INFO 247 Final Project Report

A Path to Random Forest

Yiyi Chen and Sam Meyer | May 9, 2017

Project Goals

Inspired by the existing visualizations on machine learning algorithms (such as this one on Decision Trees), we chose to create our own visual story for an introduction to the random forest algorithm. Our goal was to visualize and explain the inner mechanism and training process of the algorithm to someone who might be new to machine learning and/or have little familiarity with the algorithm. We also wanted to explore different D3 visualization techniques, especially animations, to show how the algorithm works in the context of solving a specific problem.

In particular, we hope our visualization accomplishes two goals -

- Enhance people's understanding of the random forest algorithm
- Provide a more interactive and engaging learning experience compared to the traditional all-text explanation

We designed our visualization with the two goals in mind, and conducted our user testing based on those goals.

Related Work

Choosing Color for Data Visualization

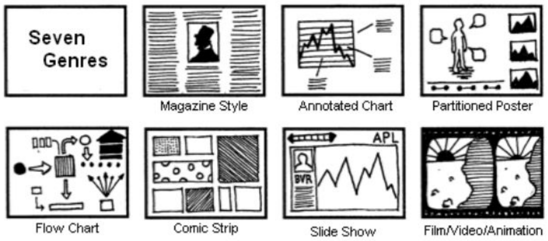
Ston suggested that "in most design situations, the best results are achieved by limiting hue to a palette of two or three colors, and using hue and chroma variations within these hues to create distinguishably different colors". We followed the same principle and used 3 base hues throughout the visualization - 2 for the two classes in the binary classification problem and 1 for the layout structure - and varied their lightness and saturation to create different colors for the visualization. We also chose split complementary colors, turquoise and purple, for the 2 class classification, as they are contrasting but not too dissonant.

Base hues

Color 1 (poisonous mushroom)	Color 2 (edible mushroom)	Class 3 (overall layout)
🟪	🟩	◼

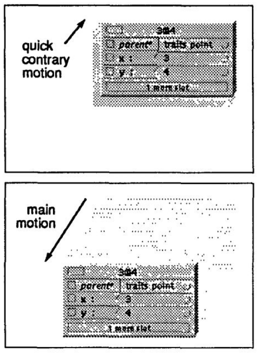
Narrative Visualization: Telling Stories with Data by Edward Segel and Jeffrey Heer

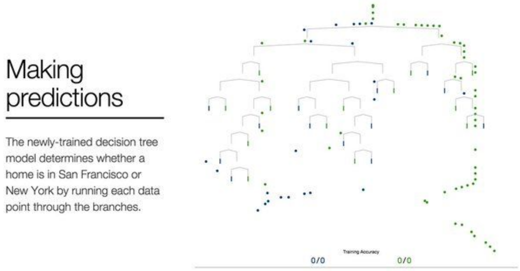
Segel and Heer lay out many forms for data visualization, providing a palette for designers to think about new designs. They show how visualizations can take on forms similar to posters, slide shows, annotated charts, and others. For our project of explaining the mechanism of random forest, we chose a very linear flow. This is appropriate because the process itself is linear. Within visualizations, we allow a bit more exploration to engage users and let them explore the data in their selected order. Because we are not exploring a single data set, but are explaining distinct parts of an algorithm, our story is similar to an interactive slideshow where each visualization stands alone but may allow some interactivity within that visualization.



Animation: From Cartoons to User Interface

Chang and Ungar bring the lessons of animation to bear on user interface, which is important for our project that uses animation to connect steps within a process. Rather than existing visualizations of random forest which depend on static images, we are using animation because it creates connection between steps. For important motions, the audience's eye will be attracted by exaggeration using anticipation before the action, and by follow through after the action. For more normal actions, motion can be reinforced using slow-in slow-out.





Making predictions

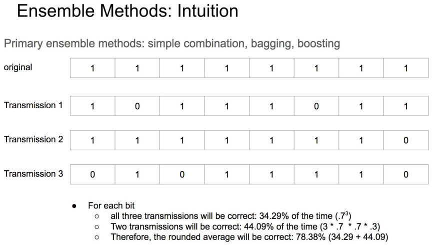
The newly-trained decision tree model determines whether a home is in San Francisco or New York by running each data point through the branches.

R2d3: Introduction to Machine Learning

Yee and Chu made machine learning approachable with their work on "A Visual Introduction to Machine Learning." This project walks the user through the process of training a decision tree. It is notable for its goal of making machine learning approachable for those who have never encountered it before. A combination of engaging visuals and trees that allow for interactivity, it inspires us to make more of machine learning available to users.

Ensemble Methods: Intuition (slide 5)

Zachary Pardos used a signal transmission example in his Data Mining and Analytics class (INFO 290T) to explain the intuition behind ensemble methods. We adopted the logic to visually demonstrate why a crowdsourced algorithm is more robust against noise than individual algorithms. Based on the user testing feedback, we used the same mushroom example in that visualization to be consistent with the rest of the site.



