Prediction: Classification and Regression

Data 100: Principles and Techniques of Data Science

Sandrine Dudoit

Department of Statistics and Division of Biostatistics, UC Berkeley

Spring 2019

Outline

Outline

5.3 MNIST Handwritten Digit Recognition

Version: 09/04/2019, 18:03

Predicting Rent Using Craigslist Data

- In earlier lectures, we discussed the broad question of finding housing in Berkeley.
- We went through the process of framing the question more precisely and identifying relevant data.
- One of the relevant data sources we explored are listings scraped from Craigslist.
- Rather than "manually" examine each listing matching particular search criteria (possibly in the thousands!) we compute on the listings as follows.
  1. Fetch the HTML page for each Craigslist post matching the criteria and write it to disk.
  2. Process this collection of HTML documents into a data frame.

Predicting Rent Using Craigslist Data

- As illustrated in the notebook from Lecture 07 (`https://github.com/DS-100/sp19/blob/master/lectures/lec07/craigslistEDA.ipynb`), we can then perform exploratory data analysis (EDA) and data cleaning on the listings.
- Here, we will use an up-to-date version of such data to predict rent for Berkeley apartments. The search criteria are `https://sfbay.craigslist.org/search/eby/apa?nh=47&nh=48&nh=49&nh=112&nh=58&nh=59&nh=60&nh=61&nh=62&nh=63&nh=66&nh=64&nh=65&min_price=500&max_price=7500&min_bedrooms=1&min_bathrooms=1&availabilityMode=0&sale_date=all+dates`.
- A CSV file of the listings data and a Jupyter Notebook for their analysis are posted on the class website.

Predicting Rent Using Craigslist Data

- Specifically, we fill consider predicting rent ("price") based on the following rental features extracted from the lisitings:
  - square footage ("sqft"),
  - number of bedrooms ("bedrooms"),
  - number of bathrooms ("bath"),
  - latitude ("lat"), and
  - longitude ("long").
- We randomly divide the dataset of $n = 1271$ listings into a learning set (80% of the listings) to "train" the predictors and a test set (20% of the listings) to assess their performance.

Predicting Rent Using Craigslist Data

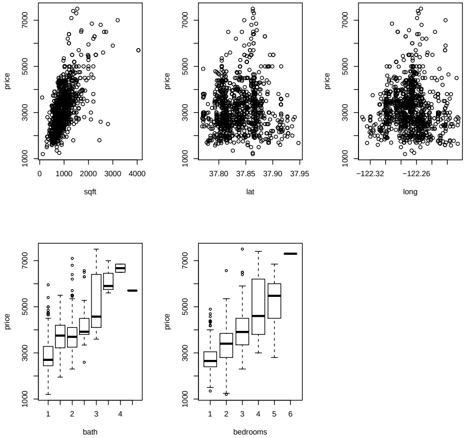


Figure 1: *Craigslist.* Scatterplots of rent vs. five covariates.

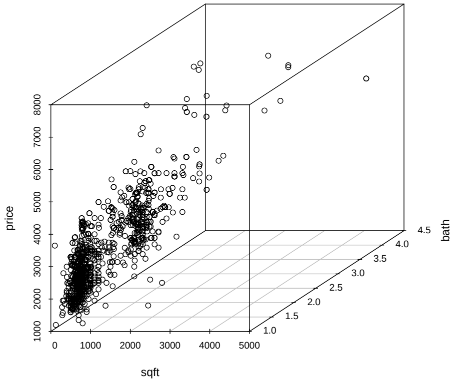


Figure 2: *Craigslist.* 3D scatterplot of rent vs. square footage and number of bathrooms.

MNIST Handwritten Digit Recognition

- A common problem in image processing is handwriting recognition, i.e., the ability to computationally infer handwriting from images.
  E.g. Handwritten zipcodes.
- The handwritten digit recognition question can be framed as a 10-class classification problem, where the outcome of interest is the digit (0 through 9) and the covariates are pixel values.
- The MNIST (Modified National Institute of Standards and Technology) database of handwritten digits provides a learning set of 60,000 gray-scale images of digits from 0 to 9 and a test set of 10,000 such images (`http://yann.lecun.com/exdb/mnist/`).

MNIST Handwritten Digit Recognition

- The digits have been size-normalized and centered in a fixed-size image.
- The website also provides test set error rates for a variety of classifiers, with top performers having error rates as low as 0.5%.
- The learning and test sets can be downloaded in CSV format from
  https://www.kaggle.com/oddrationale/mnist-in-csv.
- Each row in the table corresponds to an image/digit, the first column is the digit label (0 to 9) and the remaining 784 (28×28) columns are the pixel intensities (0 to $2^8 - 1$).

MNIST Handwritten Digit Recognition

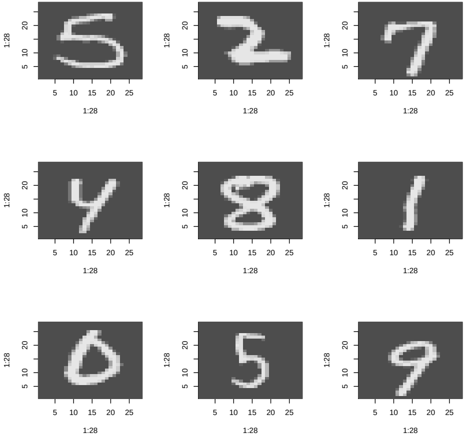


Figure 3: *MNIST digits.* Random sample of 9 images from the learning set.

Prediction: Classification and Regression

- Both the Craigslist and MNIST case studies involve predicting an outcome Y given covariates X.
- However, the outcome for Craigslist is quantitative (i.e., rent), while that for MNIST is qualitative (i.e., one of ten labels corresponding to the digits 0 through 9).
- The terms classification and regression are often used to refer to the prediction of qualitative and quantitative outcomes, respectively.
- Although different types of predictors are used in classification and regression, there are commonalities between the two problems.

- Classification and regression can be handled within the unified general framework of risk optimization, with different loss functions for the different types of outcomes.
- Additionally, some predictors such as trees can handle both qualitative and quantitative outcomes.

Prediction: Classification and Regression

- Let $\mathcal{L}_n = \{(X_i, Y_i) : i = 1, \ldots, n\}$ denote a learning set used to "train" predictors.
- Let $\mathcal{T}_n = \{(X_i^*, Y_i^*) : i = 1, \ldots, n^*\}$ denote a test set used to assess their performance.

Regression

- In regression, the outcome Y is quantitative, i.e., $Y \in \mathbb{R}$. The covariates X can be either qualitative or quantitative. E.g. Rent in Craigslist case study.
- The parameter of interest is the regression function, i.e., the conditional expected value $\theta(X) = \text{E}[Y|X]$ of the outcome Y given the covariates X.
- Let $\hat{\theta}$ denote an estimator of the regression function based on the learning set, e.g., from linear regression.
- The predicted outcome for an observation with covariates X is $\hat{Y} = \hat{\theta}(X)$, i.e., the fitted value for covariates X.

Regression

- The usual loss function for regression is the squared error or L_2 loss function

$$L((X,Y),\theta) = (Y - \theta(X))^2, \quad (1)$$

for which risk is the mean squared error (MSE)

$$R(P,\theta) = \mathrm{E}_P[(Y - \theta(X))^2]. \quad (2)$$

- As the data generating distribution P is unknown, we cannot compute the true population risk.

