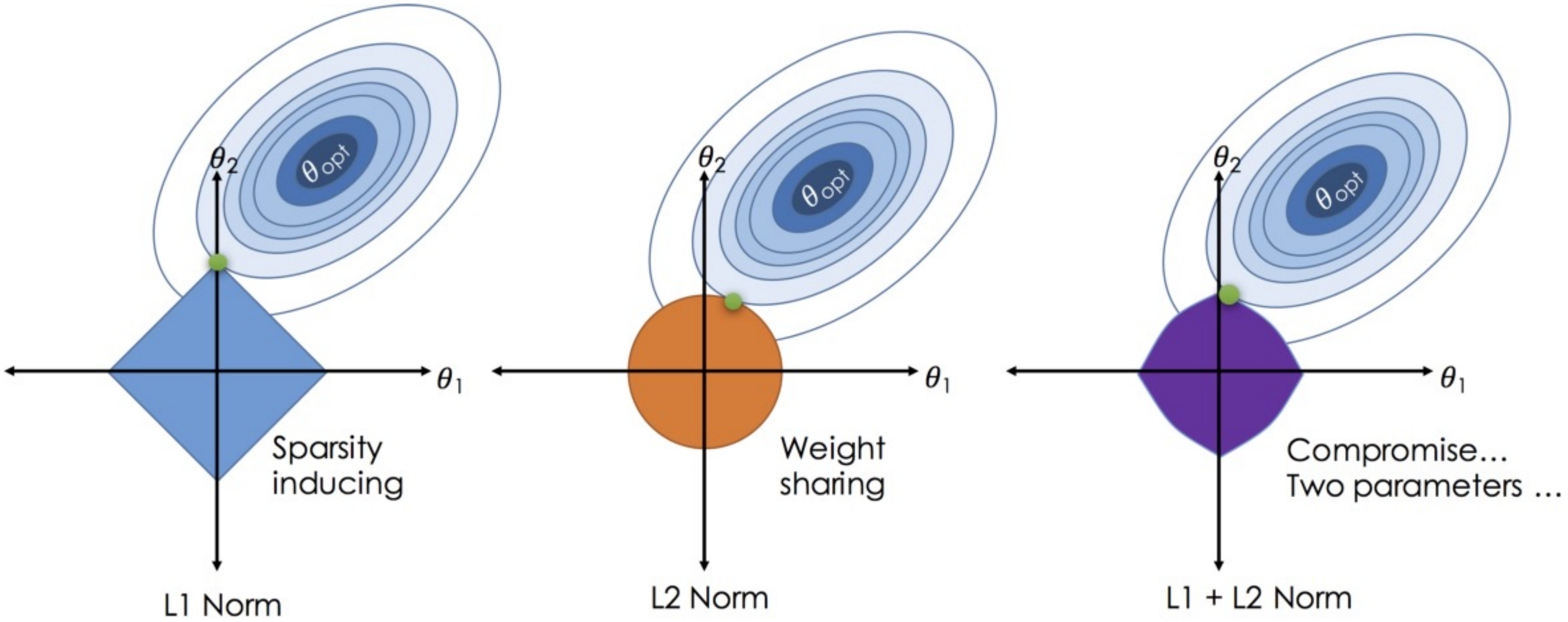
$$\min_{\beta} L(\mathbf{X}\boldsymbol{\beta}, \mathbf{y}) + \lambda(\alpha||\boldsymbol{\beta}||_2 + (1-\alpha)||\boldsymbol{\beta}||_1)$$

- Selects variables like lasso

- Shrinks coefficients of correlated predictions like ridge

- Computational advantages over general $L_q$ penalties

# RIDGE VS LASSO VS ELASTIC NET



# RIDGE & LASSO REGULARIZATION: NOTES

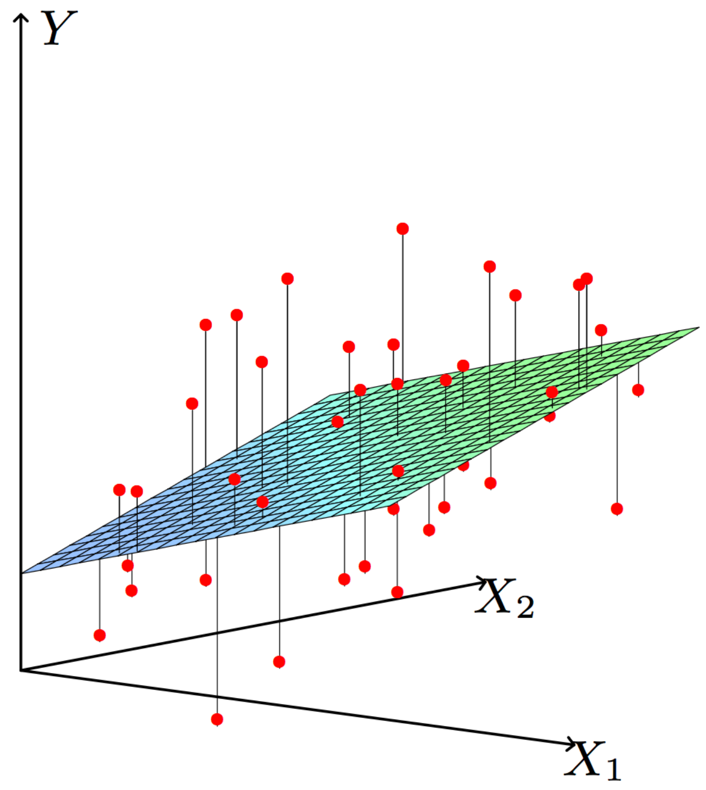
- If intercept term is included in regression, this coefficient is left unpenalized
- Can center the data, only perform regression on other coefficients
- Penalty term can be unfair if predictors are on different scales
  - Normalization