Ensemble Methods: Intuition

Primary ensemble methods: simple combination, bagging, boosting

original	1	1	1	1	1	1	1	1
Transmission 1	1	0	1	1	1	0	1	1
Transmission 2	1	1	1	1	1	1	1	0
Transmission 3	0	1	0	1	1	1	1	0

- For each bit
  - all three transmissions will be correct: 34.29% of the time ($.7^3$)
  - Two transmissions will be correct: 44.09% of the time (3 * .7 * .7 * .3)
  - Therefore, the rounded average will be correct: 78.38% (34.29 + 44.09)

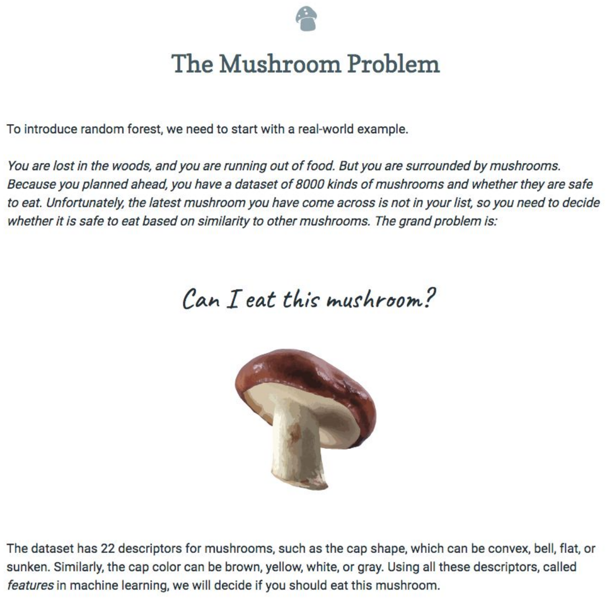
Distill Research Journal
Distill is a modern machine learning journal dedicated to clear explanations of machine learning. When we brainstormed the layout and composition of our site, we deliberated between 2 approaches: the first having the text explanation and visualization side by side, and the second having a vertical scroll of alternating text and visualization. We ultimately decided on the latter, which is used by Distill, as we felt it makes a smoother logical progression and renders a nice flow when we have a relatively long section of text.

Visualization Description

We divided the process of training and using random forest into various sections to guide users. Each has a header on the web page that matches the headers below.

The Mushroom Problem

We began with an introduction to the problem of deciding whether a mushroom is poisonous or not. We introduce the dataset and some example features to give readers an understanding of the value of classification problems. We set off a playful story in italics to make clear that we are setting a stage for the problem rather than giving instructions on complicated machine learning (Fig. 1). The story relaxes readers who may consider "algorithms" to be distant and inscrutable.



The Mushroom Problem

To introduce random forest, we need to start with a real-world example.

You are lost in the woods, and you are running out of food. But you are surrounded by mushrooms. Because you planned ahead, you have a dataset of 8000 kinds of mushrooms and whether they are safe to eat. Unfortunately, the latest mushroom you have come across is not in your list, so you need to decide whether it is safe to eat based on similarity to other mushrooms. The grand problem is:

Can I eat this mushroom?

The dataset has 22 descriptors for mushrooms, such as the cap shape, which can be convex, bell, flat, or sunken. Similarly, the cap color can be brown, yellow, white, or gray. Using all these descriptors, called *features* in machine learning, we will decide if you should eat this mushroom.

Figure 1. Introductory story and image

Decision Trees

Here, we needed a basic introduction to decision trees, as random forest depends on them. However, we did not want to need to replicate all the work done by longer explanations. As a result, we used a text explanation to bring readers up to speed on decision trees, then included an annotated decision tree for readers to explore. (Fig. 2) Three annotations were needed to explain all parts of the decision tree.