Regression

- Instead, we can assess performance of a predictor $\hat{\theta}$ using the learning set risk

$$\widehat{MSE} \equiv \frac{1}{n}\sum_{i=1}^{n}(Y_i - \hat{\theta}(X_i))^2 \qquad (3)$$

or, if possible, the more appropriate test set risk

$$\widehat{MSE}^* \equiv \frac{1}{n^*}\sum_{i=1}^{n^*}(Y_i^* - \hat{\theta}(X_i^*))^2. \qquad (4)$$

- Risk can also be estimated using cross-validation, where the learning set is randomly divided into training sets for "training" predictors and validation sets for assessing their performance, i.e., computing risk.

Regression

- In regression, it is also customary to examine the following sums of squares, each representing a different type of variation: The total sum of squares (SST), the error sum of squares (SSE), and the regression sum of squares (SSR) (or "explained" sum of squares).

$$\begin{aligned} SST &\equiv \sum_{i=1}^{n}(Y_i - \bar{Y})^2 \quad (5) \\ SSE &\equiv \sum_{i=1}^{n}(Y_i - \hat{\theta}(X_i))^2 \\ SSR &\equiv \sum_{i=1}^{n}(\hat{\theta}(X_i) - \bar{Y})^2, \end{aligned}$$

where $\bar{Y} = \sum_i Y_i/n$ is the average outcome on the learning set.

- SST is simply n (or $n-1$) times the variance of the outcome on the learning set and SSE n times the learning set MSE.
- The smaller SSE and the larger SSR, the better the fit of the model to the learning data.
- Another useful performance measure (besides MSE) is the coefficient of determination, denoted by R^2 and defined by

$$R^2 \equiv 1 - \frac{SSE}{SST}. \tag{6}$$

Regression

- The coefficient of determination reflects the proportion of variance in the outcome "explained" by the regression function on the covariates.
- As a value between 0 and 1, R^2 is easier to compare across predictors than MSE.
- For linear regression with an intercept, the total sum of squares partitions into the error and regression sums of squares

$$SST = SSE + SSR \tag{7}$$

and

$$R^2 = \frac{SSR}{SST}.$$

Regression

- Although the sum of squares partition does not extend to the test set, one can define a version of the coefficient of determination on the test set as

$$
\begin{aligned}
R^{*2} &\equiv 1 - \frac{SSE^*}{SST^*} \qquad (8)\\
SST^* &\equiv \sum_{i=1}^{n^*} (Y_i^* - \bar{Y}^*)^2 \\
SSE^* &\equiv \sum_{i=1}^{n^*} (Y_i^* - \hat{\theta}(X_i^*))^2,
\end{aligned}
$$

where $\bar{Y}^* = \sum_i Y_i^*/n^*$ denotes the average of the outcome on the test set.

Regression

- The test set R^{*2} reflects the proportion of variance of the outcome in the test set that is explained by a predictor trained on the learning set $(\hat{\theta})$.
- As discussed in previous lectures, using the same dataset for training and assessing the performance of an estimator leads to underestimating risk and to overfitting.
- We therefore compute MSE and R^2 on both the learning and test sets, giving more weight to the latter.

Classification

- In classification, the outcome Y is qualitative, i.e., takes on values arbitrary labeled as $\{1, \ldots, K\}$. The covariates X can be either qualitative or quantitative.
  E.g. Digit in MNIST case study.
- Parameters of interest are (functions of) the conditional class probabilities $\Pr(Y = k|X)$, $k = 1, \ldots, K$.
- A classification function or classifier θ generates a partition of the covariate space $\mathcal{X}$ into K disjoint and exhaustive subsets, $\mathcal{C}_1, \ldots, \mathcal{C}_K$, such that for an observation with covariates $X \in \mathcal{C}_k$ the predicted class is k. That is,

$$\theta(X) = \sum_{k=1}^{K} k \, \mathrm{I}(X \in \mathcal{C}_k). \tag{9}$$

Classification: Logistic Regression

- Logistic regression can be used for binary classification purposes as follows

$$\hat{\theta}(X) = \begin{cases} 0, & \text{if } \widehat{\Pr}(Y = 1|X) = g(X^\top \hat{\beta}) \leq 0.5 \\ 1, & \text{if } \widehat{\Pr}(Y = 1|X) = g(X^\top \hat{\beta}) > 0.5 \end{cases}, \quad (10)$$

  where $g(x) = \exp(x)/(1 + \exp(x))$ is the inverse of the logit function.

- That is, the predicted class is 1 if the estimated conditional probability that $Y = 1$ is greater than 0.5; the predicted class is 0 otherwise.
- Logistic classifiers partition the covariate space into two regions based on the hyperplane defined by

$$g(X^\top \hat{\beta}) = 0.5.$$

Classification: Logistic Regression

- For instance, for two-dimensional covariates $X = (X_1, X_2) \in \mathbb{R}^2$, the partition boundary is the line defined by

$$\hat{\beta}_0 + \hat{\beta}_1 X_1 + \hat{\beta}_2 X_2 = g^{-1}(0.5) = \ln\left(\frac{0.5}{1 - 0.5}\right) = 0.$$

Classification

- To get a sense of the different types of partitions produced by commonly-used classifiers, consider the simulated toy two-class dataset of Figure 4.
- The dataset consists of 1,000 covariate-outcome pairs (X, Y), where the covariates $X = (X_1, X_2) \in R^2$ are two-dimensional and the outcome $Y \in \{0, 1\}$ is binary.
- The two classes can clearly be separated in 2D, but not 1D.
- Furthermore, the classes cannot be separated in 2D by a single line, which is problematic for classifiers such as logistic regression if we do not include interaction terms.

Classification

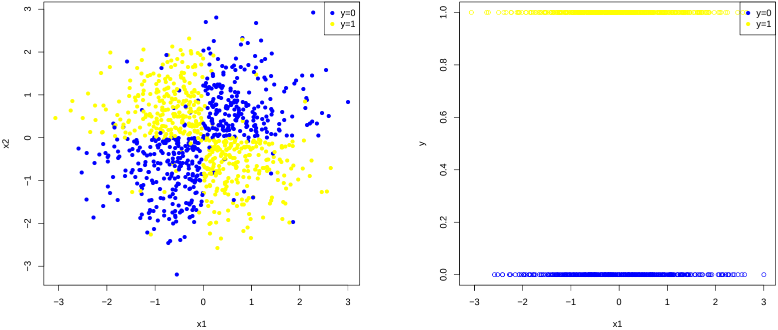


Figure 4: *Classification: Simulated two-class dataset.* The class of each observation is indicated by color.

Classification: Logistic Regression

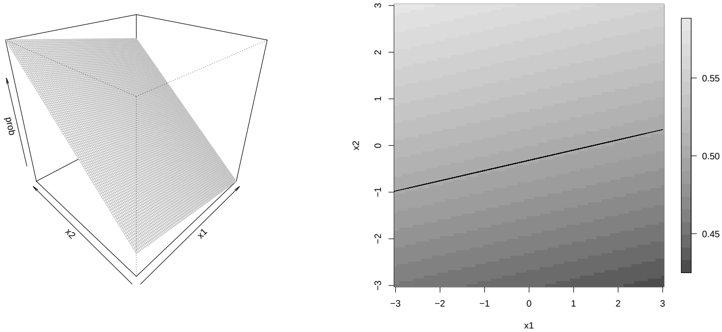


Figure 5: *Classification: Logistic regression.* Fitted probabilities, $\widehat{\Pr}(Y = 1|X) = g(X^\top \hat{\beta})$. The line in the right panel indicates the logistic classifier boundary.

