How to use VisuaLearning: Bayes Net Debugger

**Purpose**

The purpose of this application is to allow beginner machine-learning programmers to better understand and debug their Bayes net machine learning models. The application helps programmers visualize their model in the form of a direct acyclic graph. In addition, the user is able to visualize how data flows through the model and the model makes classifications.

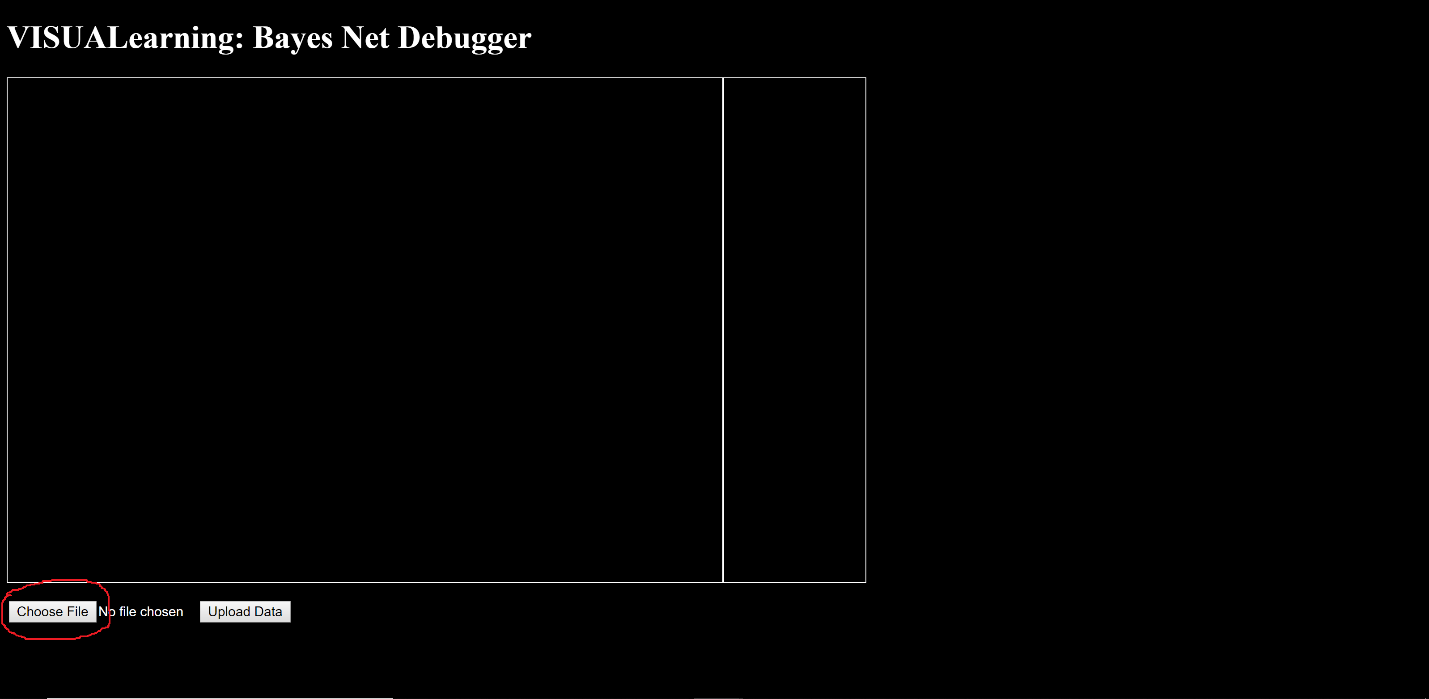
**Scenario**

Alice is a freshman at MIT who needs extra tutoring for 8.01 (physics). However, the only time the tutor is available to meet is at 8 in the morning on Sundays. Attending sessions at 8 AM may be tough for Alice for several reasons: she often goes out partying Saturday nights and thus goes to bed really late, she is not a morning person and is often too lazy to get out of bed at an early time, she is taking a rigorous schedule at MIT so sometimes she would rather do other work than attend tutoring sessions. Our job is to help the tutor figure out the probability that Alice will attend the tutoring session because the tutor does not want to get out of bed at 8 AM if she doesn’t have to. We can figure this out by looking at how Alice behaved in past situations and use conditional probabilities based on other events to calculate the probability that Allice will attend the next session.

The user is provided with two data files containing training data and testing data. The training data contains information about what Alice did in the past and whether or not she attending the tutoring session; it is used to compute probability distributions that will allow the model to make predictions. The testing data is used to determine if the built Bayes net model makes predictions accurately. It also contains information about how Alice behaved in the past and whether or not she attending the tutoring session.

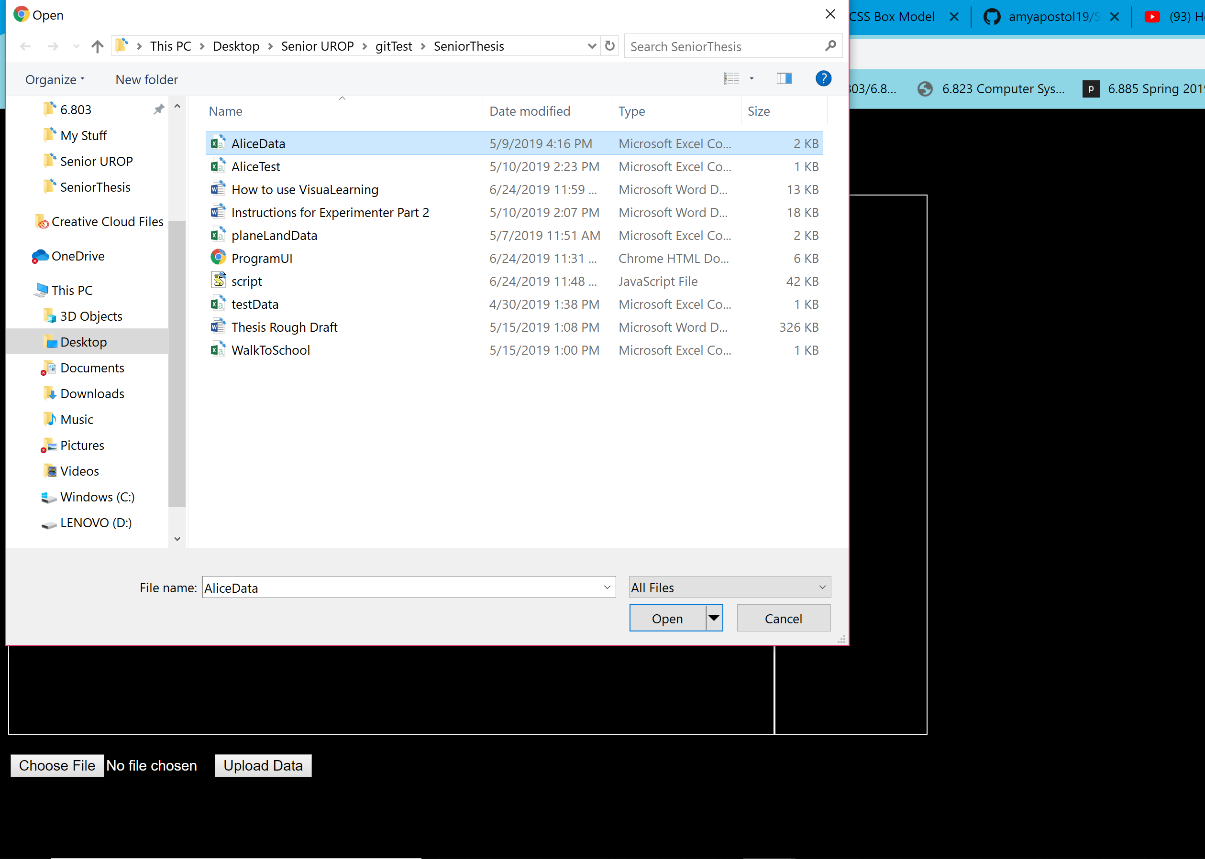
**Step 1: Upload the Data**

* The initial screen when first opening the program looks like this:

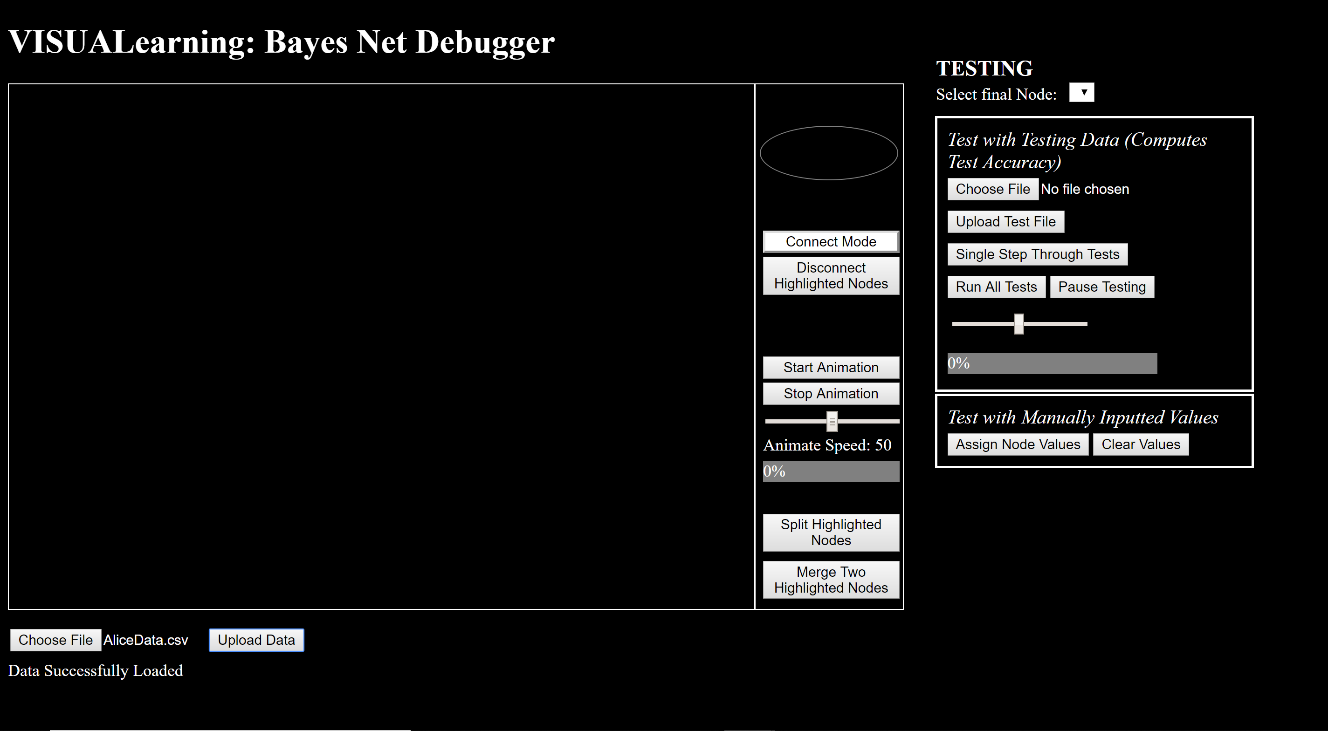


Select the button that says “Choose File” to select training data.

* Select the file “AliceData.csv”. All data files are in the form of a CSV file.

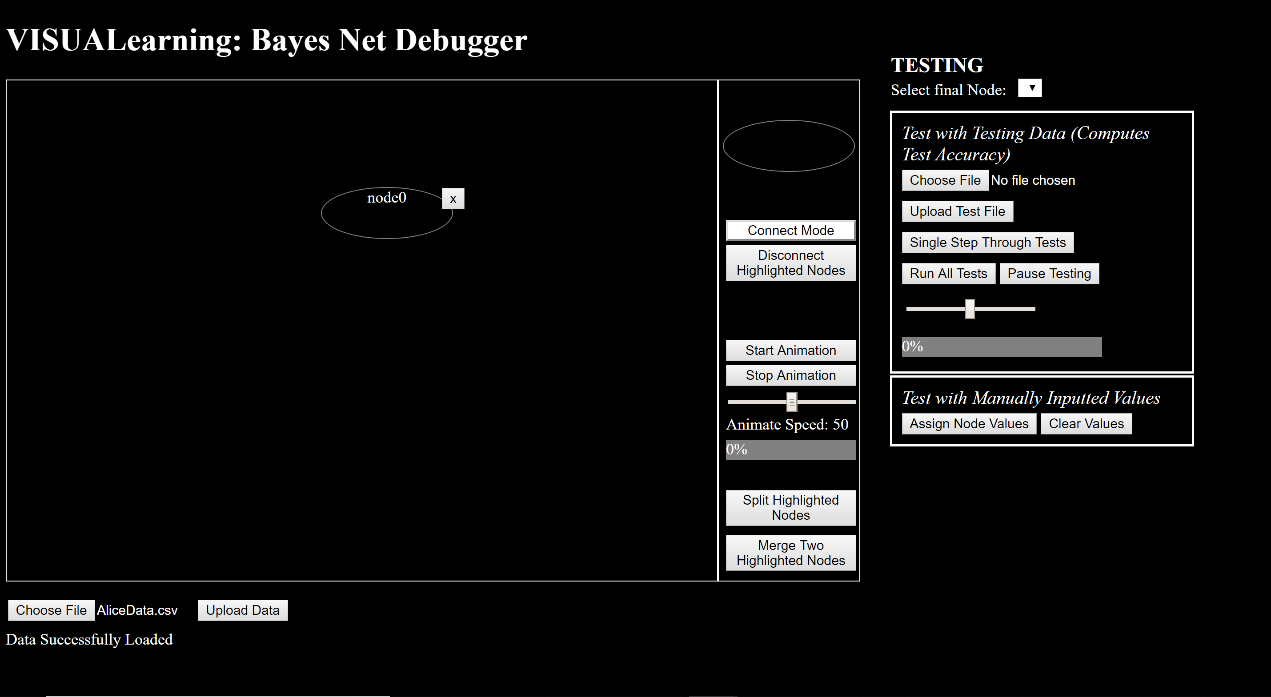


* After the training data file has been selected, click the button that says “Upload Data”. This will bring the user to the main screen that allows the user to build and test their machine learning model (shown below).