# LINEAR REGRESSION

- Closed form (direct solution)
- Iterative algorithms: Gradient descent (GD) and Stochastic gradient descent (SGD)
- Regularization: Ridge and Lasso
- Assessment

# LINEAR REGRESSION: ASSESSING THE ACCURACY

- Residual error
- $R^2$ statistic



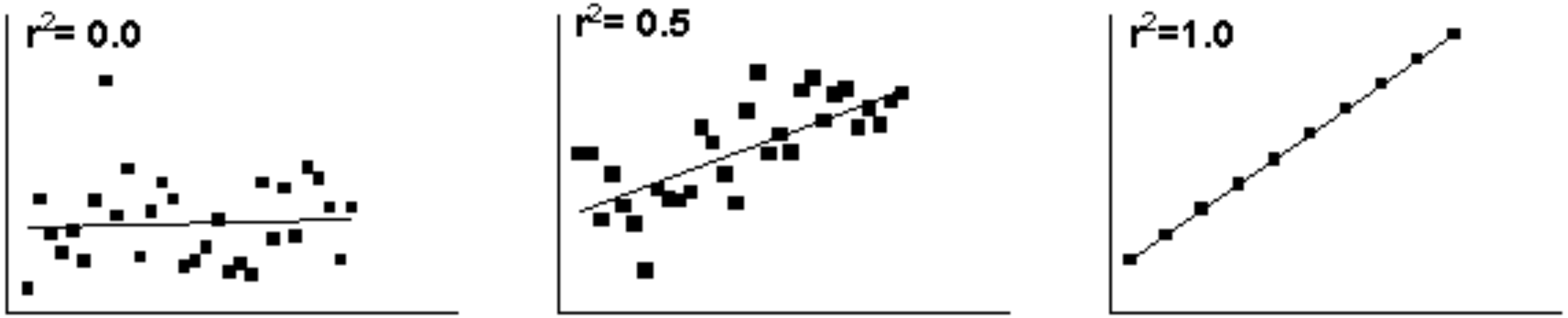
# MEASURE OF FIT: $R^2$

- "Goodness" of fit measure

$$R^2 = 1 - \frac{\sum_i (y_i - \hat{y}_i)^2}{\sum_i (y_i - \bar{y})^2}$$

- Interpretation: The proportion of variability in y explained by the model

- Always lies between 0 and 1

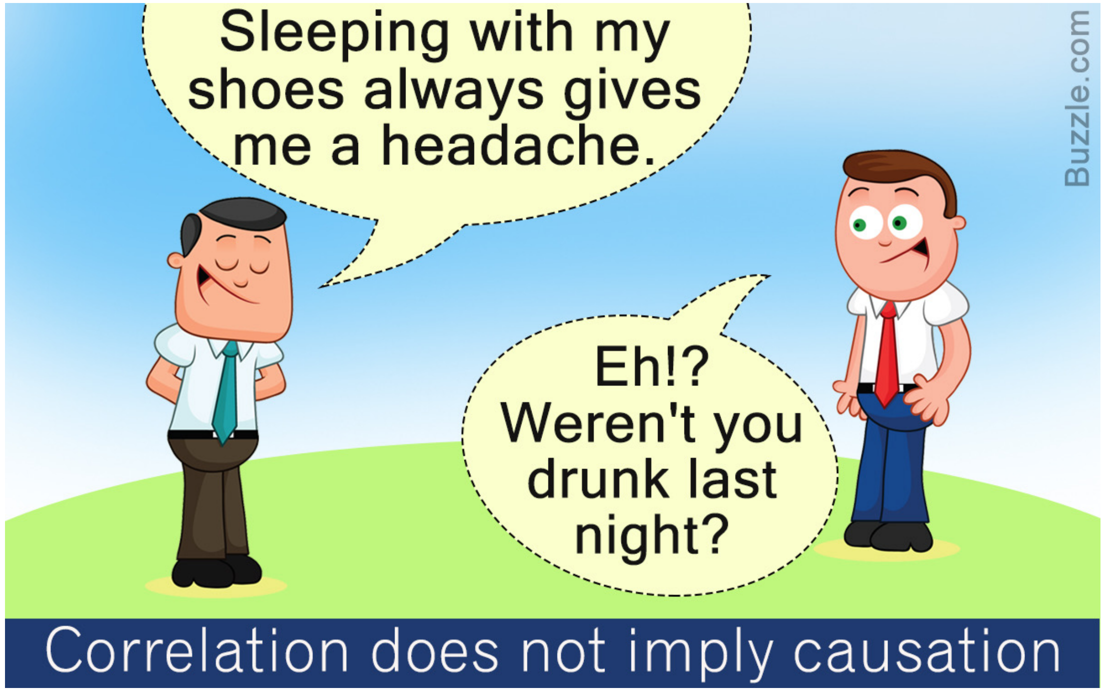


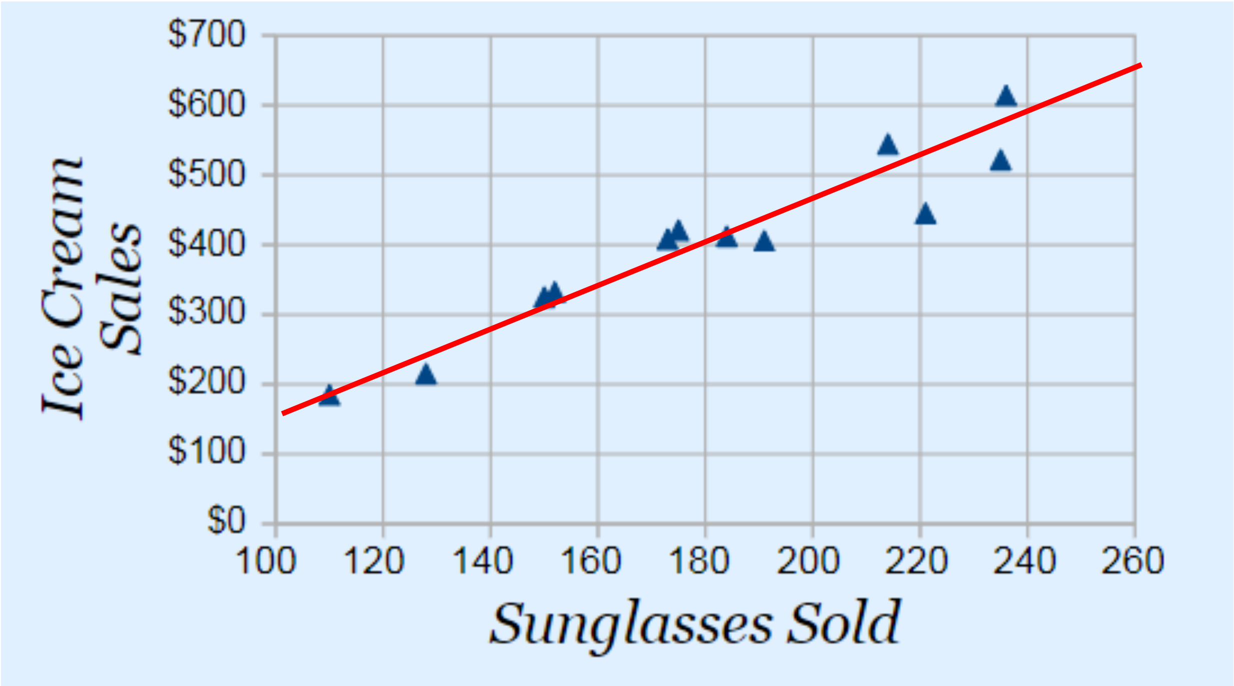

https://socratic.org/questions/why-doesn-t-an-r-squared-value-indicate-anything-about-causation

# UNDERSTANDING MLR

- Coefficients have a nice interpretation and show level of correlation
- Correlation != causality
    - Any correlation (association) could be caused by other variables in the background

# CORRELATION DOES NOT IMPLY CAUSALITY



# LINEAR REGRESSION: SKLEARN

- sklearn.linear_model.LinearRegression
- sklearn.linear_model.Ridge
- sklearn.linear_model.Lasso
- sklearn.linear_model.ElasticNet
- sklearn.linear_model.SGDRegressor

# FEATURE SELECTION

CS 334: Machine Learning

# PROSTATE CANCER DATASET

How would you choose a subset of relevant features to predict lpsa (besides using LASSO)?