Classification: Logistic Regression

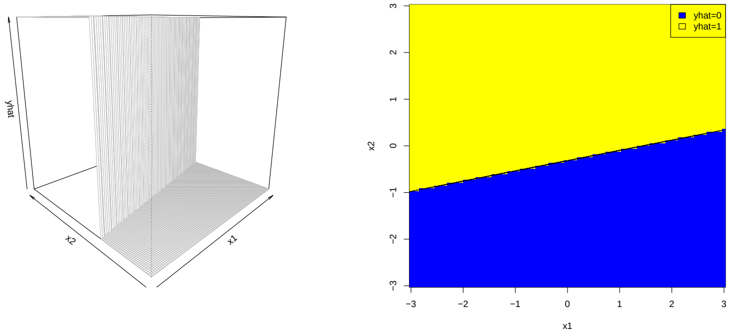


Figure 6: *Classification: Logistic regression.* Classifier partition, obtained by applying a cutoff of 0.5 to the fitted probabilities.

Classification: Logistic Regression

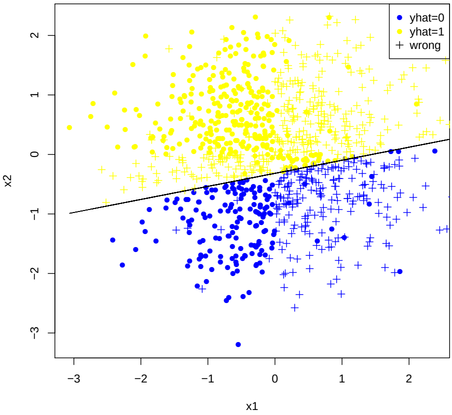


Figure 7: *Classification: Logistic regression.* The predicted class is indicated by color, crosses indicate an incorrect prediction.

Classification: Nearest Neighbors

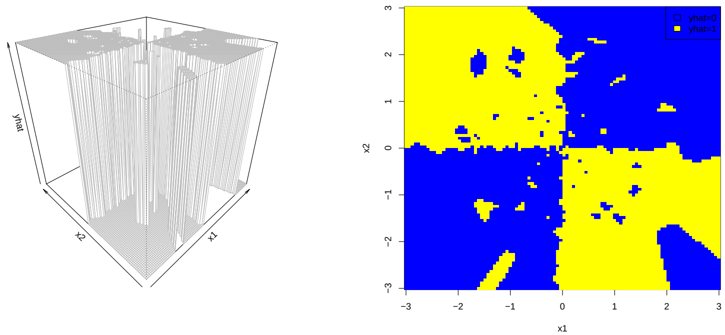


Figure 8: *Classification: k-nearest neighbors.* Classifier partition for $k = 1$.

Classification: Nearest Neighbors

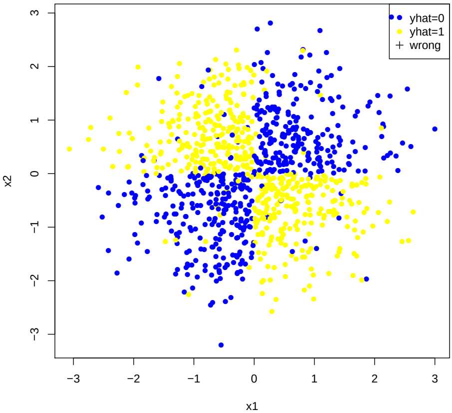


Figure 9: *Classification: k-nearest neighbors.* Predicted classes for $k = 1$.

Classification: Nearest Neighbors

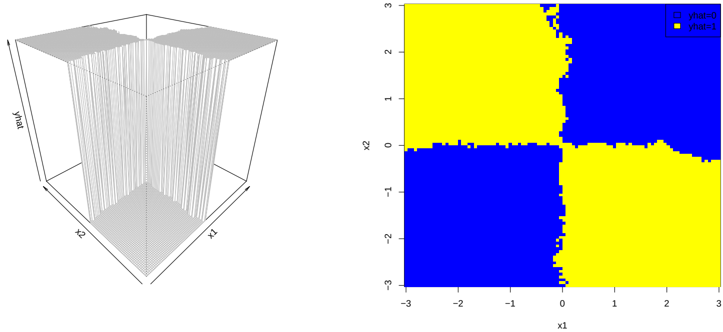


Figure 10: *Classification: k-nearest neighbors.* Classifier partition for $k = 10$.

Classification: Nearest Neighbors

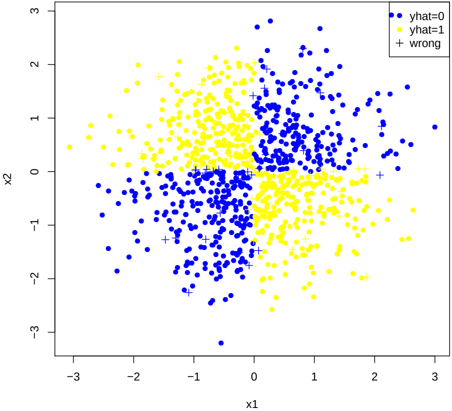


Figure 11: *Classification: k-nearest neighbors.* Predicted classes for $k = 10$.

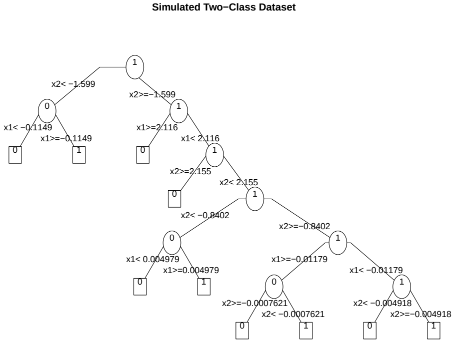


Figure 12: *Classification: Classification trees.* Decision tree.

Classification: Classification Trees

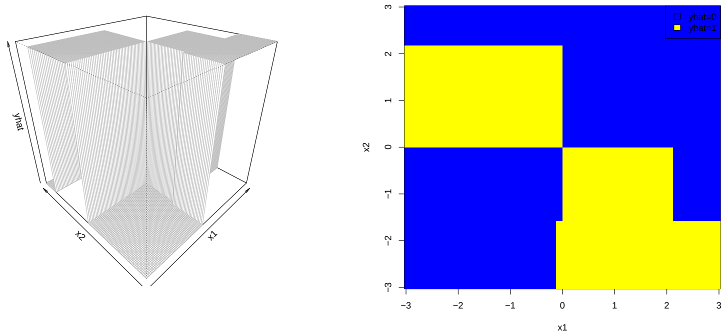


Figure 13: *Classification: Classification trees.* Classifier partition.

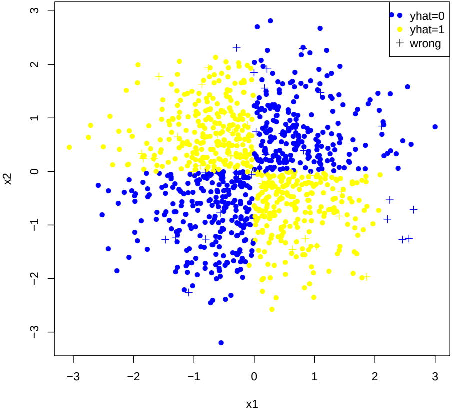


Figure 14: *Classification: Classification trees.* Predicted classes.

Classification

Table 1: *Classification: Simulated two-class dataset.* Learning set classification error rates (%).