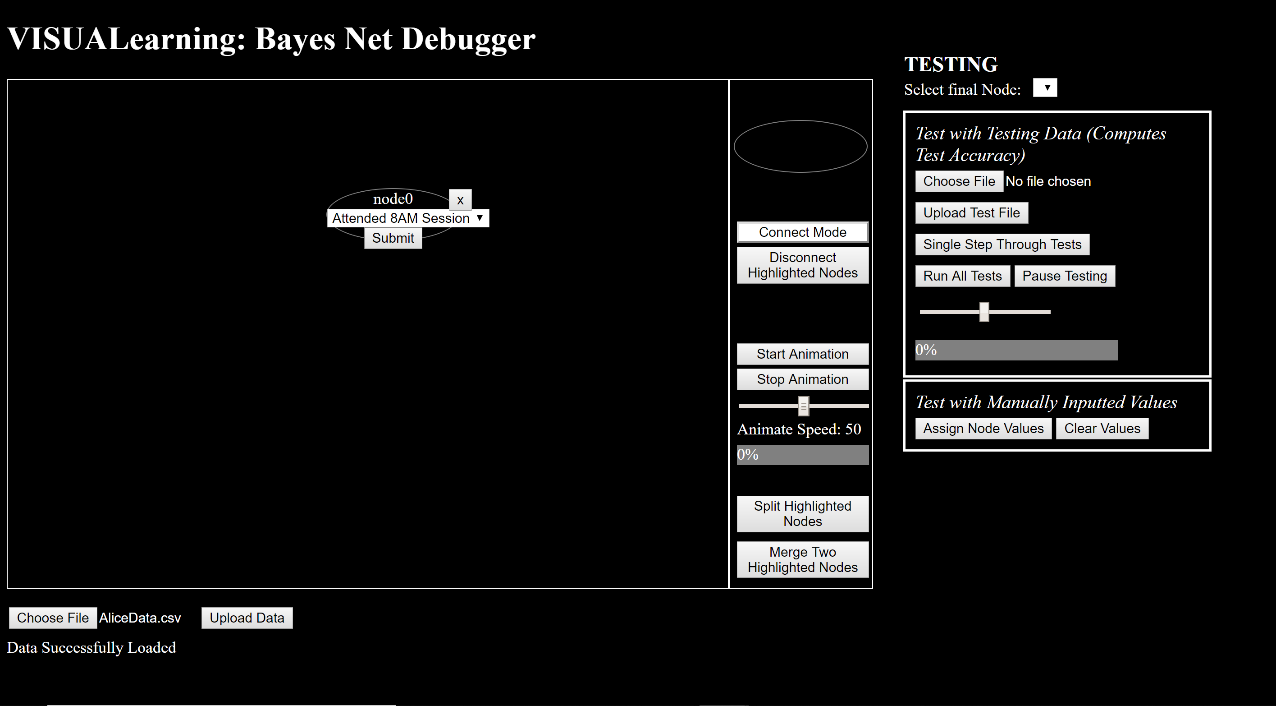


**Step 2: Creating Nodes and Naming Nodes**

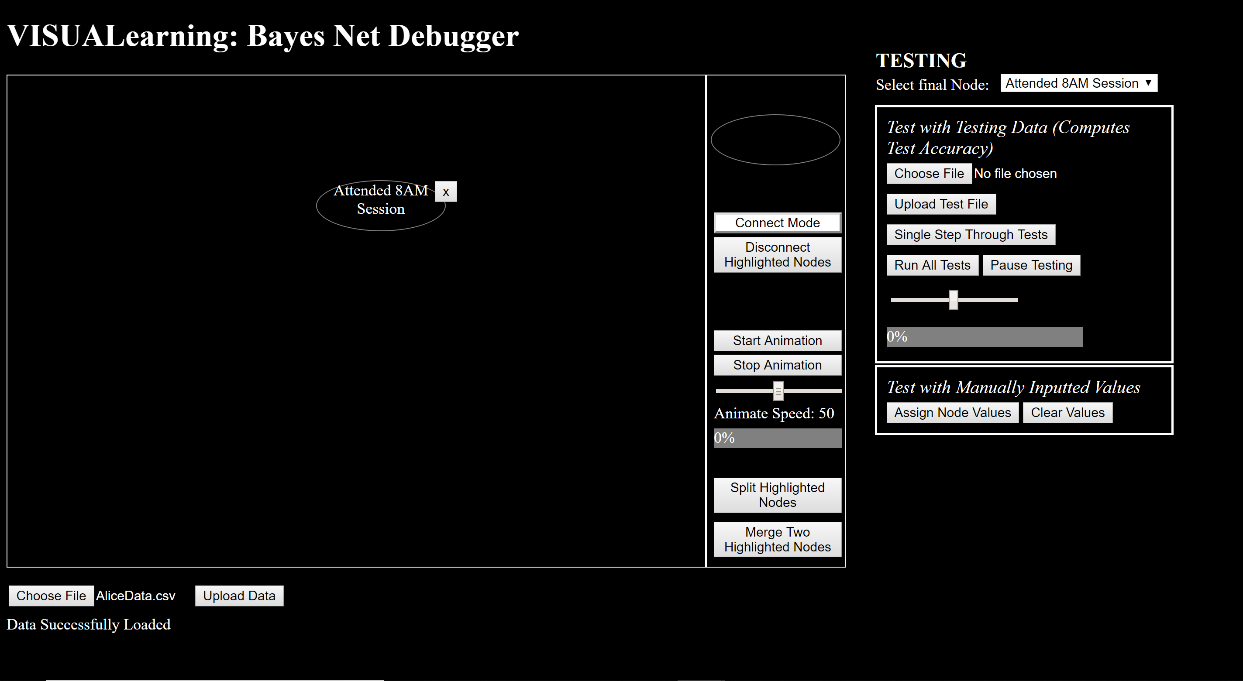
* Create a new node by clicking and dragging on the oval shape in the tools dock.



* Rename the node by double clicking on the node.



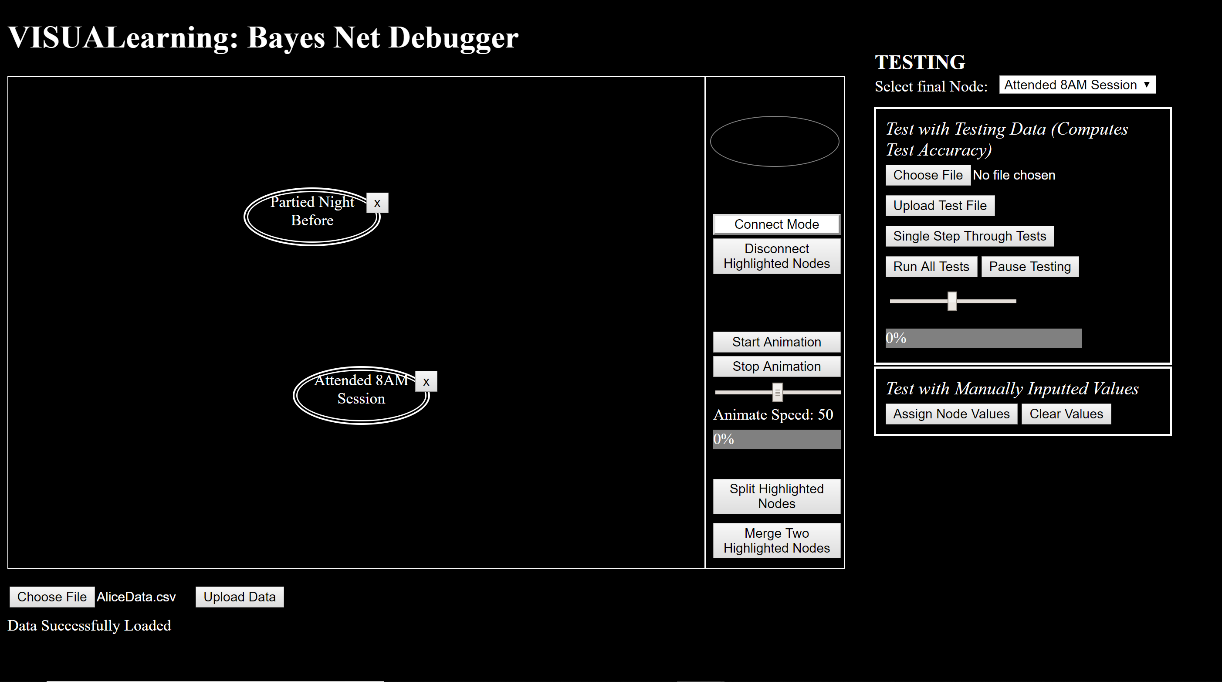
Select the feature name from the dropdown form and click “Submit”. The node should appear like this:



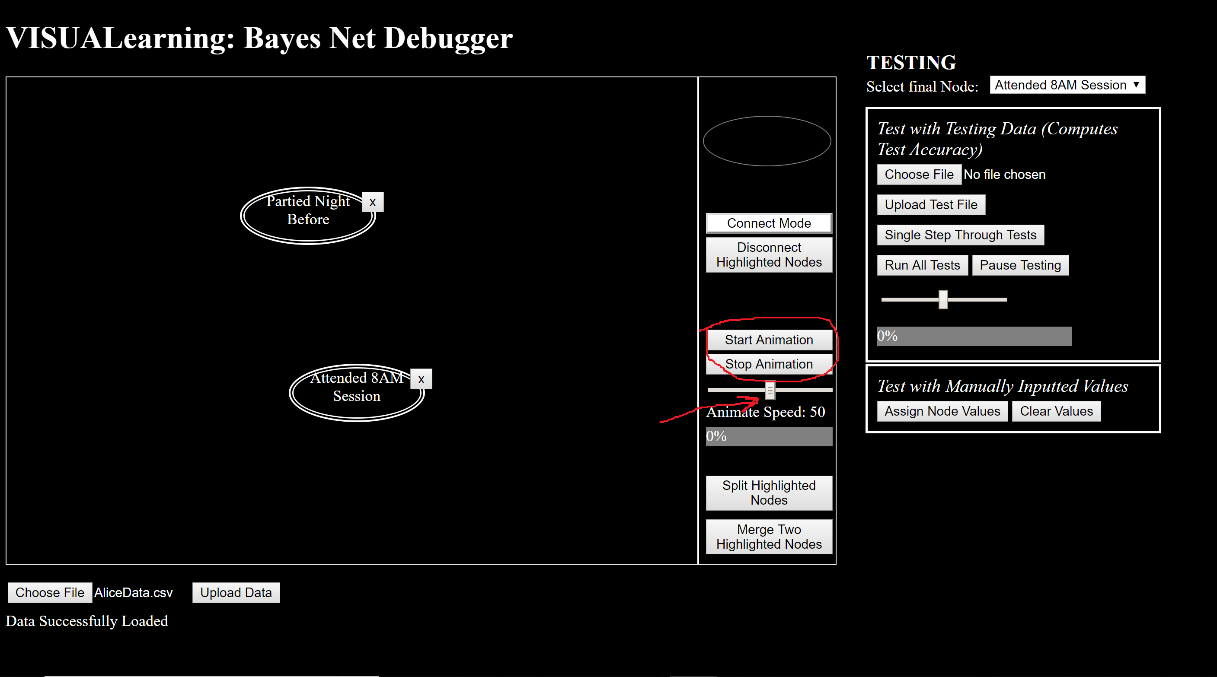
**Step 3: Animating Nodes**

Animating nodes helps the user visualize the data. During animation, the user is able to see how often certain features are true at the same time. Visualizing the correlation between features thus allows the user to make conjectures about whether or not there is causation between two features.

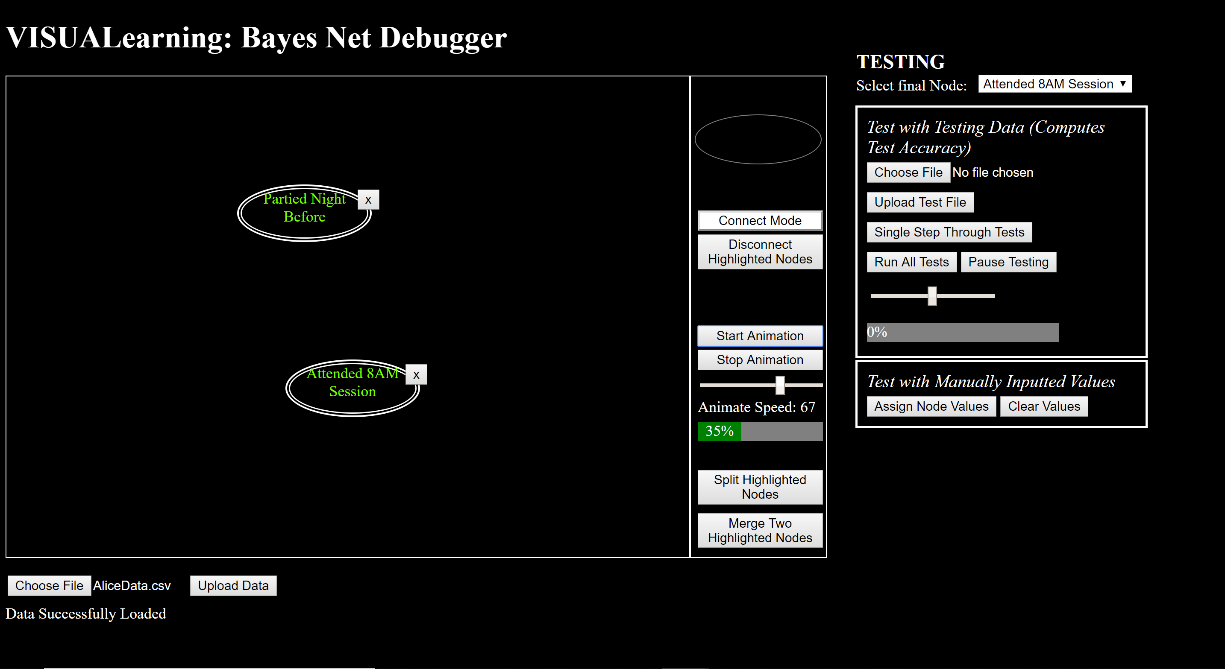
* Highlight two nodes that may be correlated to one another by Ctrl+click on each node.



* All highlighted nodes will animate. Start and Stop Animation by clicking the “Start Animation” and “Stop Animation” buttons in the tool dock. Change the speed of the animation by sliding the slider.