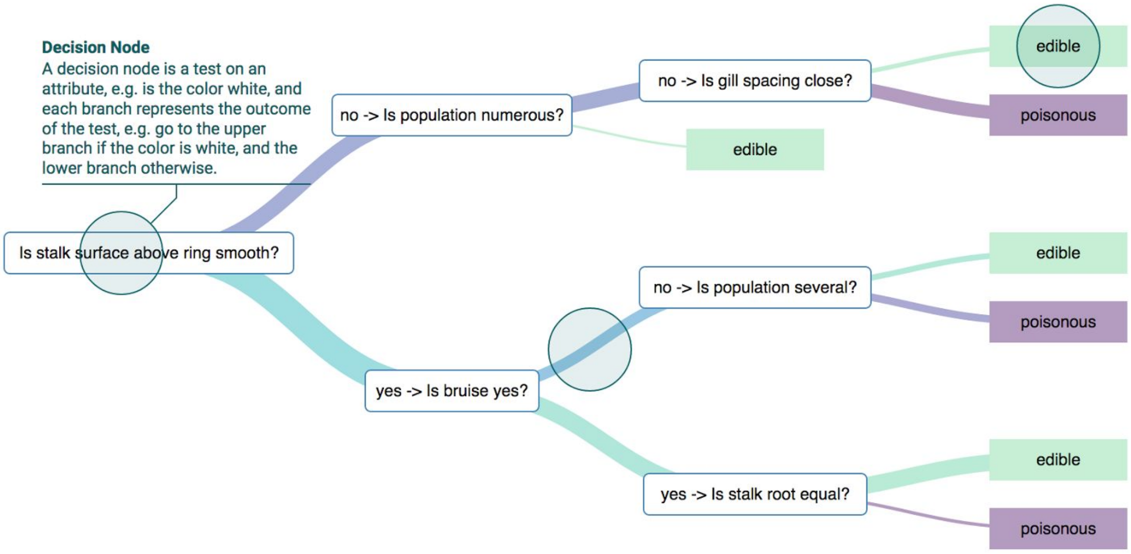
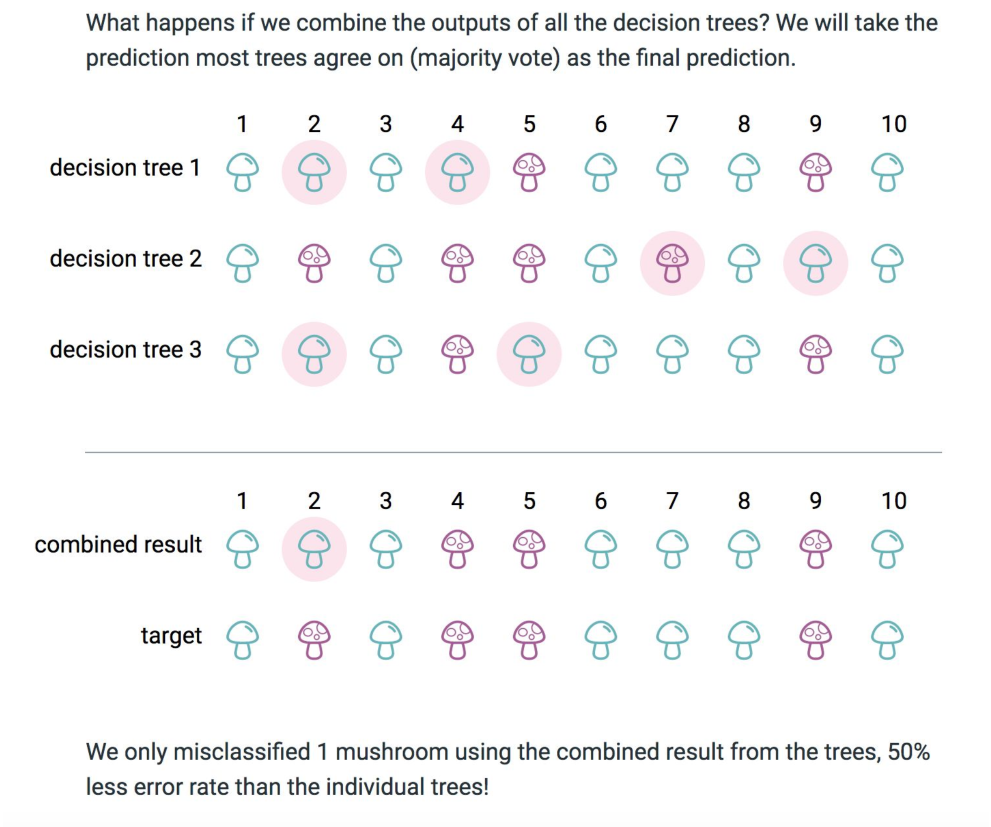

Decision Node
A decision node is a test on an attribute, e.g. is the color white, and each branch represents the outcome of the test, e.g. go to the upper branch if the color is white, and the lower branch otherwise.
Is stalk surface above ring smooth?
no -> Is population numerous?
no -> Is gill spacing close?
edible
poisonous
edible
yes -> Is bruise yes?
no -> Is population several?
edible
poisonous
yes -> Is stalk root equal?
edible
poisonous

Figure 2. Decision tree with annotations

Fallacy of Individual Predictors

Here, we introduce the idea of ensemble predictors, using multiple predictors together. Using isotypes, we illustrate how each decision tree can make mistakes on predicting which mushrooms are poisonous, and together they can be better predictors. We use the same mushroom classification example to maintain consistency throughout the visualization. (Fig. 3)



