INFO 251: Applied Machine Learning

Practical Considerations

Key concepts: Missing and Imbalanced Data

- Stratified randomization
- Upsampling and downsampling
- SMOTE and AdaSyn
- Reweighting
- Algorithm-level adjustments for imbalance
- Problems with data missingness
- Selective labels
- Model-free imputation (e.g., zero-coding, mean imputation)
- Model-based imputation (e.g., hot deck)

Course Outline

- Causal Inference and Research Design
  - Experimental methods
  - Non-experiment methods
- Machine Learning
  - Design of Machine Learning Experiments
  - Linear Models and Gradient Descent
  - Non-linear models
  - Neural models
  - Fairness and Bias
  - **Practicalities**
  - Unsupervised Learning
- Special topics

Outline

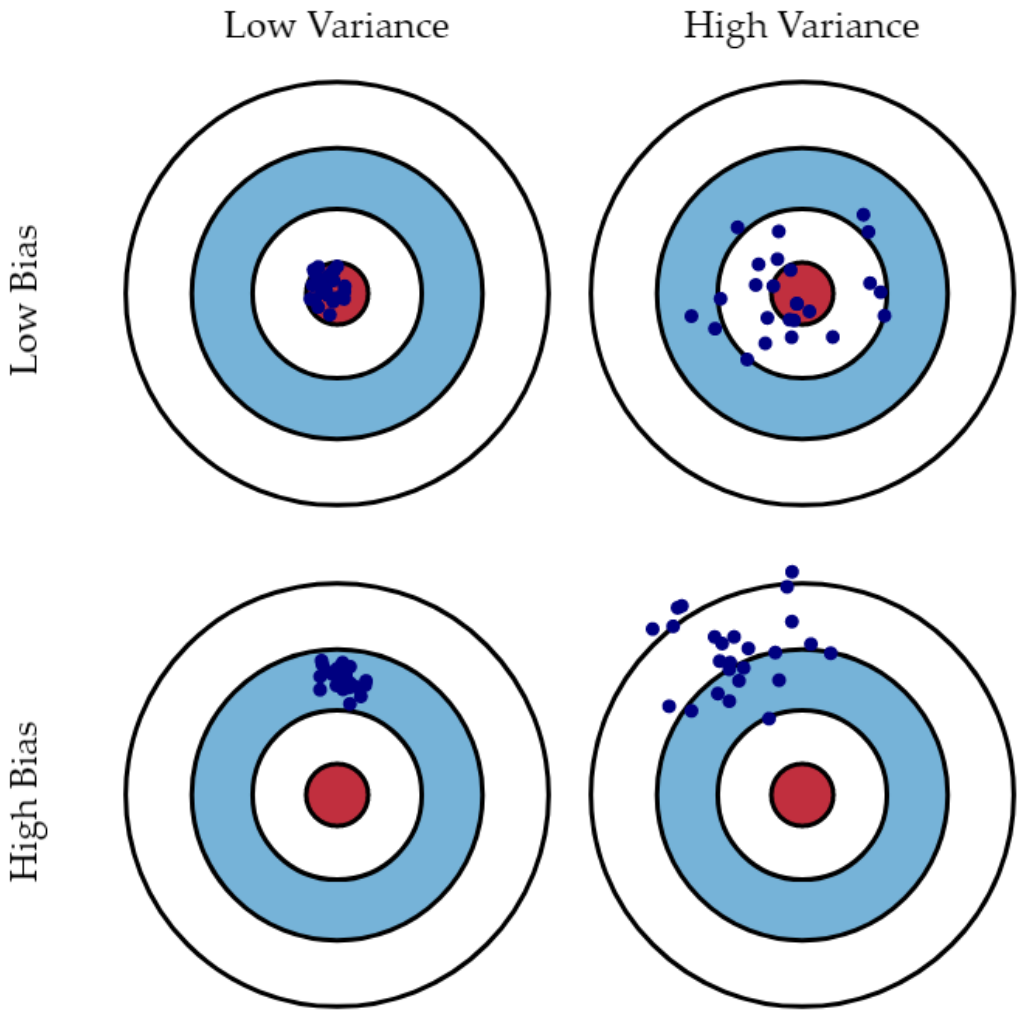
- **Bias-variance tradeoff**
- Features, features, features
- Multi-class classification
- Interpreting models

Bias and Variance

- **Intuition:**
  - **Bias**: Difference between expected model prediction and true value
    - If you retrained the same model on similar data, how bad is the average prediction for a given data point?
  - **Variance**: Difference between model's predictions for the same (or very similar) input observations
    - If you retrained the same model on similar data, how much variance in prediction for a given data point?
    - Sometimes described as how model and predictions change when trained with different training data

Bias and Variance

- Graphical intuition (part 1)



Bias and Variance

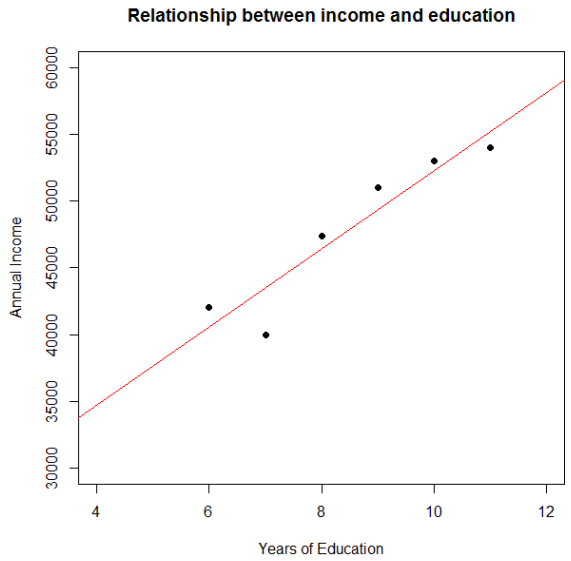
- Formalization
  - James et al. chapter 2
  - Daume chapter 5