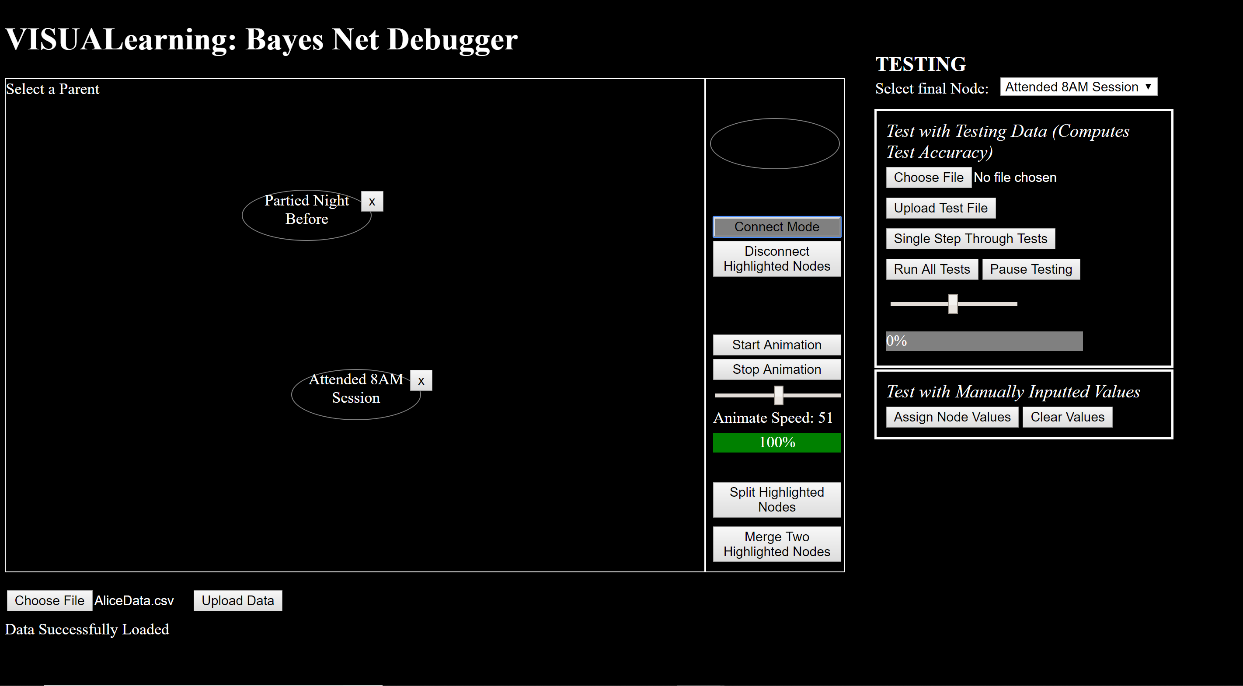


* When two features are true at the same time they will both flash a green color. Progress for the animation is displayed by the green bar underneath “Animate Speed”.

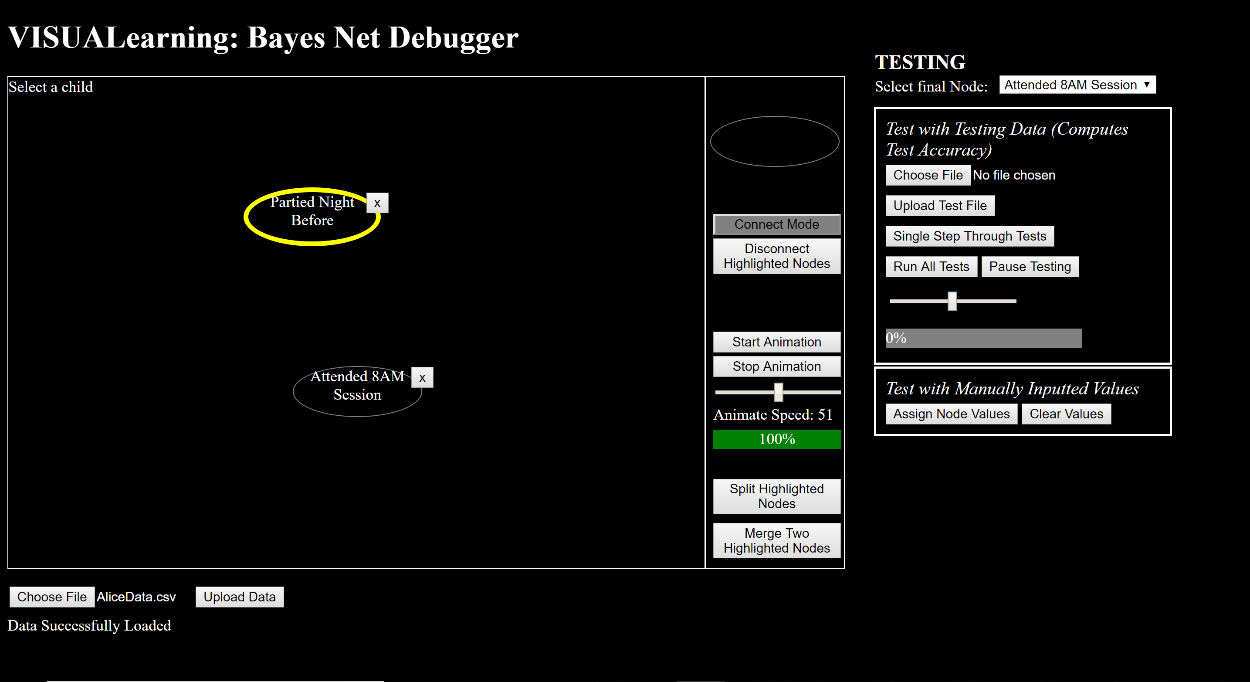


**Step 4: Connecting and Disconnecting Nodes**

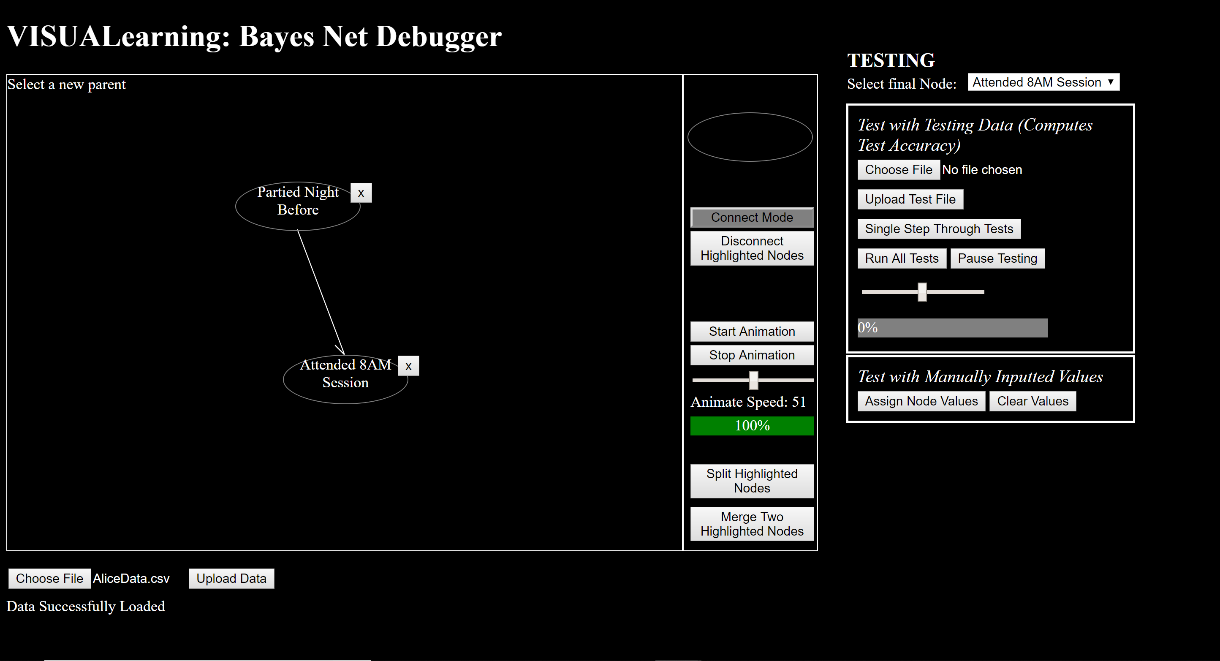
* To connect two nodes, click the button that says “Connect Mode”. No nodes can be highlighted in order to enter connect mode.



* Once in connect mode, click the node that is supposed to be the parent. This highlights the node yellow.