Logistic	1-NN	10-NN	CART
55	0	7	6

Classification

- A widely used loss function in classification is the indicator or zero-one loss function

$$
\begin{aligned}
L((X,Y),\theta) &= \mathrm{I}(Y \neq \theta(X)) \qquad (11\\
&= \begin{cases} 1, & \text{if } Y \neq \theta(X) \text{ (incorrect classification)} \\ 0, & \text{if } Y = \theta(X) \text{ (correct classification)} \end{cases}
\end{aligned}
$$

- The indicator loss function can be extended to accommodate different costs for different types of error, e.g., in clinical diagnosis.
- For the indicator loss function, the population risk is simply the probability of an erroneous classification

$$R(P,\theta) = \mathrm{E}_P[\mathrm{I}(Y \neq \theta(X))] = \Pr(Y \neq \theta(X)). \qquad (12)$$

Classification

- The optimal classifier, i.e., risk minimizer, for the indicator loss function yields the class with maximum posterior probability given the covariates X and is known as the Bayes rule

$$\theta(X) = \text{argmax}_k \Pr(Y = k|X). \qquad (13)$$

- In practice, however, the class posterior probabilities are unknown and one relies on the learning set to build a classifier $\hat{\theta}$ that is as close as possible to the Bayes rule in terms of risk.

Classification

- As the data generating distribution P is unknown, we can assess performance of a classifier $\hat{\theta}$ using the learning set risk

$$\widehat{CE} \equiv \frac{1}{n} \sum_{i=1}^{n} \mathrm{I}(Y_i \neq \hat{\theta}(X_i)) \tag{14}$$

or, if possible, the more appropriate test set risk

$$\widehat{CE}^* \equiv \frac{1}{n^*} \sum_{i=1}^{n^*} \mathrm{I}(Y_i^* \neq \hat{\theta}(X_i^*)). \tag{15}$$

These risk estimators are simply proportions of erroneous classifications, referred to as classification error rate.

- Risk can also be estimated using cross-validation.

Craigslist: Linear Regression

- Whenever presented with a question and data as in the Craigslist case study, it makes sense to start with a simple method such as linear regression.
- Here, the outcome Y is "price" and the regression function is of the form

$$\theta(X) = \beta_0 + \sum_{j=1}^{J} \beta_j X_j, \tag{16}$$

where X_j, $j = 1, \ldots, J$, are the $J = 5$ covariates ("sqft", "bath", "bedrooms", "lat", and "long") and the β_j's are the regression parameters to be estimated.

Craigslist: Linear Regression

- We can estimate the coefficients β_j as described in the lecture "Linear Regression", with a design/model matrix comprising 6 columns, the first one for the intercept and the remaining 5 for the covariates.

Craigslist: Linear Regression

Table 2: *Craigslist: Linear regression.* Least squares estimates of the regression coefficients of rent on five covariates. Note that the probabilistic interpretations of standard errors (SE) and *p*-values are only valid to the extent that the underlying modeling assumptions are satisfied. We haven't checked these assumptions here! *p*-values can nonetheless be useful as descriptive summary statistics.

	Estimates	SE	*t*-statistics	*p*-values
(Intercept)	-33321.8836	124980.1698	-0.27	0.7898
sqft	0.8108	0.0758	10.70	0.0000
bedrooms	273.8090	34.6800	7.90	0.0000
bath	395.6733	55.5482	7.12	0.0000
lat	2210.8156	722.4763	3.06	0.0023
long	400.0591	1137.3564	0.35	0.7251

Craigslist: Linear Regression

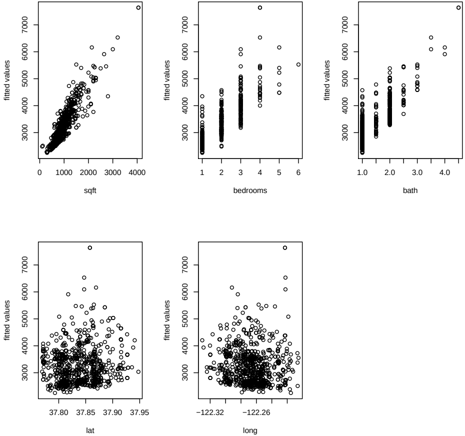


Figure 15: *Craigslist: Linear regression.* Fitted values for regression of rent on all 5 covariates.

Craigslist: Linear Regression

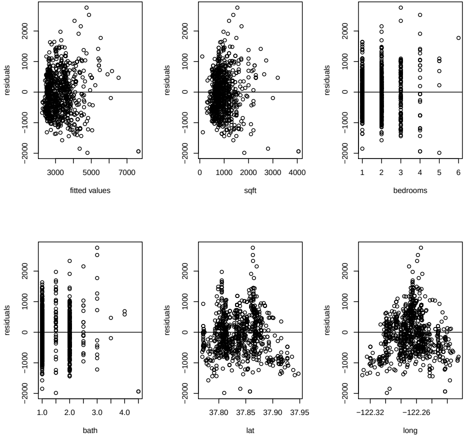


Figure 16: *Craigslist: Linear regression.* Residuals for regression of rent on all 5 covariates.

Craigslist: Linear Regression

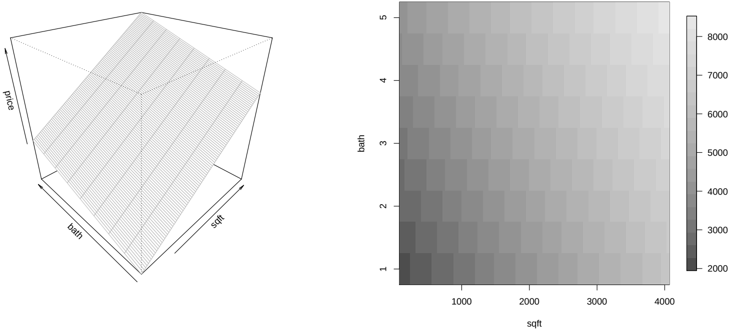


Figure 17: *Craigslist: Linear regression.* Regression function of rent on "sqft" and "bath".

Craigslist: Linear Regression

Table 3: *Craigslist: Linear regression.* MSE and R^2 on learning and test sets for linear regression of rent on all 5 covariates and on only two covariates ("sqft" and "bath").

	MSE		R^2	
	LS	TS	LS	TS
5 covariates	404706	468211	0.52	0.4
2 covariates	439781	497134	0.48	0.36

Craigslist: Linear Regression

- The coefficient of determination on the learning set barely exceeds 50%.
- When using the test set, the MSE is even larger and the coefficient of determination even lower than on the learning set.
- As expected from the scatterplots of Figure 1, the covariates with smallest *p*-values (for testing whether their regression coefficients are zero) are "sqft", "bath", and "bedrooms".
- We therefore also consider fitting a simpler model with only two covariates, "sqft" and "bath" (Figure 17). The increase in MSE and decrease in R^2 are modest.

Craigslist: Linear Regression

- Overall, the prediction accuracy of linear regression is mediocre.
- Can we improve upon linear regression?
- We consider next a completely different type of regression model, a regression tree, which is obtained by recursive binary partitioning of the covariate space.

Craigslist: Regression Trees

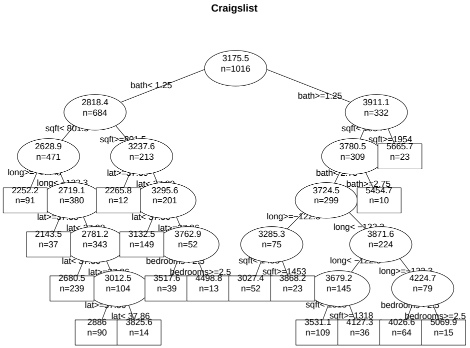


Figure 18: *Craigslist: Regression trees.* Decision tree for regression of rent on all 5 covariates.

Tree-Structured Predictors

- Tree-structured predictors can be used for predicting either qualitative or quantitative outcomes, i.e., for either classification or regression.
- Tree-structured predictors are constructed by repeated splits of subsets of the covariate space $\mathcal{X}$, or nodes, into descendant subsets, starting with $\mathcal{X}$ itself.
- Each terminal node, or leaf, is assigned a fitted value and the resulting partition of $\mathcal{X}$ corresponds to the predictor.