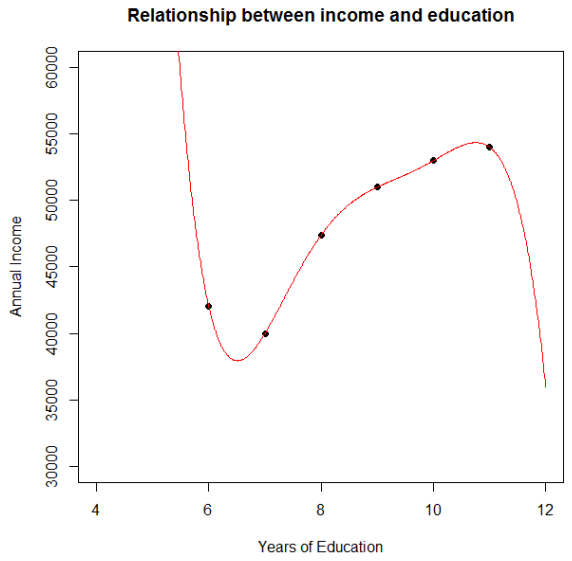
$$\text{error}(f) = \underbrace{\left[\text{error}(f) - \min_{f^* \in \mathcal{F}} \text{error}(f^*)\right]}_{\text{estimation error}} + \underbrace{\left[\min_{f^* \in \mathcal{F}} \text{error}(f)\right]}_{\text{approximation error}}$$

- **Estimation error (variance)**: How far is the actual learned model f from the optimal model f^*?
- **Approximation error (bias)**: How good could the model family F be (given infinite training data)?

Bias and Variance

- Graphical intuition (part 2)





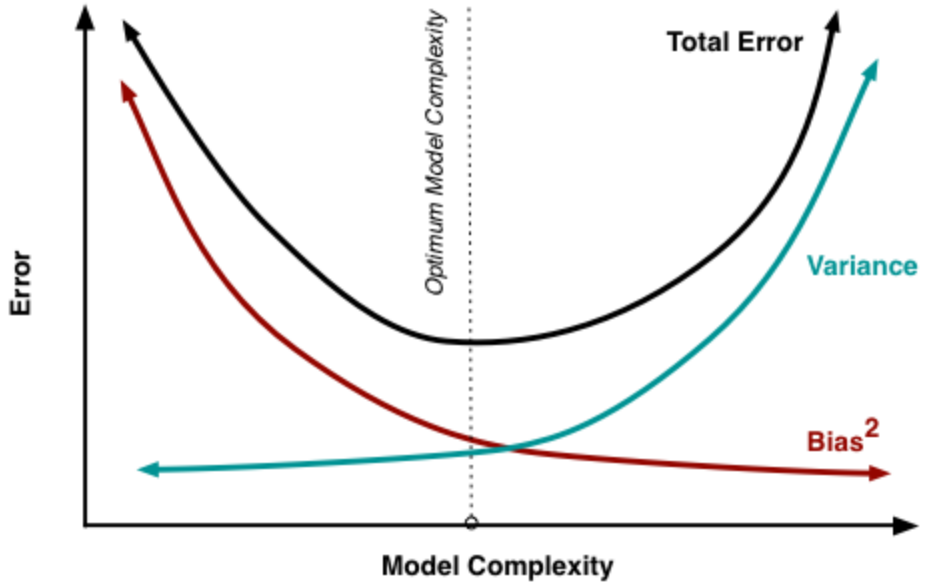
High bias
(underfit)

High variance
(overfit)

Bias and Variance

- Finding the "sweet spot"

$$\frac{\delta Bias}{\delta Complexity} = \frac{\delta Variance}{\delta Complexity}$$



Bias and Variance

- Intuition check: I have a classification model that always predicts true. Is this a high bias or high variance model?

- A rough comparison:

High Bias	Low Bias
Low Variance	High Variance
Simple models	Complex models
Underfitting	Overfitting
Lower estimation error	Lower approximation error

Outline

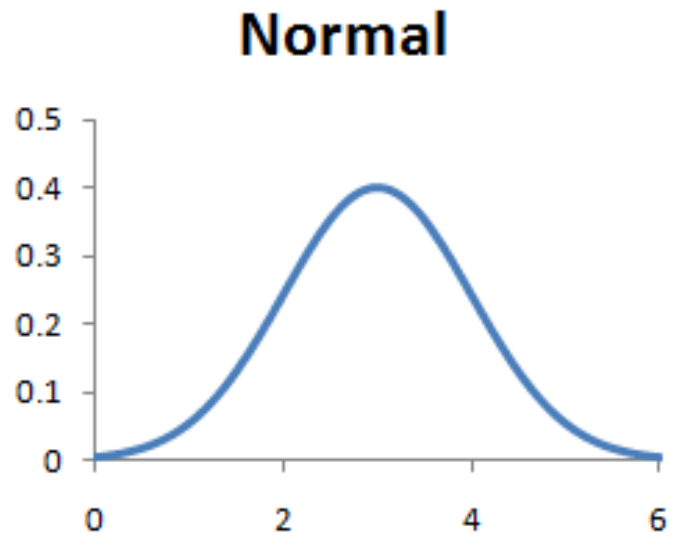
- Bias-variance tradeoff
- **Features, features, features**
- Multi-class classification
- Interpreting models

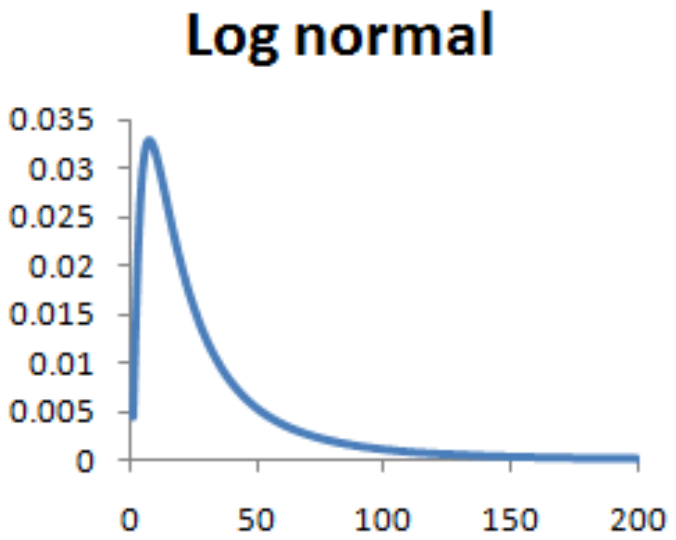
Thinking about features

- What to do with the raw data?
  - Feature engineering
  - Feature expansion
  - Feature selection
  - Redundant features
  - Irrelevant features
  - Feature and instance scaling