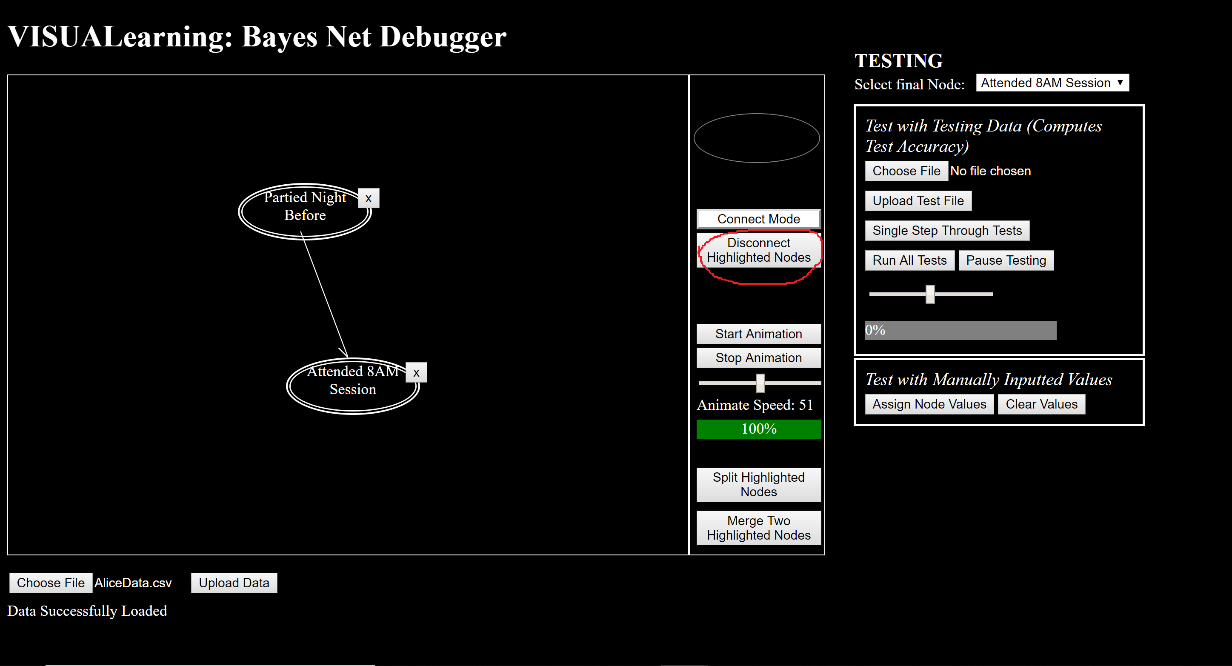


* Once the parent has been selected, click on the child node. This draws an arrow between the parent and the child.



Exit connect mode when finishing connecting nodes.

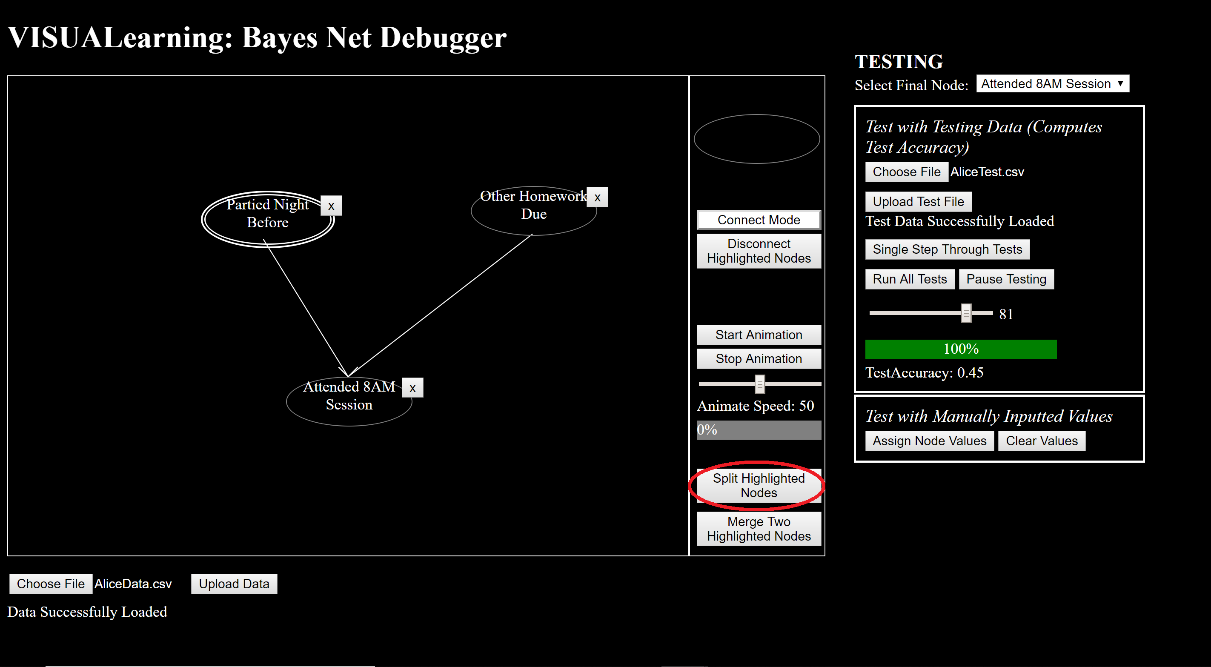
* To disconnect nodes, highlight the two nodes you want to disconnect and select the “Disconnect Highlighted Nodes” button.

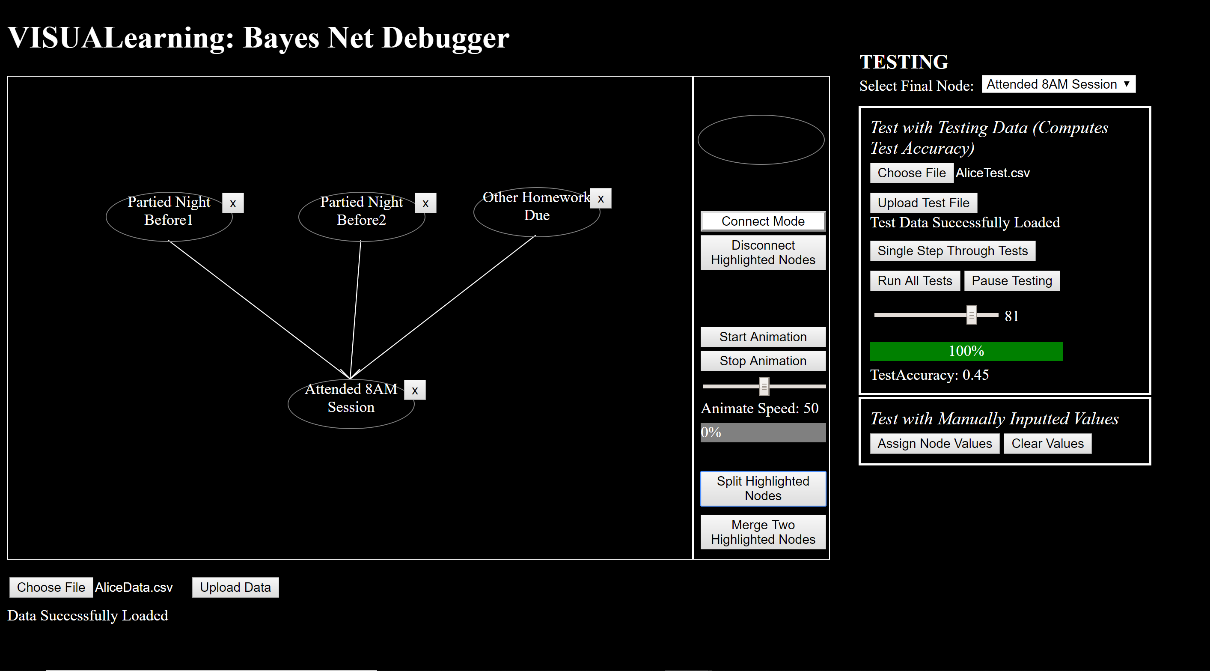


**Step 5: Splitting and Merging Nodes**

Sometimes a model performs better if we split a single node into two separate nodes. This makes the model more specific. For example, we can split “Partied Night Before” into two separate categories: whether Alice partied and was drinking, or whether she partied and was not drinking.

