They Shoot Pictures, Don’t They?

A decision tree to navigate the 1000 greatest films of all-time

Justin Kraus - DVIA - Fall 2020

Final Visualization Visible here:

https://justinkraus.github.io/TSPDT/

Initial Dataset

- “They Shoot Pictures Don’t They” is a list of the top 1000 movies of all time aggregated from over 11,000 individual lists.
- The raw dataset has qualitative and quantitative information about each movie including:
  - Country of origin
  - Year the film was made
  - Film length
  - Director
  - Genre
  - Film’s ranking within the list.

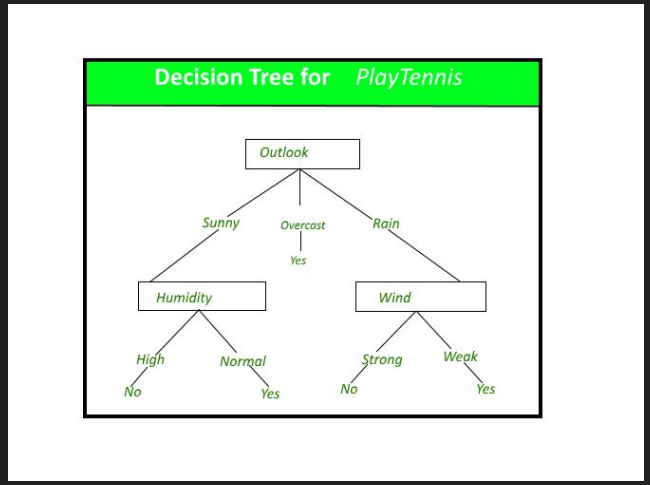
Creating the Hierarchy

- Using the qualitative and quantitative information, I classified these items into categories to create a hierarchy.
- During this process, consideration was given to determine how many items would be assigned to each category in order to create an even distribution.
- **Defining the user** - the final categories are structured around considerations a person would make when choosing to watch a movie:

Language	Length of Film	Film Era	Genre
• English • Foreign	• Long • Medium • Short	• Pre-1950 • 1950-1990 • 1990-Now	• Many

Visualization Form

- D3 has many options for visualizing a hierarchy
- I chose the tree layout to create a decision tree that addresses uncertainty in movie watching.
- Other hierarchical forms emphasize depictions of quantities within the hierarchy (for example, how much one node contributes to the whole) but as the visualization method here served to propose questions to lead to a movie the tree layout was appropriate.
- Navigation of the tree is done through a series of questions which acts as a filter to remove titles that user would not be interested in seeing.
- Instead of more traditional button filters, the tree is meant it enable exploration by allowing users to visualize other possible paths if they don't like the final selection.



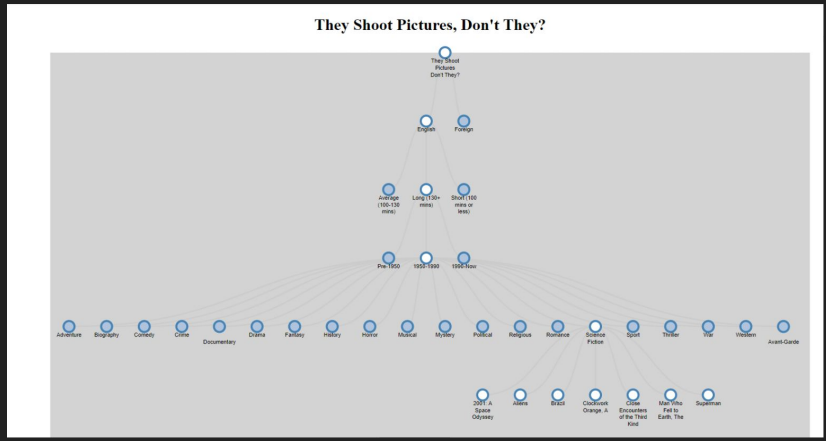
Decision Tree to determine if it is suitable for playing tennis

Early Iteration - Structure and Fit

The initial iterations required organizing the data in a format that could be used by the d3.tree method and ensuring basic functionality worked correctly. While the primary visualization structure followed a similar form that is used in the final, a number of items were worked through that contribute to the overall functionality of the visualization:

- **Genre organization**
  - The initial iteration had multiple genre and subgenre levels which led to a more specific final movie selection.
  - From a user-experience perspective I decided this wasn't necessary as it created too many decisions before arriving at the goal (selection of movie)

- **Text wrapping**
  - Wrapping of the node labels is functionality not native to D3. To avoid node labels from overlapping, additional functionality needed to be implemented as a wrap function.
  - D3 tree visualizations often depict a horizontal expansion as opposed to the vertical orientation I used here which avoids this issue entirely.