Feature expansion

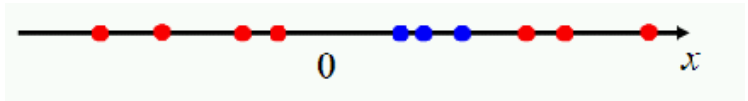
- Expanding your features can add expressivity to your model
  - Log(feature) – very common!
    - Especially when your original feature is lognormally distributed
    - And especially if you are using linear models (e.g., regression)

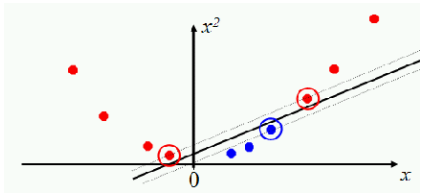


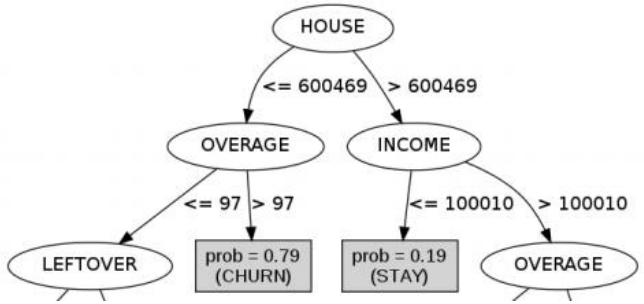


Feature expansion

- Expanding your features can add expressivity to your model
  - Log(feature) – very common!
    - Especially when your original feature is lognormally distributed
    - And especially if you are using linear models (e.g., regression)
  - Feature polynomials (x_1, x_1^2 , x_1^3)
  - Feature interactions (x_1, x_2, $x_1 * x_2$)
  - Feature interactions with decision trees
    - E.g., use paths through a decision tree to determine which features to include (e.g., in a downstream regression model)







Feature selection

- Disciplining your features can improve generalizability
  - Many forms of regularization will do this for you
- Other common brute force techniques (especially when there are simply too many features to simply pass to your model)
  - Univariate selection (e.g., choose features based on univariate correlation with response variable)
  - Forward/Backward selection
  - Remove low variance features
  - See `sklearn.feature_selection`

Feature selection

- Exhaustive backward selection
  - Train model with all *k* features
  - Train *k* models with *k-1* features, choose best model
  - Train k^2 models with *k-2* features, choose best model
  - Repeat until a stopping condition is reached
- Recursive Feature Elimination (iterative)
  - Train model with all features
  - Identify least important feature and remove it
    - E.g., using feature importance from tree-based model
    - E.g., based on smallest coefficient in linear model
  - Repeat until a stopping condition is reached

Redundant features

- Common pitfall for many learning algorithms: noisy, irrelevant, redundant features
  - With N observations and K features, chance of randomly finding perfectly correlated features is 0.5^{N-K}
  - Chance of finding highly correlated features much higher
- Consider k-Nearest Neighbors: all features are treated equally
  - Worth considering: how can the other algorithms we've studied deal with irrelevant features? Collinear features?

Redundant features

- Common solutions for irrelevant, redundant features
  - Remove features with low variance

$$s_k^2 = \frac{1}{N}\sum_{i=1}^{N}(x_{ik} - \bar{x}_k)^2$$

    - Remember, should be computed on your training set (i.e., store $\bar{x}_k$ and s_k^2 so that they can later be applied to the test set!)
  - In the case of binary features, this includes features that are heavily imbalanced
    - e.g., rare features, like the word "prestidigitation"
    - e.g., common features, like the word "the"

Feature and Instance Scaling

- We've talked previously about feature scaling
  - Feature standardization: subtract mean, divide by SD

$$x'_{ik} = \frac{x_{ik} - \bar{x}_k}{s_k}$$

- Also common: Example normalization
  - Rescale each observation so that the length of the feature vector is one

$$x'_n = \frac{x_n}{||x_n||}$$

  - E.g., document with same words repeated 2x looks same as document with same exact words occurring 1x
  - All observations lie on the unit hypersphere
  - Makes comparisons across datasets possible

Outline

- Bias-variance tradeoff
- Features, features, features
- **Multi-class classification**
- Interpreting models

Multi-class problems

- Most response variables come in one of four types
  1. Continuous variables
  2. Binary variables
  3. Multi-class categorical variables
  4. Rank-ordered variables

Multi-class problems

- One Versus All (OVA, aka "one versus rest")
  - Given *K* output classes
  - Train *K* binary classifiers $\{f_1, \ldots, f_K\}$
  - Each classifier trains on all training data
  - Classifier f_j trains on a new response variable:

$$Y_{ij} = \begin{cases} 1 & \text{if } Y_i = j \\ 0 & \text{otherwise} \end{cases}$$

  - Test instances passed to all classifiers, final prediction is whichever classifier predicts positive
    - Ties can be broken via coin toss
    - Ties can be broken based on classifier confidence

Multi-class problems

- One Versus All (OVA, aka "one versus rest")
  - Weakness: can be brittle
  - If one classifier makes a mistake, entire prediction can be thrown off
  - Handling of ties can be arbitrary
  - In many cases, no classifier predicts TRUE

Multi-class problems

- All Versus All (AVA, aka "all pairs")
  - "Tournament of classifiers"
  - One classifier for each pair of outputs - $\binom{K}{2}$ total
  - Call f_{ij} the classifier that seeks to differentiate between classes i ("California") and j ("Texas")
  - f_{ij} trains with i instances labeled TRUE and j FALSE
  - Test instances passed through all $\binom{K}{2}$ classifiers
    - Whenever f_{ij} predicts TRUE, *i* gets a point
    - Whenever f_{ij} predicts FALSE, *j* gets a point
    - Class with the most votes "wins" (can also build confidence)