* To split a node, highlight the specific node you want to split. Then click the button “Split Highlighted Nodes”.

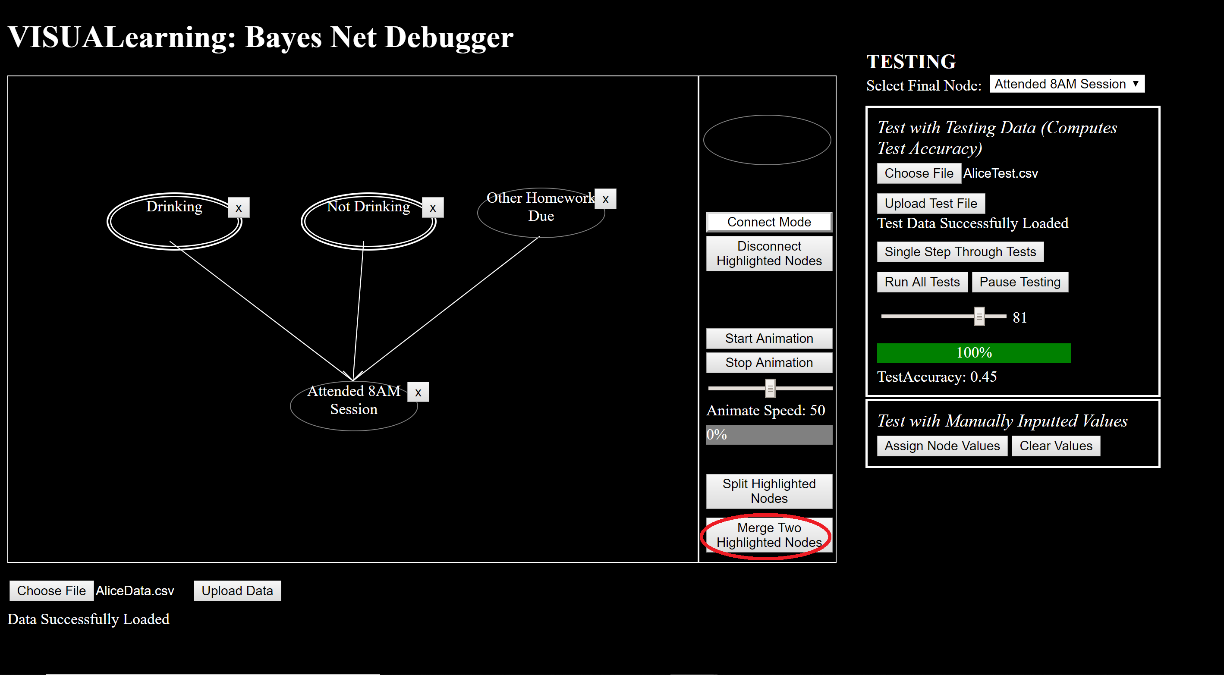




Remember to rename the split nodes to existing feature names.

Sometimes splitting nodes causes the model to be too specific and results in overfitting. Thus we also have the ability to merge nodes. For example, maybe splitting “Partied Night Before” doesn’t actually help improve model accuracy.

* To merge nodes, highlight the **two** nodes you would like to merge. Then click the button “Merge Two Highlighted Nodes”.

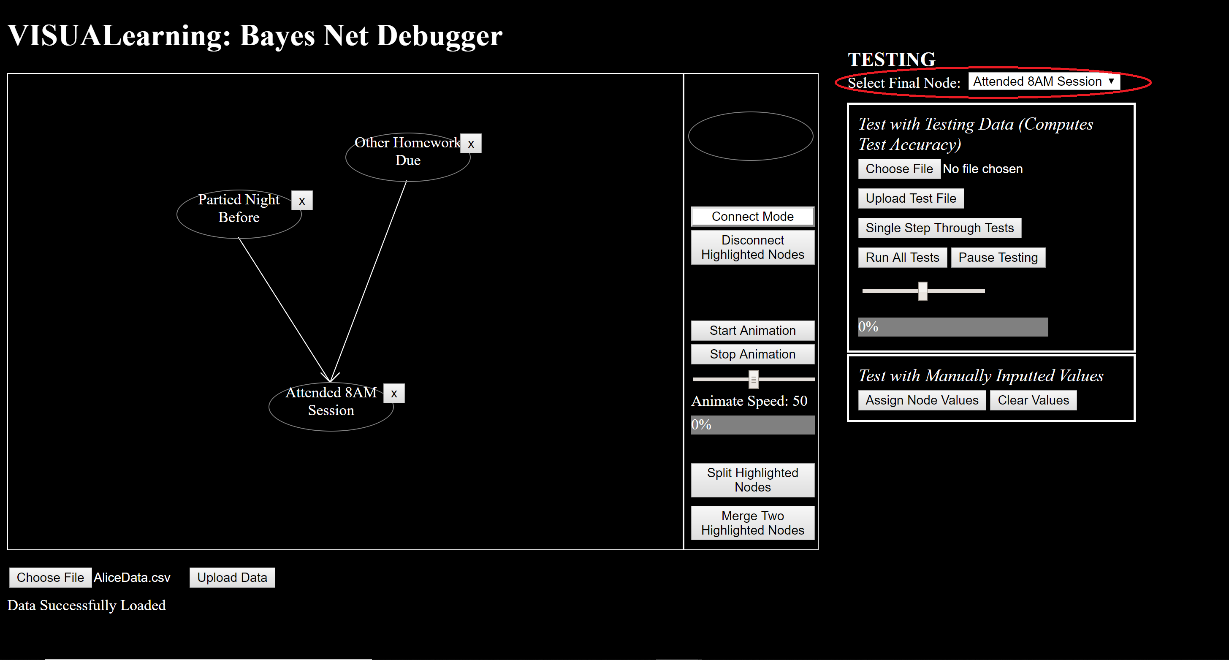


Remember to rename the merged node to an existing feature name.