Tree-Structured Predictors

- For a tree, the classification/regression function has the form

$$\theta(X) = \sum_{h=1}^{H} \beta_h \mathrm{I}(X \in \mathcal{A}_h), \tag{17}$$

where the sets $\mathcal{A}_h$ form a partition of the covariate space and β_h is the predicted outcome for an observation with covariates in $\mathcal{A}_h$.

- There are three main aspects to tree construction:
  1. the selection of the splits;
  2. the decision to declare a node terminal or to continue splitting;
  3. the assignment of a fitted value for each terminal node.

Tree-Structured Predictors

- Different tree predictors use different approaches to deal with these three issues. Here, we consider classification and regression trees or, in short, CART (Breiman et al., 1984).
- Other tree predictors are C4.5, FACT, and QUEST; an extensive comparison study is found in Lim et al. (2000).

Classification and Regression Trees

1. Node-splitting rule. At each node, choose the split that maximizes the decrease in empirical risk.
   - Classification. Various loss functions, or impurity measures, have been proposed, e.g., Gini index, entropy, and twoing rule.
   - Regression. The most common lost function is the squared error loss function. One could also consider the absolute or Huber loss functions.
2. Split-stopping rule. Obtaining the "right-sized" tree and accurate estimators of risk can be achieved as follows.
   - Grow a large tree, selectively prune the tree upward, getting a decreasing sequence of subtrees.
   - Use cross-validation to identify the subtree having the lowest risk, i.e., classification error (in classification) or mean squared error (in regression).

Classification and Regression Trees

3. Fitted values. For each terminal node, choose the fitted value that minimizes the empirical risk.
   - Classification. The predicted class is the most common class in the leaf, cf. majority vote.
   - Regression. The fitted value is the average outcome for all the observations in the leaf.

Classification and Regression Trees

- Classification and regression trees have many tuning parameters/inputs, as well as output values in addition to the tree itself and fitted values at the leaves. There are also differences in implementation across software packages. Make sure to consult the documentation to understand how the trees are built and how to interpret the results.
- Trees yield a number of useful by-products, including surrogate splits/variables and variable importance measures.
- A surrogate split is a split based on another variable (surrogate) than the primary variable used for splitting a node, but that partitions the data in a "similar" way.

Classification and Regression Trees

- Surrogate variables are helpful for handling missing values, as the surrogate can be used to split a node when an observation has a missing value for the primary variable.
- An overall variable importance measure can be defined based on the decreases in empirical risk for each node for which the variable is used for either a primary or a surrogate split.

Classification and Regression Trees

- Pros.
  - Applicable to both classification and regression.
  - Can handle categorical covariates naturally.
  - Can handle highly non-linear interactions and classification boundaries.
  - Perform automatic variable selection.
  - Can handle missing values through surrogate variables.
  - Easy to interpret if the tree is small. The picture of the tree can give valuable insights into which variables are important and where.
  - Computationally simple and quick to fit, even for large problems.
- Cons.
  - Unstable, i.e., small changes in the learning set can lead to large changes in the tree. This makes interpretation not as straightforward as it first appears.

Classification and Regression Trees

- Often outperformed in terms of accuracy by methods such as support vector machines (SVM) or even classical linear discriminant analysis or k-nearest neighbors.

Craigslist: Regression Trees

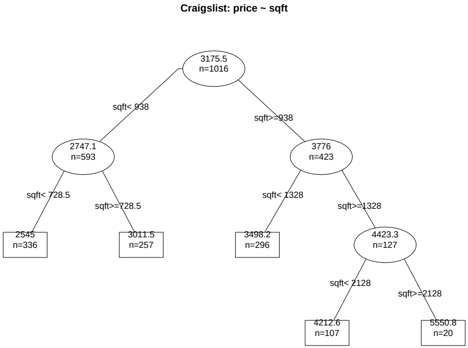


Figure 19: *Craigslist: Regression trees.* Decision tree for regression of rent on "sqft".

Craigslist: Regression Trees

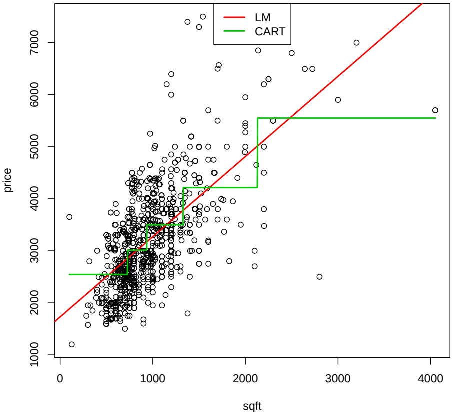


Figure 20: *Craigslist: Regression trees.* Regression function of rent on "sqft", linear regression (red) and regression tree (green).

Craigslist: Regression Trees

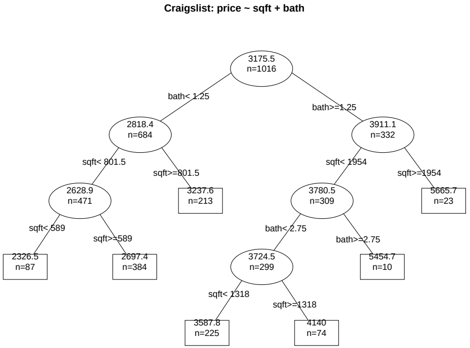


Figure 21: *Craigslist: Regression trees.* Decision tree for regression of rent on “sqft” and “bath”.

Craigslist: Regression Trees

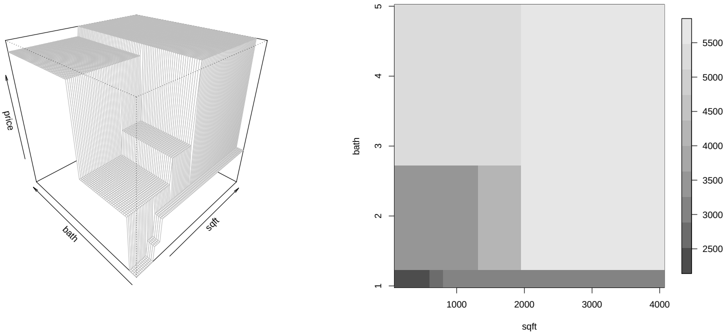


Figure 22: *Craigslist: Regression trees.* Regression function of rent on "sqft" and "bath".

Craigslist: Regression Trees

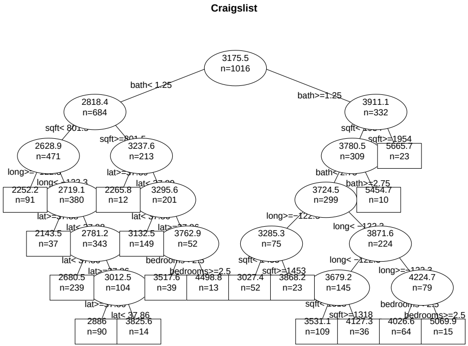


Figure 23: *Craigslist: Regression trees.* Decision tree for regression of rent on all 5 covariates.

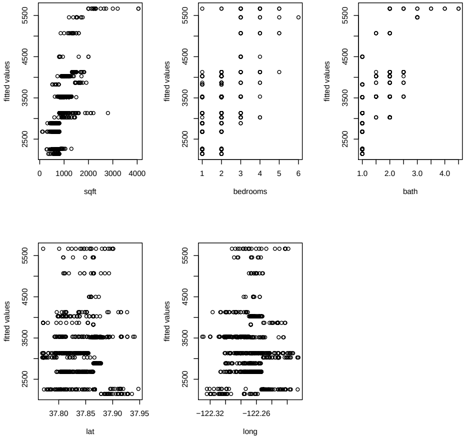


Figure 24: *Craigslist: Regression trees.* Fitted values for regression of rent on all 5 covariates.