Multi-class problems

- Softmax regression
  - Softmax: converts output from multiple logits into probabilities that sum to one
  - Aka (or closely related): conditional logit, multinomial logistic regression, polytomous logistic regression, softmax regression, maximum entropy classifier
  - Linear response function, like logistic regression
  - Assignment scores (for each observation to each category) can be converted to a *probability*
  - Interpretable coefficients (as in logistic regression)

Outline

- Bias-variance tradeoff
- Features, features, features
- Multi-class classification
- **Interpreting models**

Interpreting models

- The models we have discussed vary in terms of if and how they can be interpreted. Consider, for example:
  - Linear and logistic regression
  - Decision trees
  - k-Nearest Neighbors
  - Random forests
  - Naïve bayes
  - Neural networks

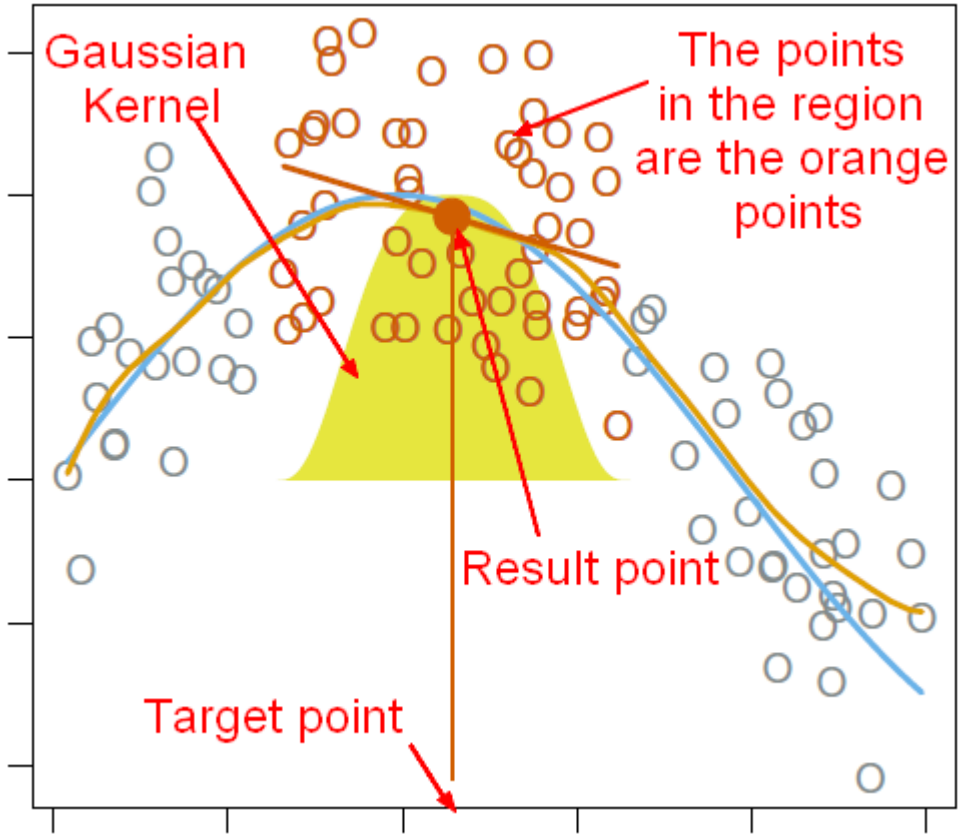
Interpreting models

- There are also several model-agnostic approaches to interpretation
  - Univariate correlation analysis: which features are most correlated with the response variable?
  - Permutation importance: average decrease in performance when a given feature is randomized (see lecture on Random Forests)
  - Non-parametric regressions (e.g., LOESS, LOWESS, Splines)
  - SHAP plots

Interpreting models: LOESS / LOWESS

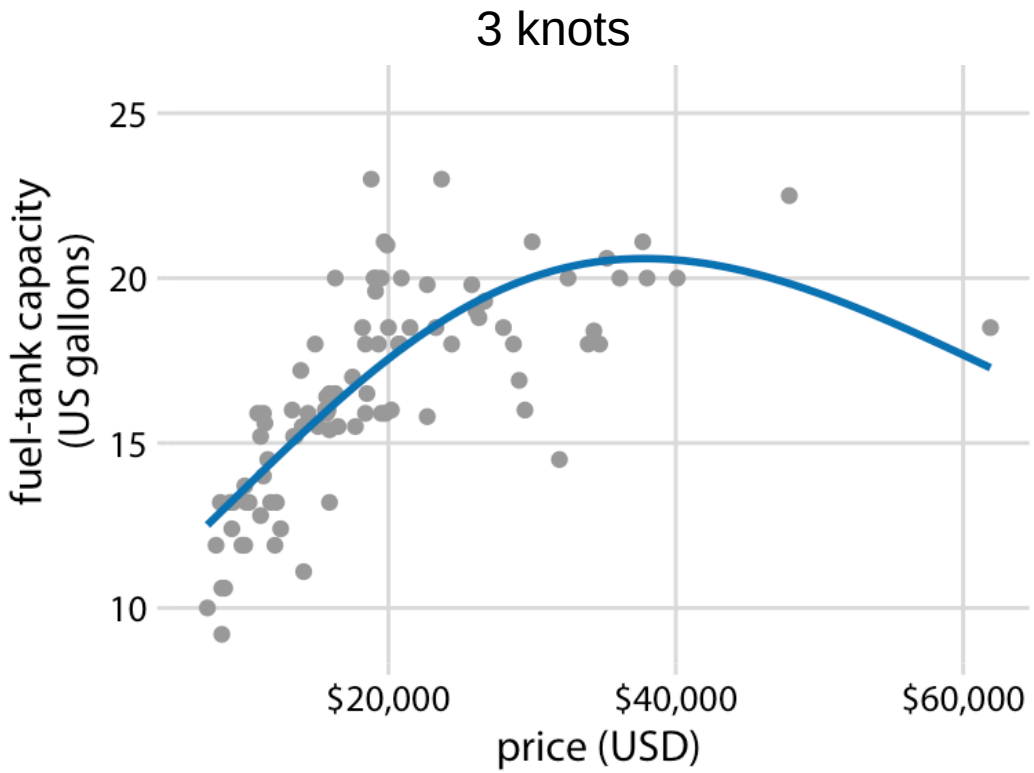
- LOWESS (Locally Weighted Scatterplot Smoothing):
  - Non-parametric regression
  - Integrates elements from k-NN