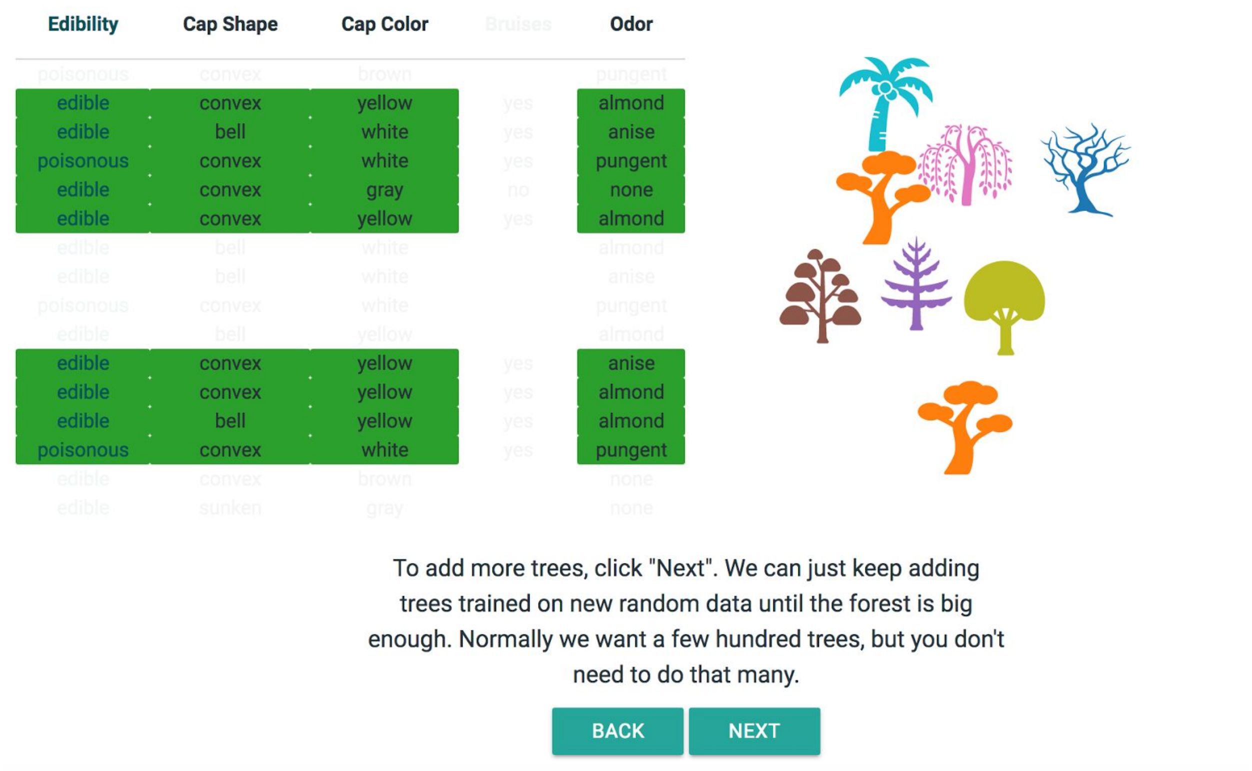
What happens if we combine the outputs of all the decision trees? We will take the prediction most trees agree on (majority vote) as the final prediction.

We only misclassified 1 mushroom using the combined result from the trees, 50% less error rate than the individual trees!

Figure 3. Combining decision trees

Training of Trees

Here, we walk users through the process of selecting data for a tree. We show highlighting of first rows and then columns, and we reduce the opacity of unselected data to show that it is not being used. Breaking these into steps allows users to think about selecting both the rows (mushrooms) and the columns (features). The color changes first to a dark blue and then to a lighter blue to attract users attention to each row as it is selected. Selected rows and columns have the same color used when each is selected. Once both are selected, the selected data turn green and other data has reduced opacity, using preattentive properties to group the selected data. The green dot that moves away from the data maintains color continuity to logically connect it to the selected data. It anticipates movement by backing up before moving, drawing the reader's attention. Tree growth animation is paused to give users a sense of the time it takes to train each tree on the selected data. Finally, users can make a forest of trees (Fig. 4). At this point, the visual metaphor of a forest is visible.



Edibility	Cap Shape	Cap Color	Bruises	Odor
poisonous	convex	brown		pungent
edible	convex	yellow	yes	almond
edible	bell	white	yes	anise
poisonous	convex	white	yes	pungent
edible	convex	gray	no	none
edible	convex	yellow	yes	almond
edible	bell	white		almond
edible	bell	white		anise
poisonous	convex	white		pungent
edible	bell	yellow		almond
edible	convex	yellow	yes	anise
edible	convex	yellow	yes	almond
edible	bell	yellow	yes	almond
poisonous	convex	white	yes	pungent
edible	convex	brown		none
edible	sunken	gray		none

To add more trees, click "Next". We can just keep adding trees trained on new random data until the forest is big enough. Normally we want a few hundred trees, but you don't need to do that many.

Figure 4. Walkthrough of creating trees for the forest

Training Results

Here, we show how each tree that was trained on its own set of data creates a different decision tree. Users can select a tree and drill down into the features that tree uses to make decisions. A legend at the bottom explains the meaning of each visual component to the user. The goal of the visualization is to demonstrate that with different training data and feature input, each tree is constructed differently.