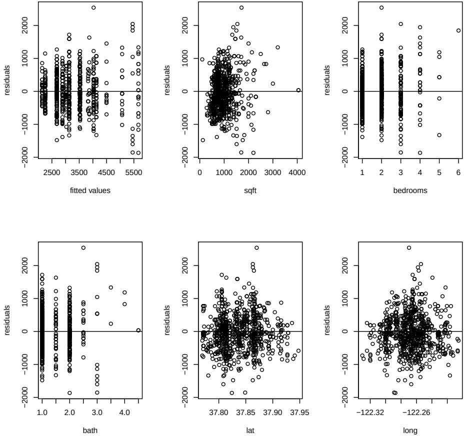


Figure 25: *Craigslist: Regression trees.* Residuals for regression of rent on all 5 covariates.

Craigslist: Regression Trees

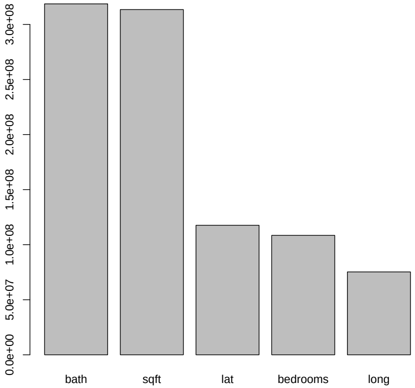


Figure 26: *Craigslist: Regression trees.* Variable importance measures for regression of rent on all 5 covariates.

Craigslist: Regression Trees

Table 4: *Craigslist: Linear regression and CART.* MSE and R^2 on learning and test sets for regression of rent on all 5 covariates.

	MSE		R^2	
	LS	TS	LS	TS
LM	404706	468211	0.52	0.4
CART	279058	345978	0.67	0.56

Craigslist: Regression Trees

- To illustrate the difference between linear regression and tree-based regression, consider simply regressing rent on only one ("sqft") or two ("sqft" and "bath") covariates.
- As seen in Figure 20, for one covariate, the tree yields a step function rather than a line.
- For two covariates, the tree yields a 2D-step function rather than a plane and is better able to capture non-linear relationships and interactions (Figure 22 vs. Figure 17).
- The tree using all 5 covariates is displayed in Figure 23. The first split is on number of bathrooms and other early splits involve square footage.
- The variable importance measures are in agreement with our expectations from EDA and linear regression.

Craigslist: Regression Trees

- With a regression tree, the coefficient of determination is 0.67 on the learning set and 0.56 on the test set, an improvement compared to linear regression.

MNIST Digits: Classification Trees

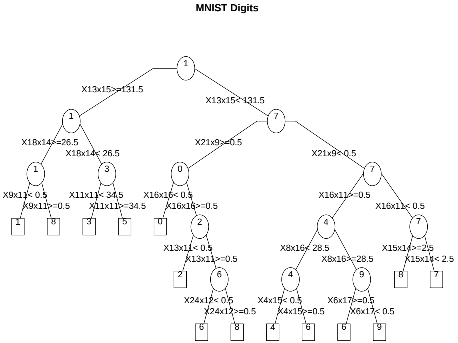


Figure 27: *MNIST digits: Classification trees.* Decision tree for predicting handwritten digits.

MNIST Digits: Classification Trees

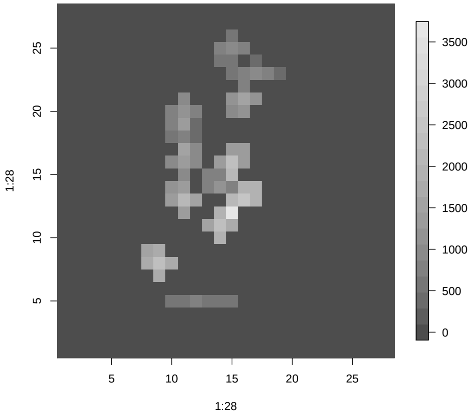


Figure 28: *MNIST digits: Classification trees.* Pseudo-color image of variable importance measures.

MNIST Digits: Classification Trees

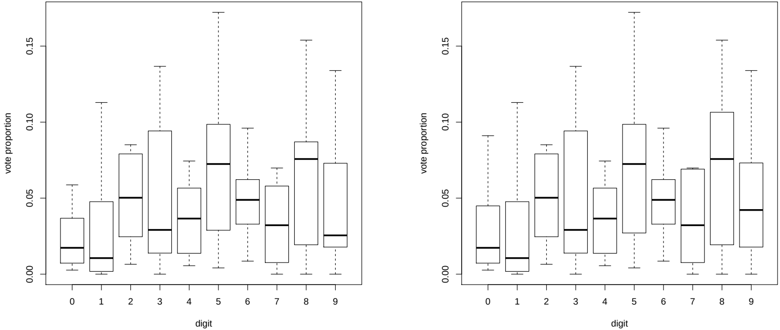


Figure 29: *MNIST digits: Classification trees.* Proportion of votes for each digit for learning set (left) and test set (right).

MNIST Digits: Classification Trees

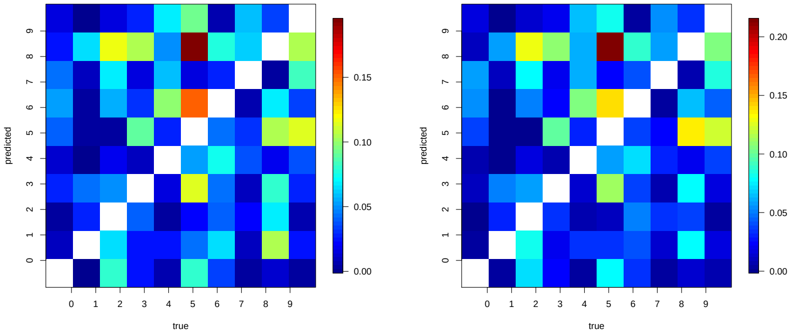
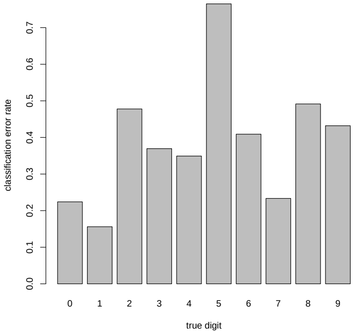


Figure 30: *MNIST digits: Classification trees.* Pseudo-color image of classification error rates for learning set (left) and test set (right), i.e., confusion matrix. Diagonal is blank.

MNIST Digits: Classification Trees



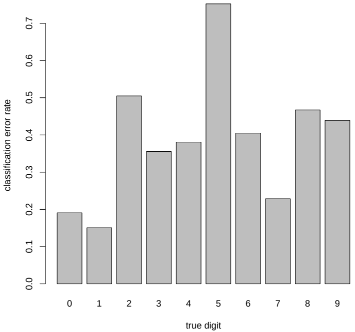


Figure 31: *MNIST digits: Classification trees.* Classification error rates by digit for learning set (left) and test set (right).