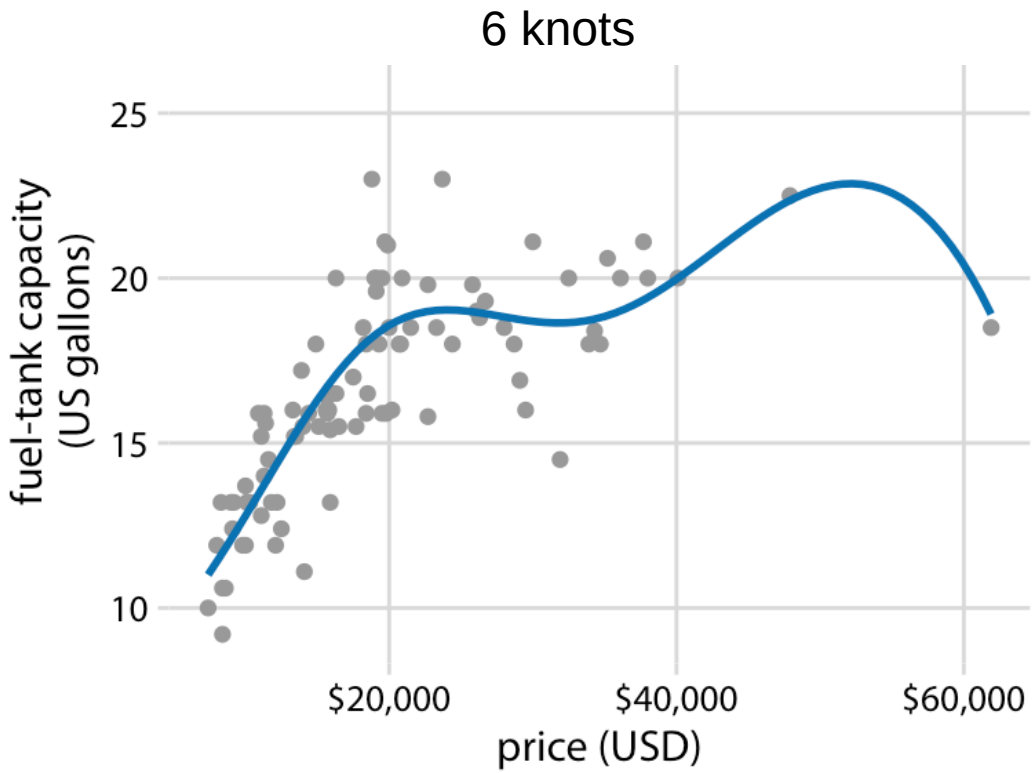
- Idea
  - Predicted value $f(x_i)$ is based on fitting a local, weighted regression in the neighborhood of x_i



Interpreting models: LOESS / LOWESS

- Spline regression uses similar intuition to LOESS
  - Main difference: data divided into segments (knots), speeds compute





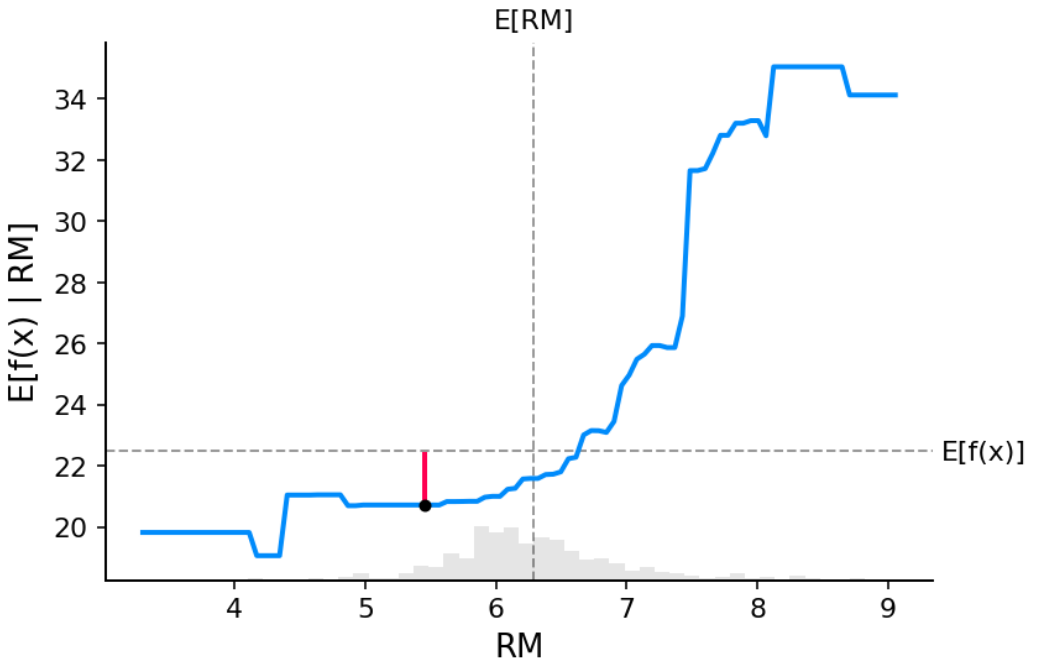
Interpreting models: SHAP plots

- SHAP (SHapley Additive exPlanations)
  - Idea: allocates credit for a model's predictions to the different input features
  - In reality: An NP-hard computation based loosely on game theory
  - In practice: A way to visualize how a model's output varies with a feature
  - In python: `import shap`