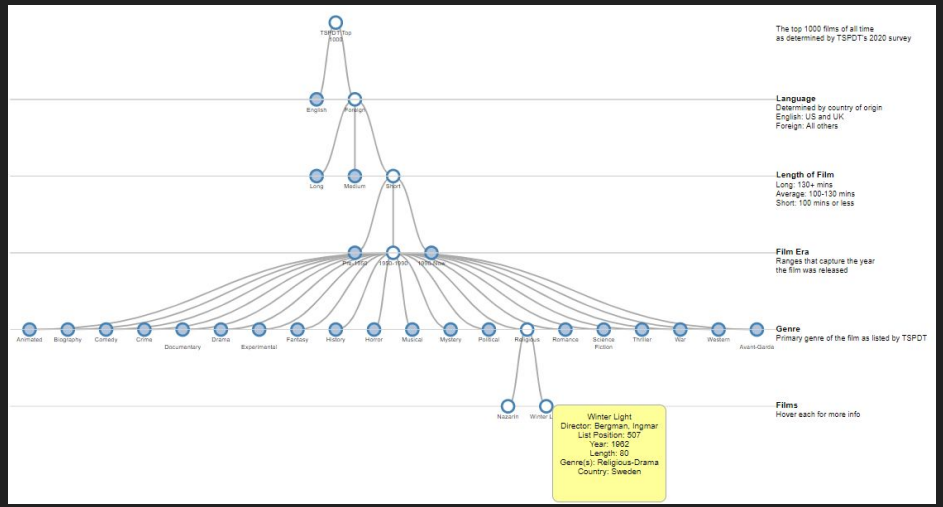


Early Iteration

Middle Iterations - Explanatory Text

- **Margin Labels**
  - The margin labels here inform the user of what each level is based on and how they are organized. Each has a transition that is only visible what that level is displayed
  - I received feedback during the final prototype review that said these margin labels weren't immediately visible. To address this I added the horizontal lines to guide the viewer to them.

- **Tooltip**
  - Each node has a hover text informing the user of what is represented by that node.
  - This is most informative for the final movie list layer where all the information about the movie is given.

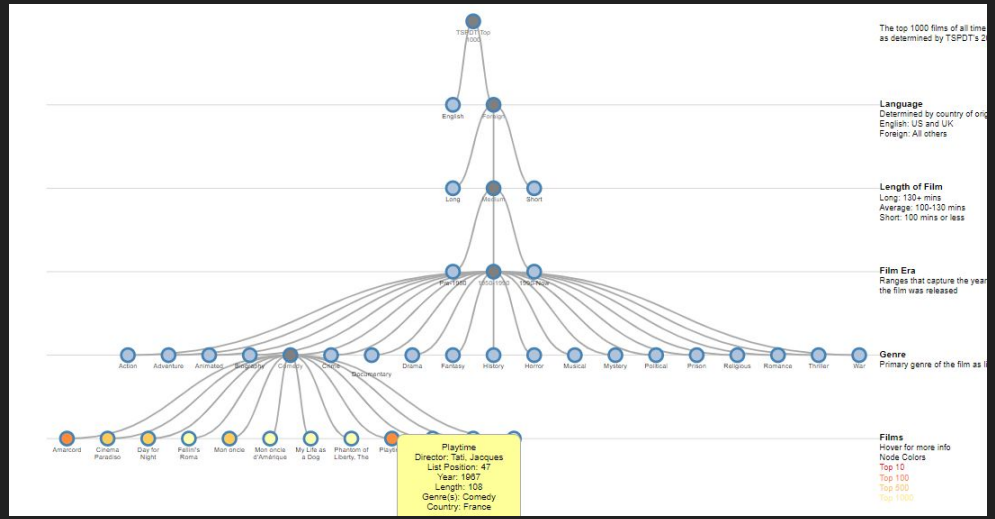


Closer to Final Iteration

Final Iteration - Color

- **Node Colors by List Position**
  - Individual movie nodes colors are determined by their ranking within the overall list.
  - Red/orange represents higher in the list, yellow represents lower.

- **Node Hover Selection**
  - Hovering on a node will highlight the parents of that node in grey to assist with orientation within the tree chart



Close to Final Iteration

Final Visualization Visible Here:

https://justinkraus.github.io/TSPDT/