MNIST Digits: Classification Trees

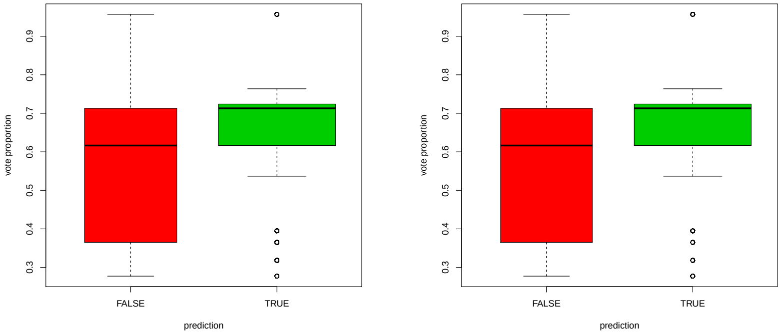


Figure 32: *MNIST digits: Classification trees.* Votes for correct and incorrect predictions on learning set (left) and test set (right).

MNIST Digits: Classification Trees

- As shown in Figure 27, the tree from the learning set has 14 leaves and splits are based primarily on the central pixels which are most informative.
- This is confirmed by examining the variable importance measures in Figure 28.
- Although the ten digits are about equally frequent in both the learning and test sets, some digits tend to have higher vote proportions than others, e.g., "5" and "8" (Figure 29).
- The classification error rates also vary between digits, with, for example, "5" being often misclassified as "8" (Figures 30 and 31).
- Vote proportions for incorrectly classified digits tend to be lower than for correctly classified digits (Figure 32).

MNIST Digits: Classification Trees

- The classification error rate is very high overall (almost 40%) on both the learning and test sets.
- This may be due to the instability of single classification trees. We will turn next to Random Forests to see if gains in accuracy can be achieved by combining predictions from multiple trees, i.e., "averaging".

Ensemble Methods

- A single classification or regression tree can be unstable, i.e., vary greatly with small changes in the learning set.
- Averaging is a natural way to reduce variability.
- This is the main idea behind Random Forests and, more generally, ensemble methods.
- An ensemble predictor can be built by combining the results of
  - the same predictor (e.g., tree) applied to multiple versions of the learning set (e.g., bootstrap samples) or
  - multiple predictors applied to the original learning set.
- In regression, predictions are aggregated by averaging and in classification they are aggregated by voting.

Ensemble Methods

- In bagging (*b*ootstrap *agg*regat*ing*), one aggregates the same predictor built on multiple bootstrap samples of the learning set.
- In boosting, one aggregates the same predictor built on data obtained by repeated adaptive resampling of the learning set, where sampling weights are increased for observations with large prediction errors.

Random Forests

- In Random Forests, one aggregates a forest of many trees, each built on distinct bootstrap samples of the learning set and where subsets of covariates are randomly selected for consideration at each node (https://www.stat.berkeley.edu/~breiman/RandomForests/cc_home.htm).
- Specifically, for each bootstrap sample of the learning set (typically 500), grow a tree as follows.
  - At each node, select a random subset of J' covariates out of all J covariates and find the best split on these selected variables.
  - Grow the trees to maximum depth.
  - Obtain predicted outcomes by voting/averaging over all trees.

Random Forests

- Random Forests yield a number of useful by-products, including variable importance measures, observation proximity measures, and risk estimates.
- The out-of-bag (OOB) observations, i.e., observations not in a bootstrap sample, can be used to obtain risk estimates: For each bootstrap sample, run OOB observations down the corresponding tree and compute empirical risk for that tree, then average empirical risk over all trees.
- There are two main types of variable importance measures for Random Forests: (1) Based on the decreases in empirical risk for splitting over a variable (aggregated over all internal nodes and trees); (2) based on the differences in risk for out-of-bag observations when permuting the values of the variable (aggregated over all trees).

Random Forests

Craigslist: Random Forests

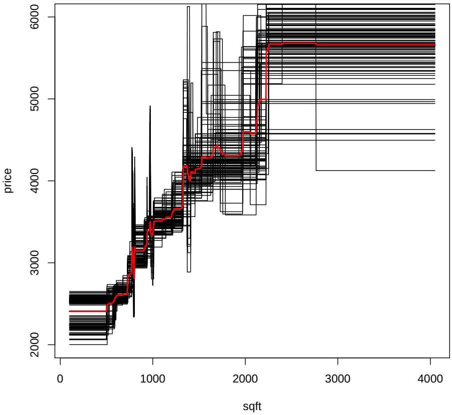


Figure 33: *Craigslist: Random Forests.* Regression function of rent on "sqft" for bootstrap samples of the learning set. Red curve is average.

Craigslist: Random Forests

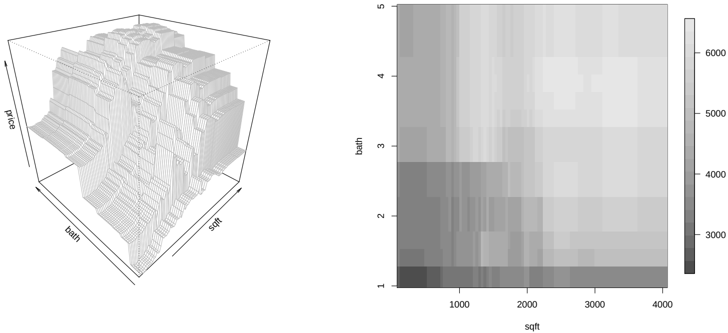


Figure 34: *Craigslist: Random Forests.* Regression function of rent on "sqft" and "bath".

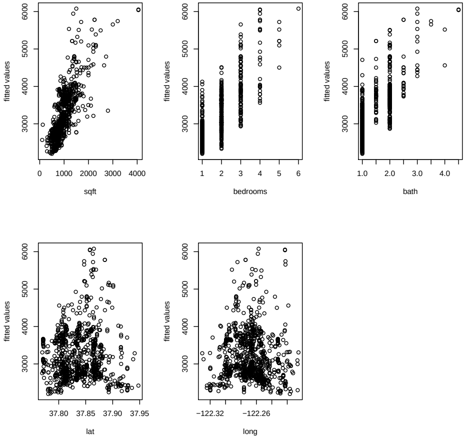


Figure 35: *Craigslist: Random Forests.* Fitted values for regression of rent on all 5 covariates.

Craigslist: Random Forests

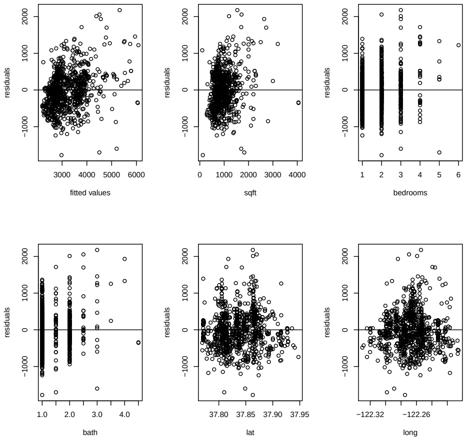


Figure 36: *Craigslist: Random Forests.* Residuals for regression of rent on all 5 covariates.

Craigslist: Random Forests

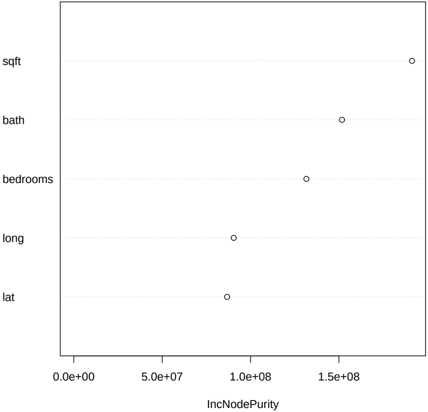


Figure 37: *Craigslist: Random Forests.* Variable importance measures for regression of rent on all 5 covariates.