Interpreting models: Partial dependence plot

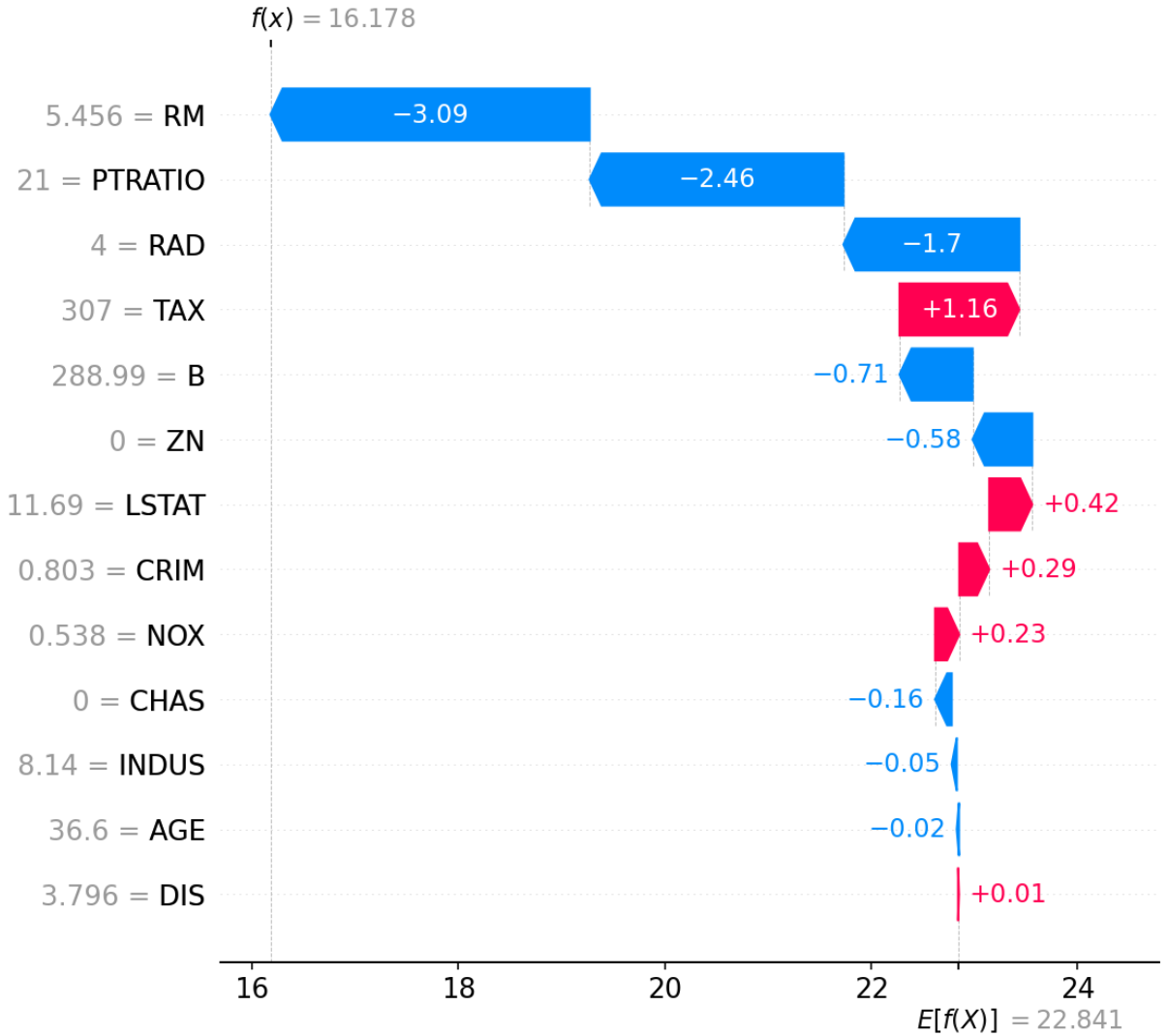
- Partial dependence plot
  - `shap.plots.partial_dependence()`

[11]:
```
# train XGBoost model
import xgboost
model_xgb = xgboost.XGBRegressor(n_estimators=100, max_depth=2).fit(X, y)

# explain the GAM model with SHAP
explainer_xgb = shap.Explainer(model_xgb, X100)
shap_values_xgb = explainer_xgb(X)

# make a standard partial dependence plot with a single SHAP value overlaid
fig,ax = shap.partial_dependence_plot(
    "RM", model_xgb.predict, X, model_expected_value=True,
    feature_expected_value=True, show=False, ice=False,
    shap_values=shap_values_ebm[sample_ind:sample_ind+1,:]
)
```