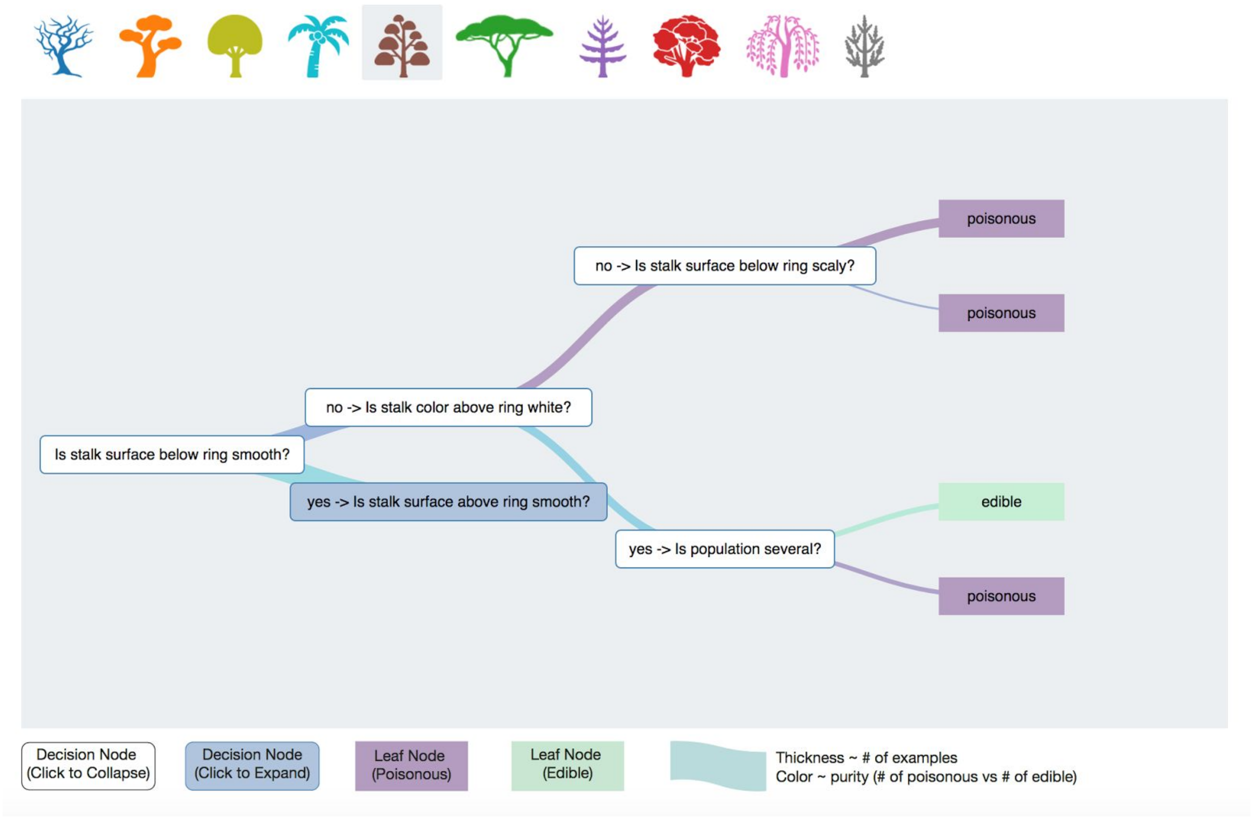
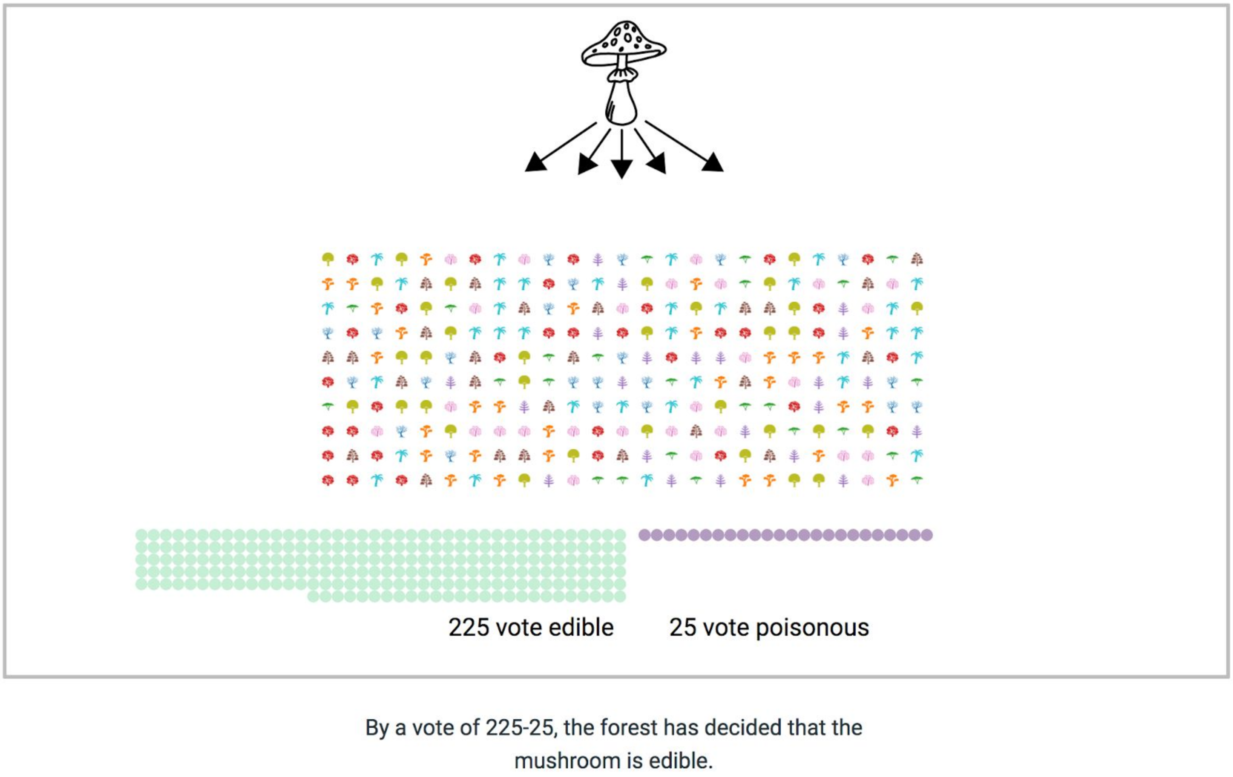