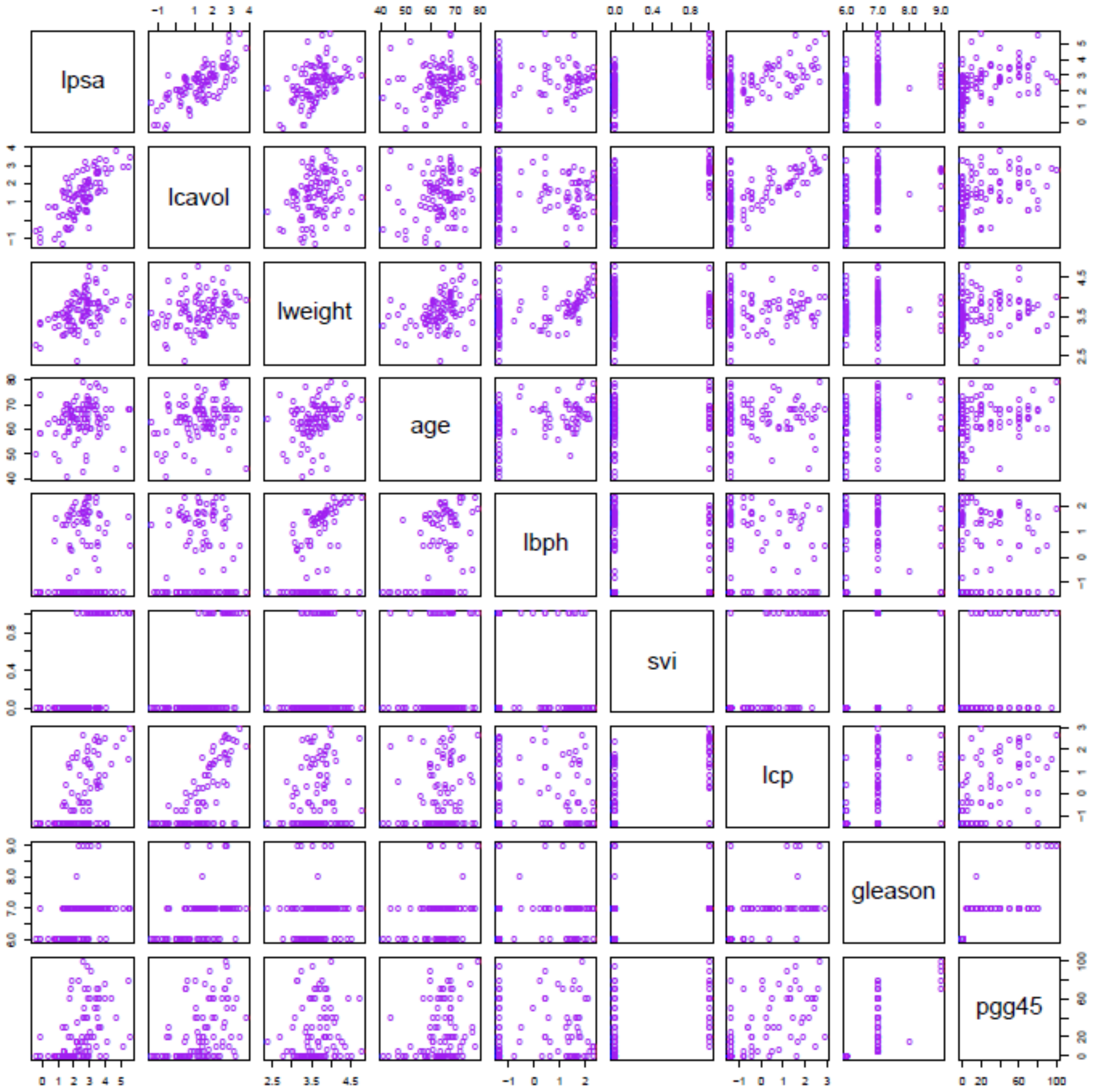
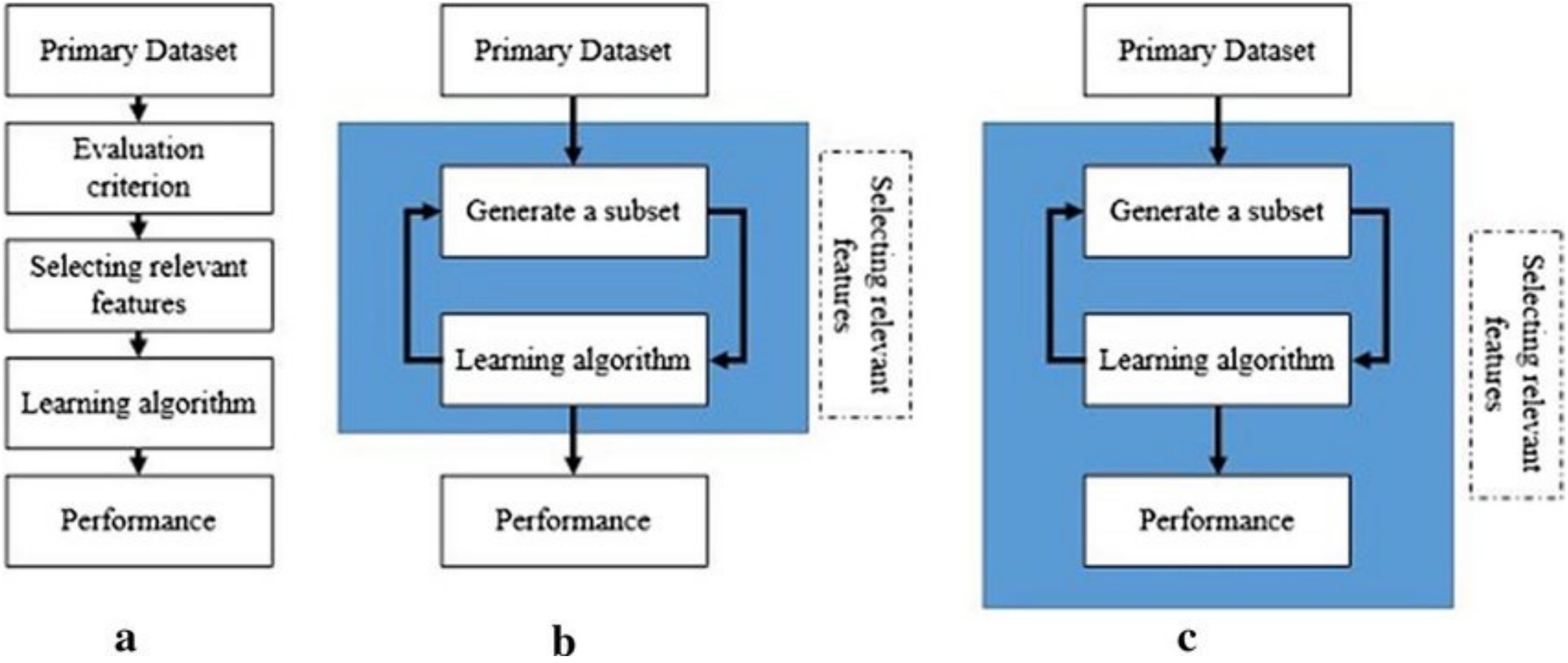