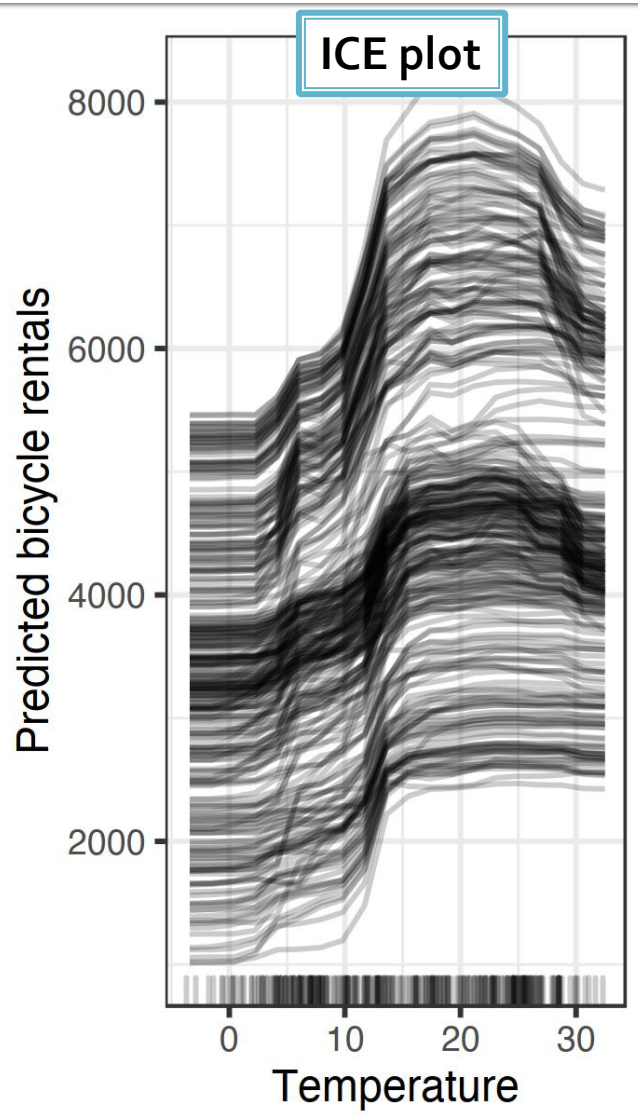


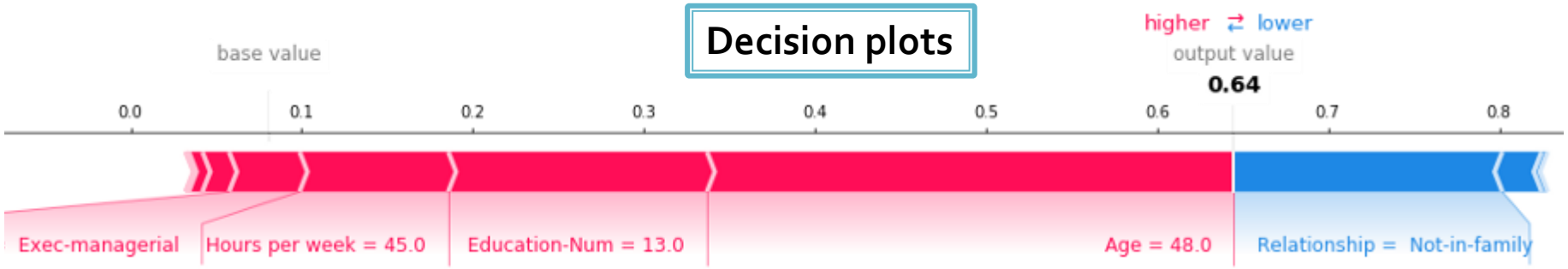
Interpreting models: Waterfall plots

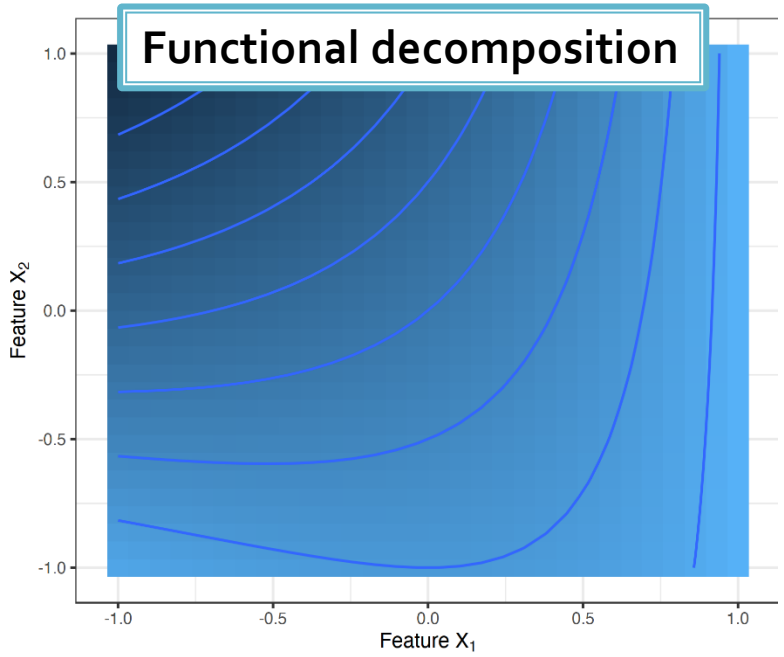
- Waterfall plot
  - `shap.plots.waterfall()`

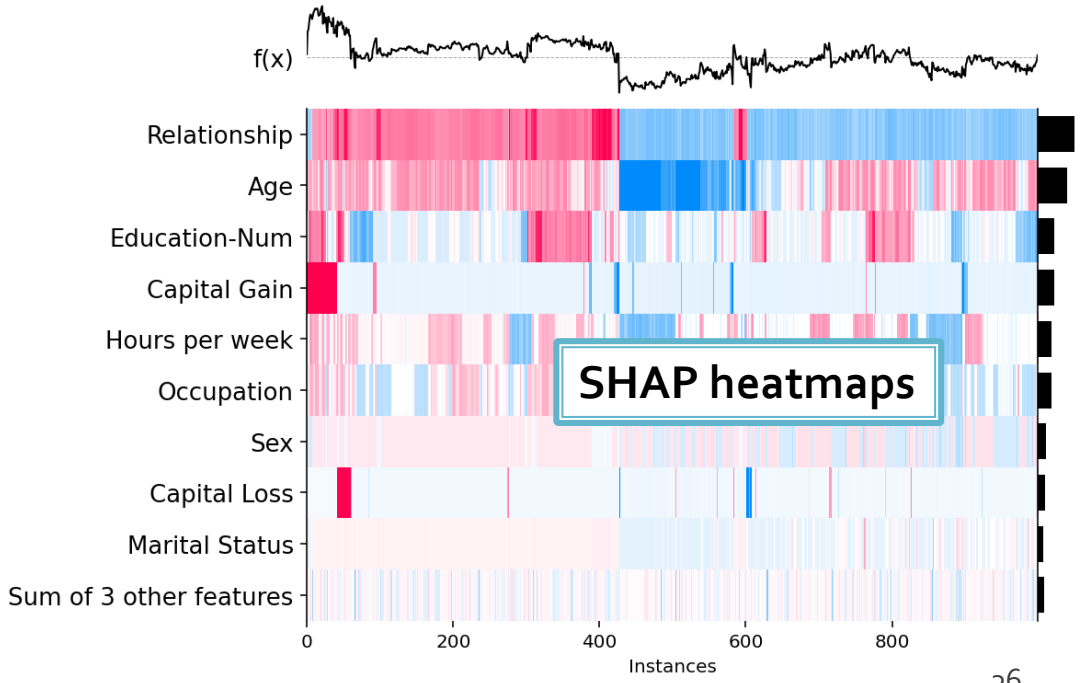


Interpreting models: Down the rabbit hole!