Is stalk surface below ring smooth?
no -> Is stalk color above ring white?
yes -> Is stalk surface above ring smooth?
no -> Is stalk surface below ring scaly?
yes -> Is population several?
poisonous
poisonous
edible
poisonous
Decision Node (Click to Collapse)
Decision Node (Click to Expand)
Leaf Node (Poisonous)
Leaf Node (Edible)
Thickness ~ # of examples
Color ~ purity (# of poisonous vs # of edible)

Figure 5. Details of individual trees

Making a new prediction

Here, we show how a random forest works as a unit to decide on whether a new mushroom is poisonous or not. We animate copies of the mushroom being sent to each tree. When the user clicks “Next”, each tree decides whether the mushroom is safe to eat or not. Based on a vote (on clicking “Next” again), the forest decides that the mushroom is safe to eat (Fig. 6). We animate the moving votes to keep continuity. The number of trees was increased after experienced testers complained that we misrepresented the normal number of trees in a random forest. As a result, we used 250 trees rather than 5.

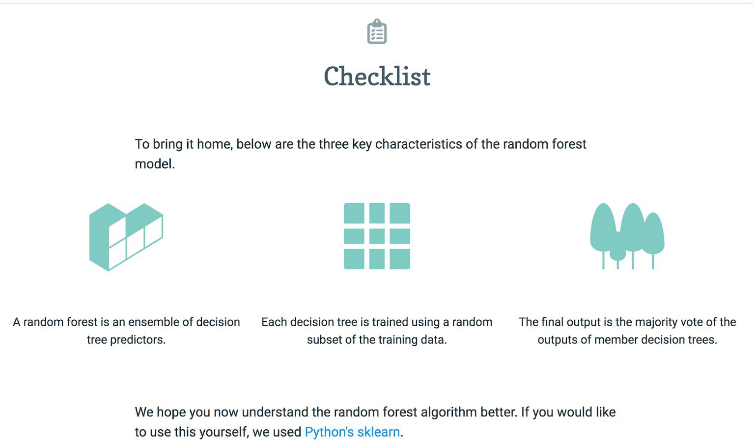


By a vote of 225-25, the forest has decided that the mushroom is edible.

Figure 6. A random forest decides whether a mushroom is safe to eat using a vote

Checklist

To summarise and remind the reader of important points, we show some major points (Fig. 7). This helps to reinforce important ideas that the reader should have learned from the page.



Checklist

To bring it home, below are the three key characteristics of the random forest model.

A random forest is an ensemble of decision tree predictors.

Each decision tree is trained using a random subset of the training data.

The final output is the majority vote of the outputs of member decision trees.

We hope you now understand the random forest algorithm better. If you would like to use this yourself, we used Python's sklearn.

Figure 7. Checklist at end of page

Approach

Data

While the intention of the visualization is to explain the random forest algorithm, it was important to illustrate it with a fun dataset. We chose a mushrooms dataset from the UCI machine learning repository (compiled by Jeff Schlimmer) with a sample of 8000+ rows and 22 categorical attributes, e.g. cap shape and color, corresponding to 23 species of gilled mushrooms in the Agaricus and Lepiota Family. Each row is classified as poisonous or edible.

In order to make the data readable for our users for the visualization demonstration, we added a descriptive name for each feature. The original values for each feature are represented by a single character key, so we recoded the labels back to their mapped meaning using the lookup in the readme file, e.g. “w” -> “white.” For the training result visualization, we trained a random forest model using Python’s sklearn library, and the code directly loads the output as a json file to generate the trees.

Tools

For the random forest analysis, we used Python’s sklearn implementation of RandomForestClassifier, supported by Pandas. Small other edits were done in Excel. The webpage was made using D3.js, jQuery, HTML, and CSS and included the Materialize web framework. Some illustrations were made using Adobe

Illustrator. Decision Trees were greatly based on Peter Prettenhofer's example, but we updated it to be compatible with D3 v4. Annotations on the decision tree used Susie Lu's D3 Annotation Library.

Full image and code lists are in Appendix A, and all used images are referenced at the bottom of the page.