**FIGURE 1.1.** *Scatterplot matrix of the prostate cancer data. The first row shows the response against each of the predictors in turn. Two of the predictors,* `svi` *and* `gleason`*, are categorical.*

# FEATURE SELECTION METHODS

- Filter methods – agnostic to the models (preprocessing)
- Wrapper methods – evaluate on the model (model selection)
- Embedded methods – part of the learning algorithm (model training), e.g. LASSO

# FEATURE SELECTION METHODS



# FILTER FEATURE SELECTION

- Based on heuristics but much faster than wrapper methods
- Use statistical measure to assign a score to each feature
- Methods are often univariate and consider the feature independently with regard to the dependent variable.

# FILTER FEATURE MEASURES

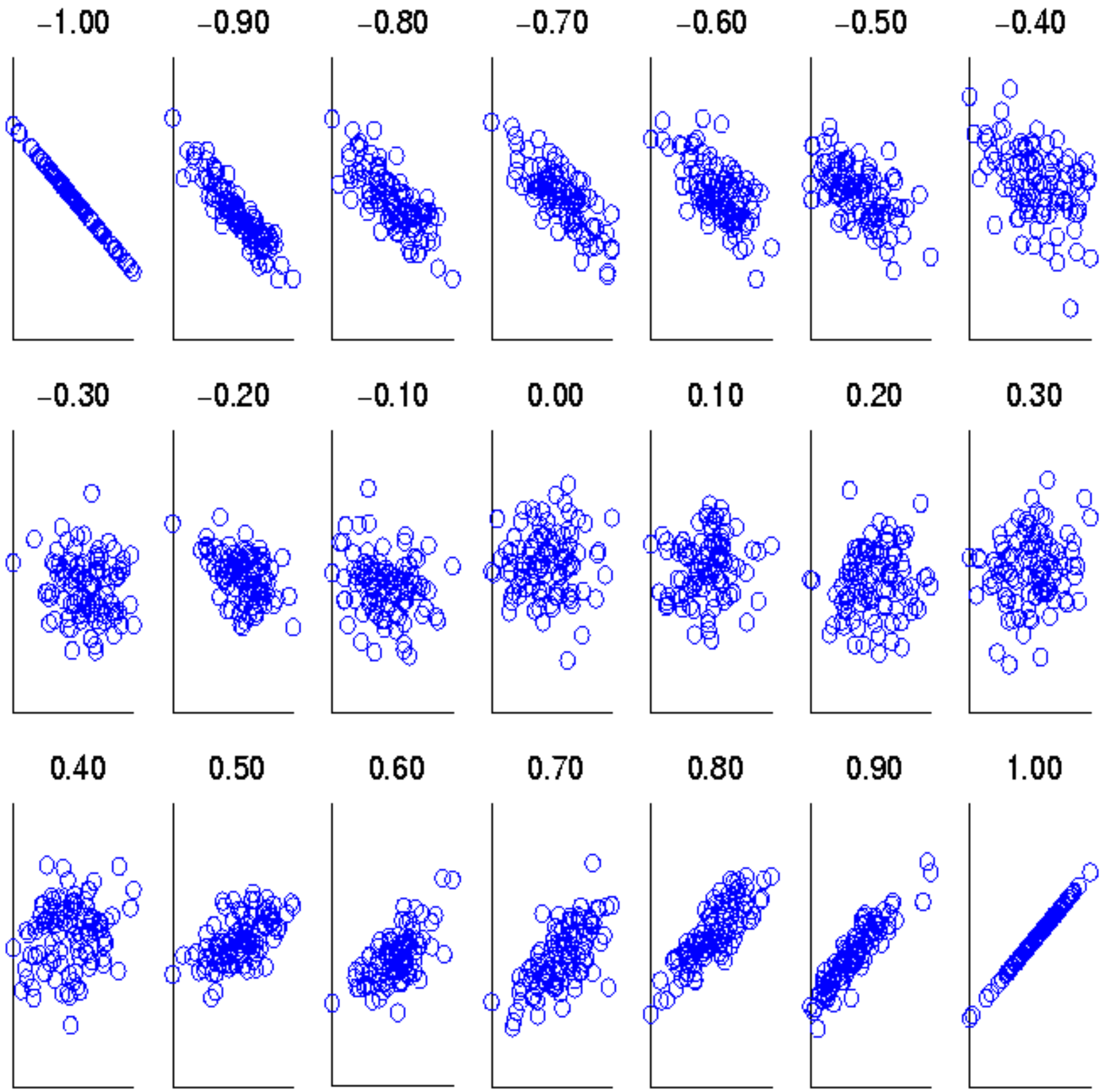
- Pearson correlation (population and sample):