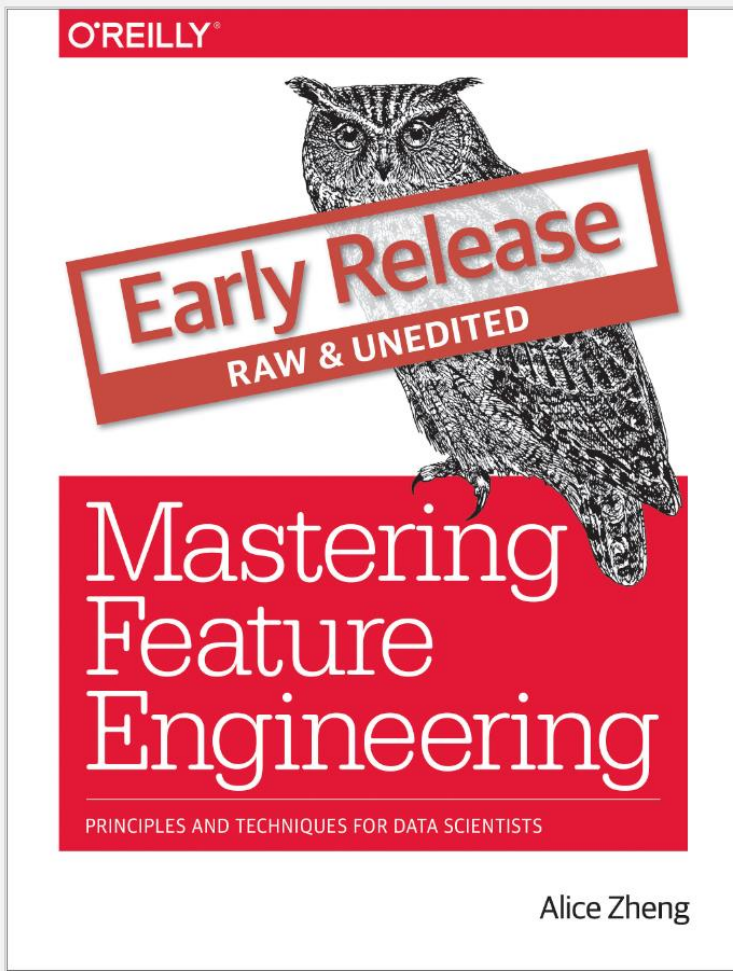


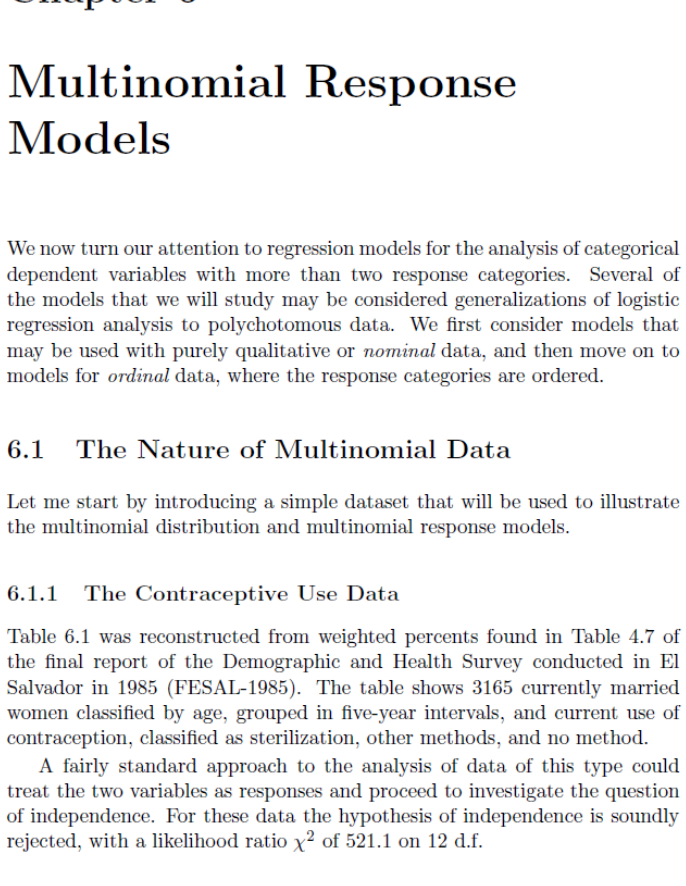






Further reading





Chapter 6

Multinomial Response Models

We now turn our attention to regression models for the analysis of categorical dependent variables with more than two response categories. Several of the models that we will study may be considered generalizations of logistic regression analysis to polychotomous data. We first consider models that may be used with purely qualitative or *nominal* data, and then move on to models for *ordinal* data, where the response categories are ordered.

6.1 The Nature of Multinomial Data

Let me start by introducing a simple dataset that will be used to illustrate the multinomial distribution and multinomial response models.

6.1.1 The Contraceptive Use Data

Table 6.1 was reconstructed from weighted percents found in Table 4.7 of the final report of the Demographic and Health Survey conducted in El Salvador in 1985 (FESAL-1985). The table shows 3165 currently married women classified by age, grouped in five-year intervals, and current use of contraception, classified as sterilization, other methods, and no method.

A fairly standard approach to the analysis of data of this type could treat the two variables as responses and proceed to investigate the question of independence. For these data the hypothesis of independence is soundly rejected, with a likelihood ratio χ^2 of 521.1 on 12 d.f.

G. Rodríguez. Revised September, 2007