Steps

1. **Random Forests Model Building and Result Generation**
   We first selected our mushroom dataset and loaded it into Python using Pandas. We then converted the poisonous/edible column to a binary value, then split the results into training and test sets. We ran RandomForestClassifier from the sklearn package. We then exported the tree descriptions as JSON we could use for our webpage.
   Tools/libraries: Python Pandas, Excel, Sklearn

2. **Initial Prototype**
   We created an initial static webpage with auto-generated trees and simple images made in Illustrator. This allowed us to discuss decisions about what would be included in each section internally and with outsiders who gave feedback. Based on feedback on the initial design, we increased our explanations of how decision trees work before continuing to the full design.
   Tools/libraries: HTML, CSS, Materialize, Illustrator

3. **Information Visualization and Storytelling**
   We divided the work and separately created visualizations and text. The work went through iterations of text and visualization order. Initially, we added a section to explain our dataset in the middle of the page, but later moved it to the top as we found we needed to set the goal of the task before explaining anything about the algorithm. As a final step, we added styling for the title and section headings to clearly delineate sections.
   Tools/libraries: Info viz: Javascript, D3. Frontend framework: Materialize

4. **User Testing and Refinement**
   We tested the design with 5 students who were somewhat familiar with random forest but did not feel that they fully understood it. We also collected feedback from many other people who viewed the tool, some of whom were very familiar with the algorithm. Based on this feedback, we updated the page. Full details are below in the user testing section.
   Tools/libraries: Paper forms

User Testing and Results

Overview

We conducted user testing with 5 prospective users. Additionally, we also collected additional informal feedback from others during our demo day. All prospective users were master students in UC Berkeley, who had heard of but were not very familiar with the random forest algorithm. Some of the users also had a background in UX. The convenience sample was therefore fairly appropriate, as students interested in machine learning are part of our target audience for the visualization. The user testing participants helped us better understand the different interactions a user would have with our site, and provided extremely valuable feedback for areas of improvement for our design both from the perspective of a user and a UX professional.

We aimed to test both the functional and aesthetic qualities of our design. Specifically, we wanted to understand

1. If our visualization led to an observable improvement in people's understanding of the random forest algorithm, and
2. If and how much people enjoyed the visualizations.

Test Setup

For our user testing, we used a combination of survey, observation and semi-structured interview. Before the participant interacted with our site, we asked them to answer 3 questions about random forest to baseline their prior knowledge of the algorithm. The participants would then go on the site, and we would observe how they interacted with different visualizations. After they finished, we asked them to answer another set of 3 questions about the algorithm. Finally, we asked them for their overall impression of the visualizations and suggestions for improvement.

In order to control for the inherent differences in the two sets of questions during pretest and posttest, we had two arrangements of the questions:

- Question set 1 for pretest, question set 2 for posttest, and
- Question set 2 for pretest, question set 1 for posttest

We randomly assigned the participants to one of the two arrangements. People who got assigned to the first arrangement were classified as Group A and those assigned to the second arrangement as Group B. The question sets and their respective pre- and posttest accuracies are shown in the result.

Result Analysis

Understanding of the Algorithm

We received mixed results for the functional test of our design. We saw an improvement in the overall correctness for group B participants and a decrease for group A. The accuracy seemed to correlate strongly with the specific questions asked. Question set 1 had a higher overall accuracy, whether answered during pretest or posttest, than question set 2.

Question Set 1

Question	Pretest Accuracy	Posttest Accuracy
1. What is the random forest algorithm?	1.00	1.00
2. What is the main strength of random forest compared to a single decision tree?	1.00	1.00
3. Why are the decision trees in a random forest different from each other?	0.50	1.00

Question Set 2

Question	Pretest Accuracy	Posttest Accuracy
4. What data and features are used during training of individual decision trees in a random forest?	0.33	0.00
5. In a classification problem, what is the output of each individual decision trees in a random forest?	0.67	0.50
6. In a classification problem, how does the random forest model generate the final prediction outcome based on predictions from individual trees?	0.67	1.00

Note: Accuracy = number of people answered correctly / number of people. The pretest and posttest accuracies are calculated off different groups of participants, as each participant will only answer the same question once, either during pretest or posttest.

Participant Group	A	B
Group Size	2	3
Pretest Questions	Set 1	Set 2
Posttest Questions	Set 2	Set 1
Pretest Average Accuracy *	0.83	0.56
Posttest Average Accuracy *	0.50	1.00
Change in Accuracy **	**-0.33**	**0.44**

* average of percentage correctness of participants in the same group, i.e. sum of score (number of questions answered correctly divided by the total number of questions) for all participants in the group divided by the number of participants in the group
** Posttest average accuracy - pretest average accuracy

Question 4 asked about the training procedure of the algorithm and had the lowest accuracy score. One of our visualizations ("Training of Trees") was designed to demonstrate the concept. We noticed during user testing that people often misunderstood what the visualization was trying to convey. The participants told us that they did not associate the highlighting of table rows and animations of decision trees with the training process. We therefore updated our visualization and text explanation to make the connection more explicit.

Overall Impression and Usability of the Visualization

We also asked the participants to rate, on a scale from 1 (not at all) to 7 (very much), how much they have enjoyed the visualization and how much they felt they have learned. Overall people rated highly on both aspects, and the ratings were similar for participants across Group A and B. Interestingly, even though the results from the above section were mixed, people generally felt that they have learned something from the visualizations.

- Question 7. Did you enjoy reading this?
- Question 8. Did you feel like you learned?

Participant Group	A	B
Average score for Q7	6	6
Average score for Q8	6	6

Note: both questions were likert style questions, with 1 representing not at all and 7 representing very much.

In addition to survey questions, we also interviewed the participants on the overall usability of the visualizations, especially if they found any parts of the visualization confusing. We asked them the following 4 questions, and included below a summary of their responses organized by visualizations.