$$\rho_{X,Y} = \frac{\mathrm{cov}(X,Y)}{\sigma_X \sigma_Y}$$

$$r_{xy} = \frac{\sum_{i=1}^{n}(x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_{i=1}^{n}(x_i - \bar{x})^2}\sqrt{\sum_{i=1}^{n}(y_i - \bar{y})^2}}$$

- Between -1 and 1
- Measures linear correlation between two variables, scale and location invariant
- Can be used to rank features in order of their correlation with the labels
- Or to evaluate pair-wise redundancy between features

# PEARSON CORRELATION




Scatter plots showing the Pearson correlation from –1 to 1.

# FILTER FEATURE MEASURES

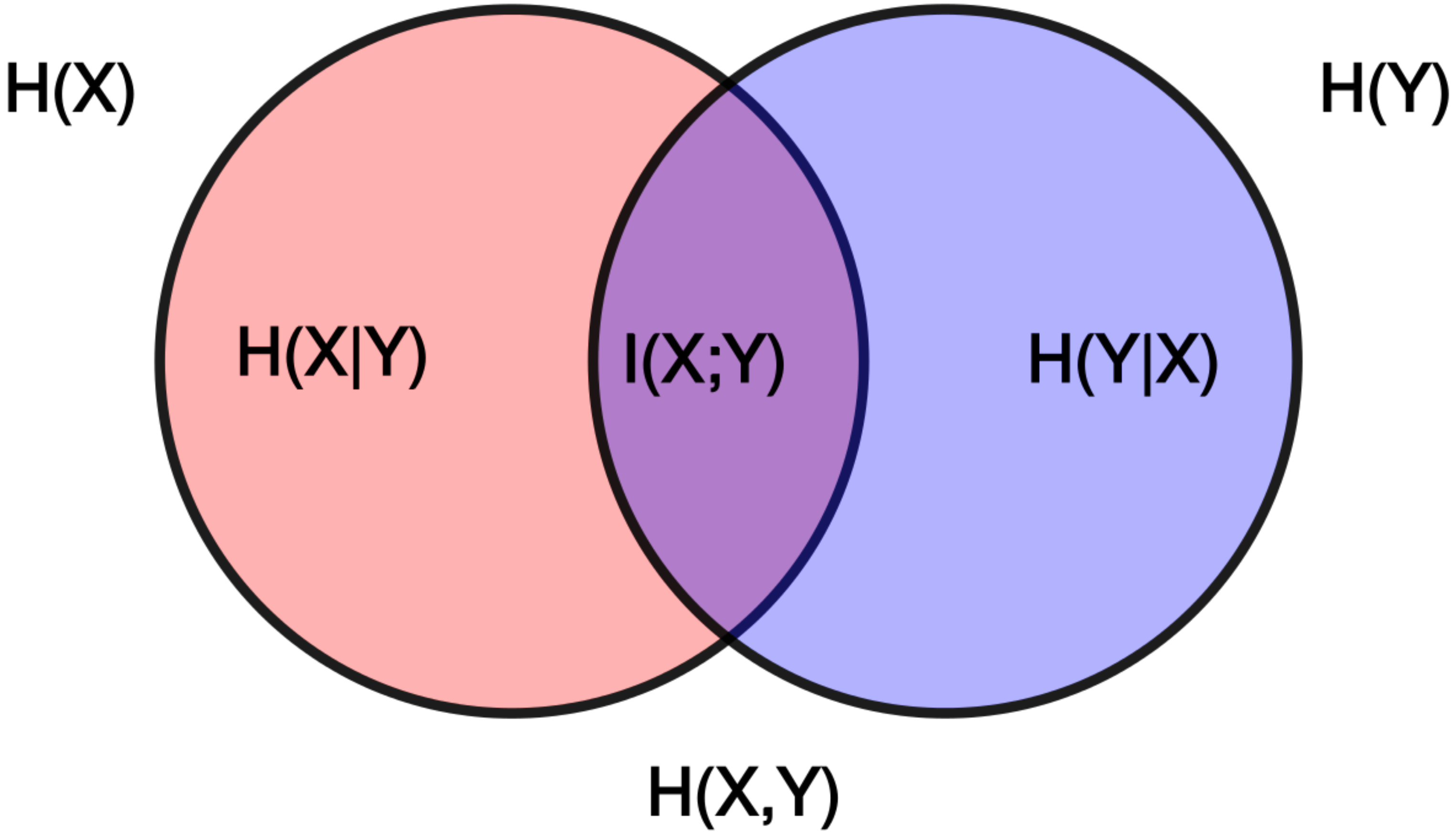
- Mutual information criterion (information gain)

$$I(X;Y) = H(X) - H(X|Y) = H(Y) - H(Y|X) = \sum_{x,y} p(x,y) \log \frac{p(x,y)}{p(x)p(y)}.$$