Craigslist: Random Forests

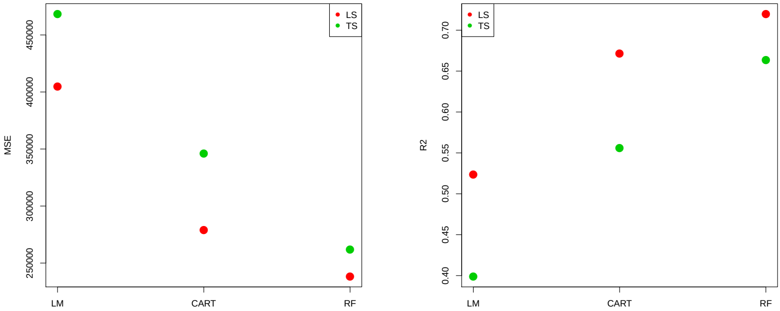


Figure 38: *Craigslist: Linear regression, CART, and Random Forests.* MSE and R^2 on learning and test sets for regression of rent on all 5 covariates.

Craigslist: Random Forests

Table 5: *Craigslist: Linear regression, CART, and Random Forests.* MSE and R^2 on learning and test sets for regression of rent on all 5 covariates.

	MSE		R^2	
	LS	TS	LS	TS
LM	404706	468211	0.52	0.4
CART	279058	345978	0.67	0.56
RF	238313	261960	0.72	0.66

Craigslist: Random Forests

- To illustrate the effect of aggregating multiple trees, we consider simply regressing rent on only one ("sqft") or two ("sqft" and "bath") covariates (Figures 33 and 34).
- Averaging predictions over multiple trees reduces variability and has a smoothing effect on the regression surface.
- The variable importance measures are in agreement with our expectations from EDA.
- Random Forests outperform both linear regression and a single regression tree in terms of MSE and R^2, but the predictive power is still modest.

MNIST Digits: Random Forests

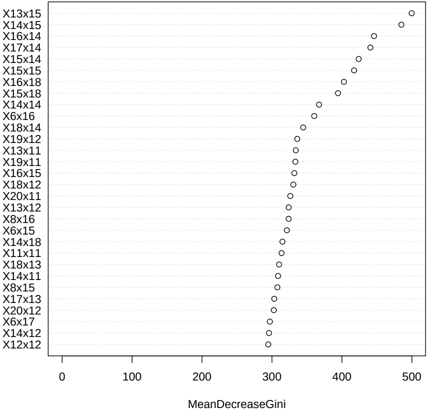


Figure 39: *MNIST digits: Random Forests.* Variable importance measures for learning set.

MNIST Digits: Random Forests

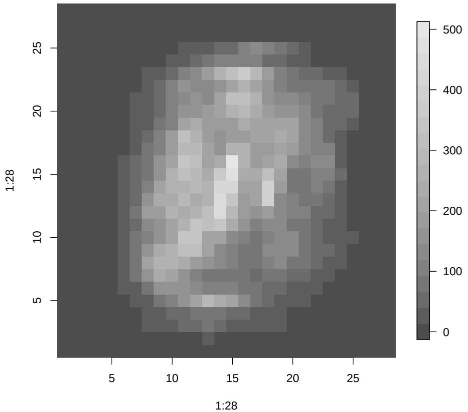


Figure 40: *MNIST digits: Random Forests.* Pseudo-color image of variable importance measures for learning set.

MNIST Digits: Random Forests

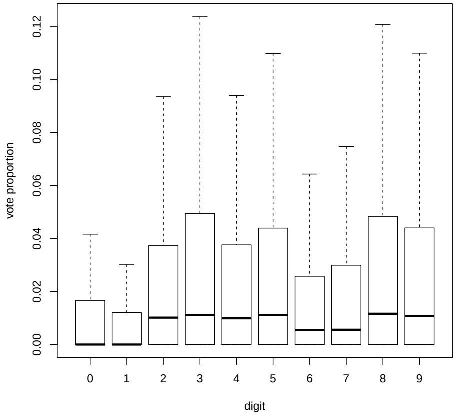


Figure 41: *MNIST digits: Random Forests.* Proportion of votes for each digit for learning set.

MNIST Digits: Random Forests

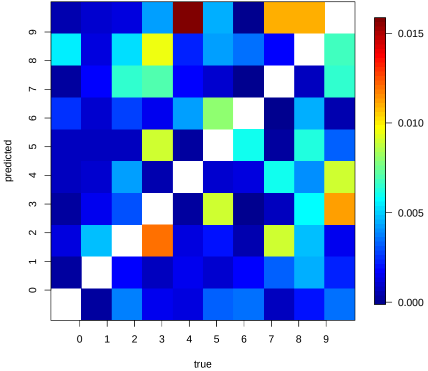


Figure 42: *MNIST digits: Random Forests.* Pseudo-color image of classification error rates for learning set (OOB), i.e., confusion matrix. Diagonal is blank.

MNIST Digits: Random Forests

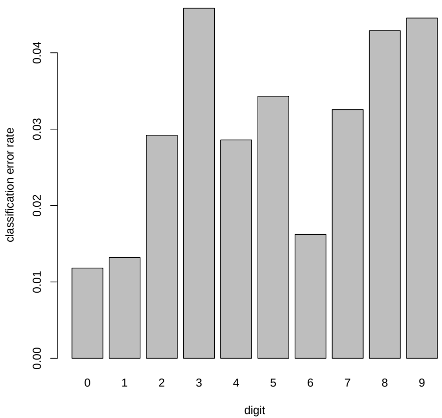


Figure 43: *MNIST digits: Random Forests.* Classification error rates by digit for learning set (OOB).

MNIST Digits: Random Forests

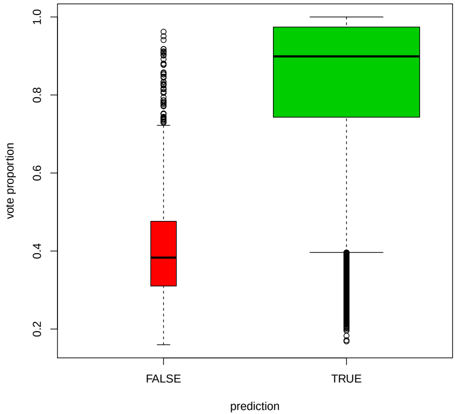


Figure 44: *MNIST digits: Random Forests.* Votes for correct and incorrect predictions on learning set.

MNIST Digits: Random Forests

Table 6: *MNIST Digits: CART and Random Forests.* Classification error rates (%) on learning and test sets.

	LS	TS
CART	38.34	38.04
RF	2.97	2.93

MNIST Digits: Random Forests

- Random Forests provide a major improvement compared to a single classification tree, with both learning and test set classification error rates reduced from around 38% to 3%.
- Variable importance measures again confirm that the most informative pixels are the central ones (Figures 39 and 40).
- Although the ten digits are about equally frequent in both the learning and test sets, some digits tend to have higher vote proportions than others (Figure 41).
- The classification error rates also vary between digits, with, for example, "4" being often misclassified as "9" (Figures 42 and 43).

MNIST Digits: Random Forests

- Vote proportions for incorrectly classified digits are markedly lower than for correctly classified digits (Figure 44).
- In the above analyses, we used the images as provided and CART and Random Forests with default arguments. Improvements in accuracy could perhaps be achieved with further training of the predictors, as was done for classifiers listed at `http://yann.lecun.com/exdb/mnist/`.

References

L. Breiman, J. H. Friedman, R. Olshen, and C. J. Stone. *Classification and Regression Trees*. Chapman & Hall/CRC, Boca Raton, FL, 1984.

T-S Lim, W-Y Loh, and Y-S Shih. A comparison of prediction accuracy, complexity, and training time of thiry-three old and new classification algorithms. *Machine Learning*, 40:203–229, 2000.