- Question 9. What's your favorite visualization?
- Question 10. What are some of the confusing things about the visualizations?
- Question 11. Are there any unnecessary or missing parts?
- Question 12. Do you have any other comments?

Visualization: Fallacy of Individual Predictors
The box color prediction example used in the visualization was unrelated to the main mushroom classification problem, and the visualization also left out the punchline about the power of a

crowdsourced algorithm. Some participants found the example disconnected and hard to follow, and felt that the visualization was missing a conclusion. In response to the feedback, we updated the visualization to use the main classification problem as our example, and added in the conclusion.

Visualization: Training of Trees

As discussed earlier, most participants did not realize the animations aimed to describe the training process of the algorithm. One person completely missed the radio buttons used to step through the process. We believed there were 2 contributing factors to the misunderstanding:

1. Confusing visual metaphors, e.g. tree icons representing the decision tree algorithms. Some participants did not recognize the underlying meaning of the icons.
2. Abstract animation, e.g. we used a circle coming out of a table and transitioning into a tree to represent the training process. Some participants were unclear what the circle meant, or what the animation signified.

In response to the feedback, we updated our visualization to clarify the outcome of each step, e.g. for random sampling of data step we masked the unselected data to further highlight the selection process. We added additional description for each step to connect the visual representation with the underlying concept. Additionally, we changed the step through control from radio buttons to regular buttons to better fit people’s mental models.

Visualization: Training Result

Participants for the large part recognized each tree represented a separate prediction model. Some of them had confusions about which part of the visualization was clickable, and some thought the differences in the tree icons carried semantic meaning with them. In response to the feedback, we added a legend to the visualization with additional explanation.

Visualization: Making a Prediction

Participants really enjoyed the interactivity of this visualization, and several of them replayed it multiple times. When asked, some participants mentioned they did not realize the final prediction outcome was decided using majority vote, a mechanism implicitly demonstrated in the visualization. In response to that feedback, we made the voting process more explicit by annotating the circle coming out of each tree as their vote for the final decision. Also, we increased the number of trees to more closely match the numbers used in default implementations of the algorithm.

Overall Feedback

In terms of the visualization as whole, a few participants noted the significant amount of text on the site, and that some of the words are relatively technical. We considered both feedback, and decided both were necessary in order to help people get acquainted with both the model and vocabulary. The text provides much needed context and explanations for the visualizations, and helps connect different components into one logically coherent piece. We did make an extra effort to highlight technical words and ensure they are defined the first time they are used.

Link to Code and Visualization

- Link to our visualization: A Path to Random Forest
- GitHub Repository for our source code

Division of Labor

Below table shows a breakdown of the work done by each of the team members with approximate contributions in percentages.

Project Component	Sub Component	Yiyi	Sam
Data Preparation	Data Sourcing	0%	100%
	Data Preprocessing	100%	0%
	Random Forest Model Running	100%	0%
Visualizations	1. Fallacy of Individual Predictors	100%	0%
	2. Training of Trees	0%	100%
	3. Training Result	100%	0%
	4. Making a New Prediction	0%	100%
Design	Website Text Writeup	30%	70%
	Website Layout Design (title and conclusion)	70%	30%
User Testing and Others	User Testing	30%	70%
	Report Writing	50%	50%

Appendix

Code Reference

- Collapsible trees (v4)
- Decision tree with color and width of the path (v3)
- Day / Hour Heatmap for blocks
- Susie Lu's D3 Annotation Library
- HTML table generation

Information on random forest

- Medium post: The unreasonable effectiveness of random forests

Random Forest User Testing Questions

- What is the random forest algorithm?
  a. A reinforcement learning that grows and prunes a decision tree iteratively
  b. A process to generate new features from existing features
  c. A regularization method that selects the most important features from a dataset
  d. An ensemble method that combines the power of individual decision trees

- What is the main strength of random forest compared to a single decision tree?
  a. It is more robust against errors and noise in data
  b. It is able to fit to the training data better
  c. It runs faster than a single decision tree
  d. It allows all data to be used for training

- Why are the decision trees in a random forest different from each other?
  a. They are initialized randomly
  b. They are trained on different data
  c. They have different preset structures, i.s. some are narrow and deep while others are fat and shallow
  d. They are not different

- What data and features are used during training of individual decision trees in a random forest?
  a. All the data with all features
  b. A random sample of data with all features
  c. All the data with a random subset of features
  d. A random sample of data with a random subset of features

- In a classification problem, what is the output of each individual decision trees in a random forest?
    a. The most important feature for the classification task
    b. The weights of each features
    c. Prediction of class for each input
    d. Probability distribution of all candidate categories

- In a classification problem, how does the random forest model generate the final prediction outcome based on predictions from individual trees?
    a. Take the prediction of the tree with the highest accuracy during the training phase
    b. Majority vote
    c. Based on a preset threshold value
    d. Randomly

- Did you enjoy reading this? (1 - not at all, 7 - very much)
- Did you feel like you learned? (1 - not at all, 7 - very much)
- What’s your favorite visualization?
- What are some of the confusing things about the visualizations?
- Are there any unnecessary or missing parts?
- Do you have any other comments?