- Measures "the amount of information" about one variable through observing the other variable

- High mutual information means high relevance

# MUTUAL INFORMATION



# FEATURE SELECTION METHODS

- Filter methods – agnostic to the models (preprocessing)
- Wrapper methods – evaluate on the model (model selection)
- Embedded methods – part of the learning algorithm (model training)

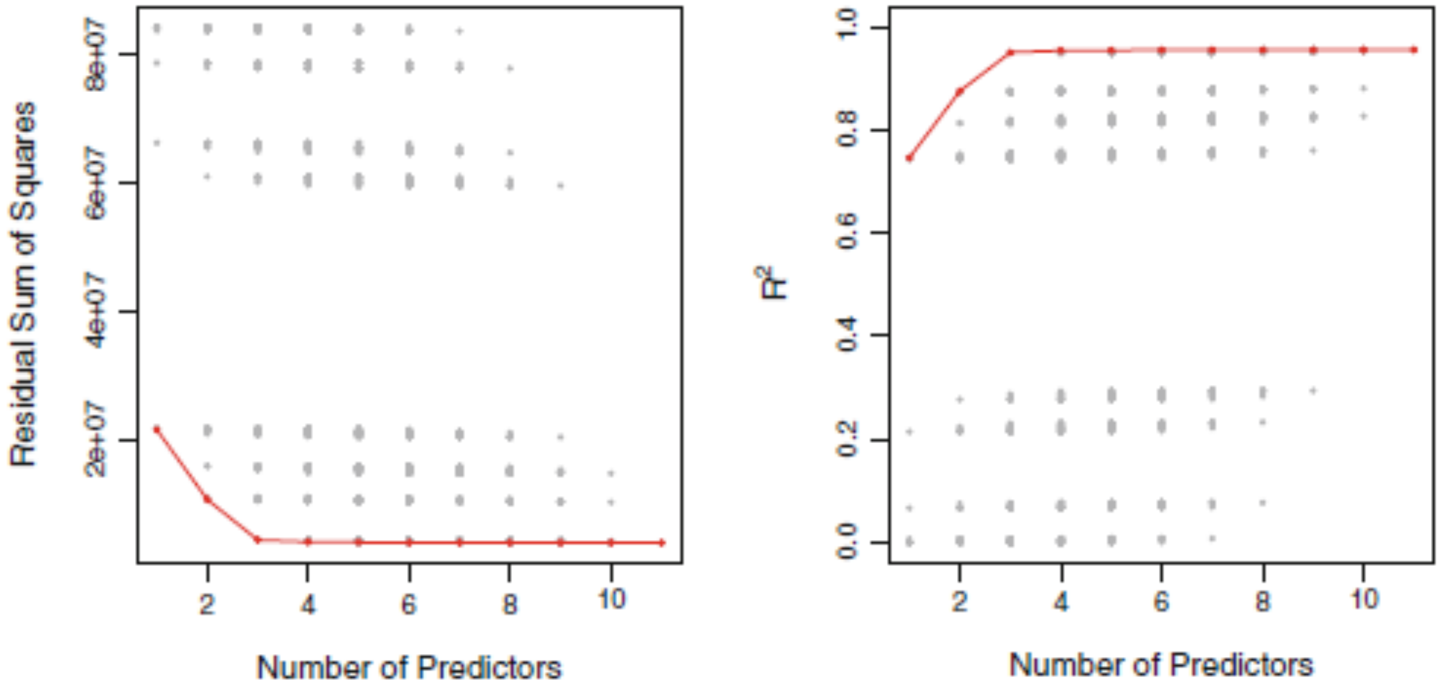
# SELECTING THE BEST FEATURES

- Brute force method: try all combinations and evaluate performance, pick the best combination
  - How to evaluate the performance?
  - How many combinations are there?

# FEATURE SELECTION FOR LINEAR REGRESSION

- Use RSS or $R^2$ on training data to select the best model given the same size

- Use cross-validation to select the best size




How many features should we choose?

Figure 6.1 James et al book

# SELECTING THE BEST FEATURES

- Brute force method: try all combinations and evaluate performance, pick the best combination
  - How to evaluate the performance?
  - How many combinations are there?

# SELECTING THE BEST FEATURES

- Brute force method: try all combinations and evaluate performance, pick the best combination
  - How to evaluate the performance?
  - How many combinations are there?

Computationally infeasible for large number of features

# WRAPPER METHOD

- Some form of searching (forward or backward)
  - Greedily add / remove features
  - Evaluate performance using cross-validation

# STEPWISE SELECTION

- Forward: Start with 0 features and sequentially add feature that best improves fit
  - Can be used whenever
- Backward: Start with full model, remove feature that is least detrimental to fit
  - Can only be used when $N > p$

# FEATURE SELECTION COMPARISON

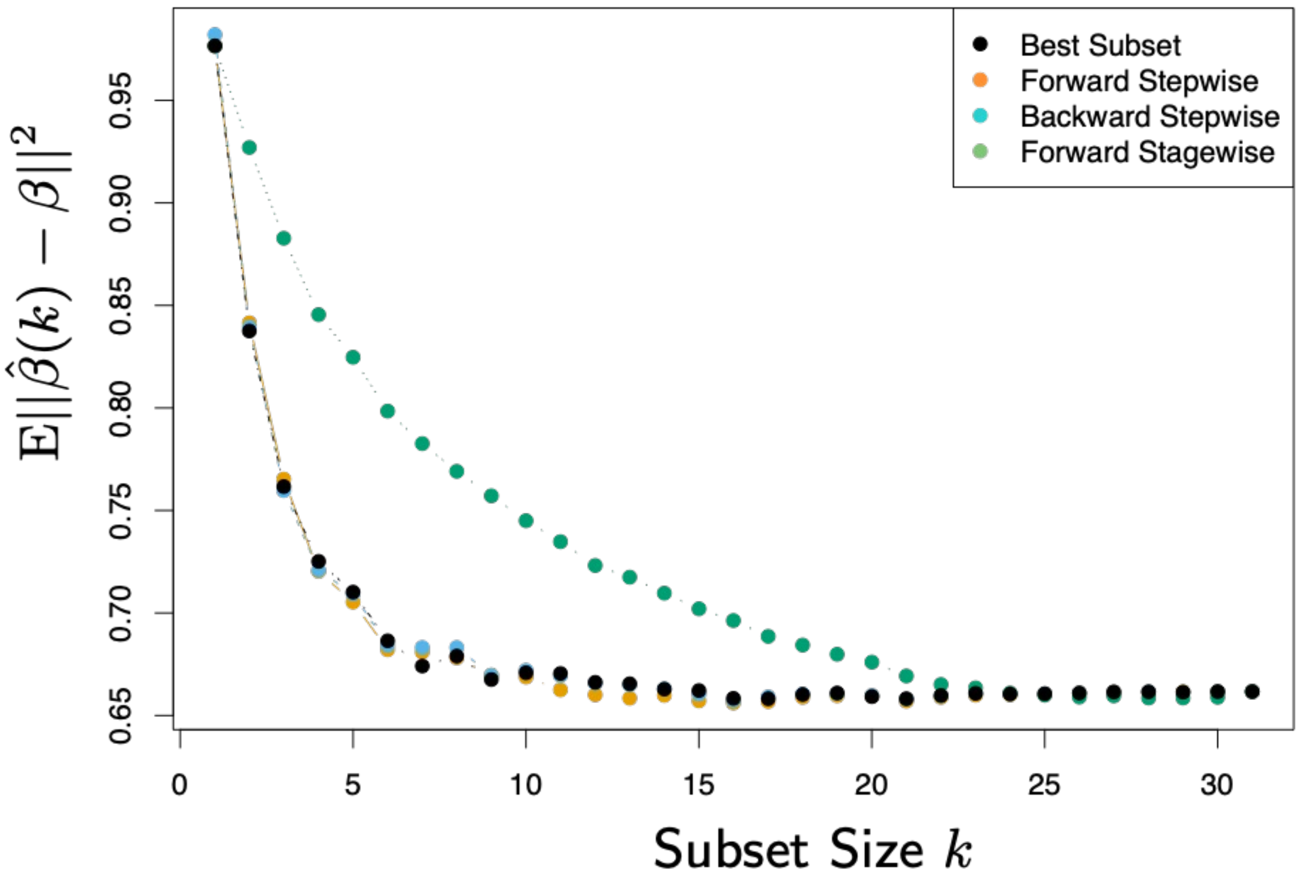



Figure 3.6 (Hastie et al.)

# FEATURE SELECTION: RECAP

| Filter methods | Wrapper methods | Embedded methods |
|---|---|---|
| Generic set of methods which do not incorporate a **specific machine learning algorithm.** | Evaluates on a **specific machine learning algorithm** to find optimal features. | Embeds (fix) features during **model building process.** Feature selection is done by observing each iteration of model training phase. |
| Much **faster** compared to Wrapper methods in terms of time complexity | **High computation time** for a dataset with many features | Sits **between Filter methods and Wrapper methods** in terms of time complexity |
| Less prone to **over-fitting** | High chances of **over-fitting** because it involves training of machine learning models with different combination of features | Generally used to reduce **over-fitting** by **penalizing** the coefficients of a model being too large. |
| Examples – **Correlation, Chi-Square test, ANOVA, Information gain** etc. | Examples - **Forward Selection, Backward elimination, Stepwise selection** etc. | Examples - **LASSO, Elastic Net, Ridge Regression** etc. |



https://towardsdatascience.com/feature-selection-using-wrapper-methods-in-python-f0d352b346f