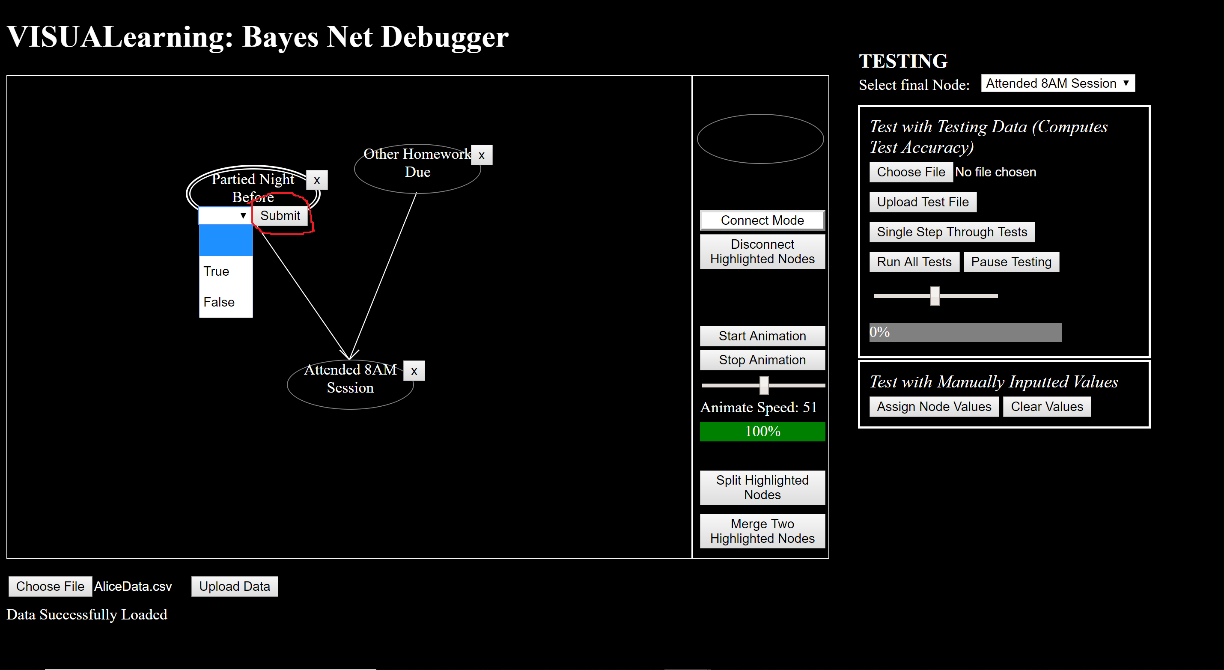
**Step 6: Understanding How the Bayes Net Model Makes Predictions**

We now have all the tools necessary to build a Bayes net machine learning model. Once an optimal model is built we can begin testing the model and examining how it works.

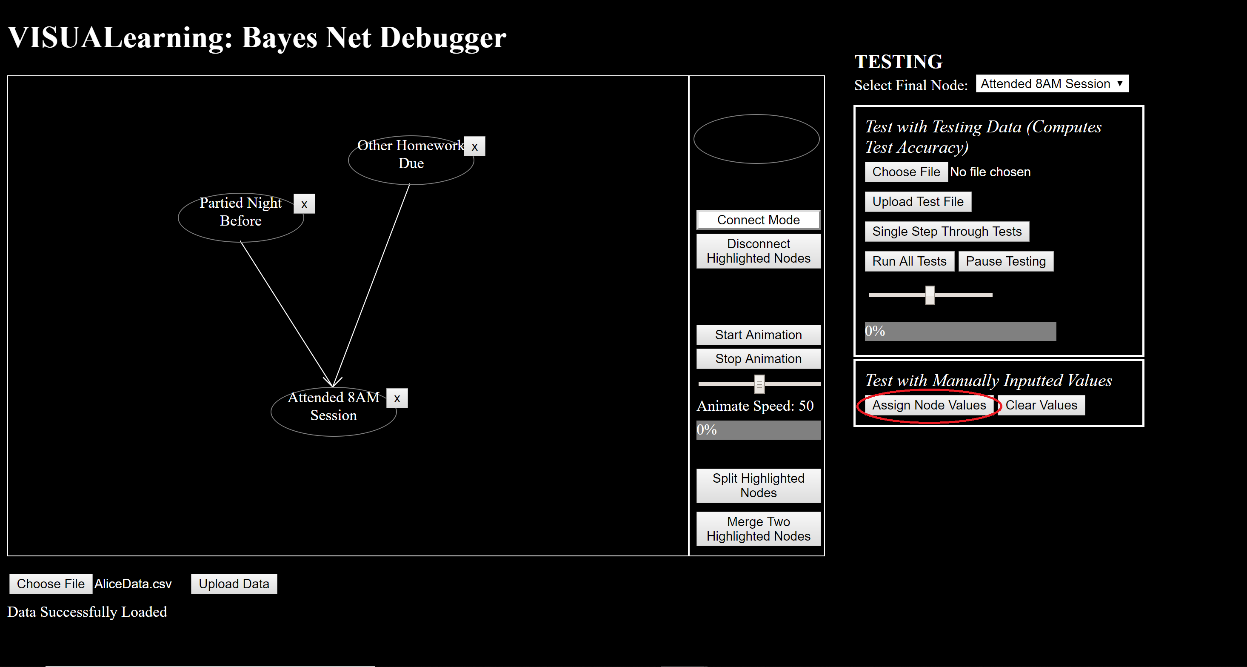
* First the user must specify the final node in the Bayes net model by selecting the feature “Attended 8AM Session” from the dropdown menu next to the label “Select Final Node”. This informs the program that the final node calculation depends on values from all other features, and thus the final node should be calculated last. The final node is ultimately the value that we want a prediction for.



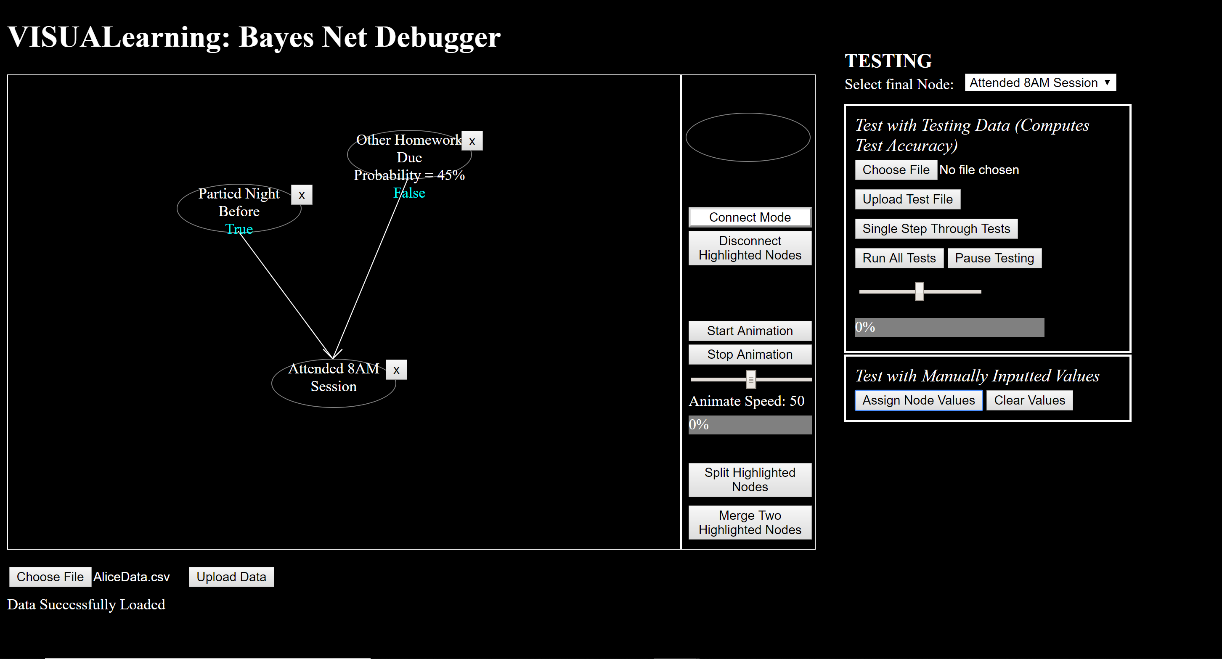
* The user can look at how data flows through the model and how predictions are made by manually assigning values to specific nodes. To assign a node a value, highlight the node, double click the highlighted node, and select a value from the dropdown menu. Click the “Submit” button to assign that node the specified value.

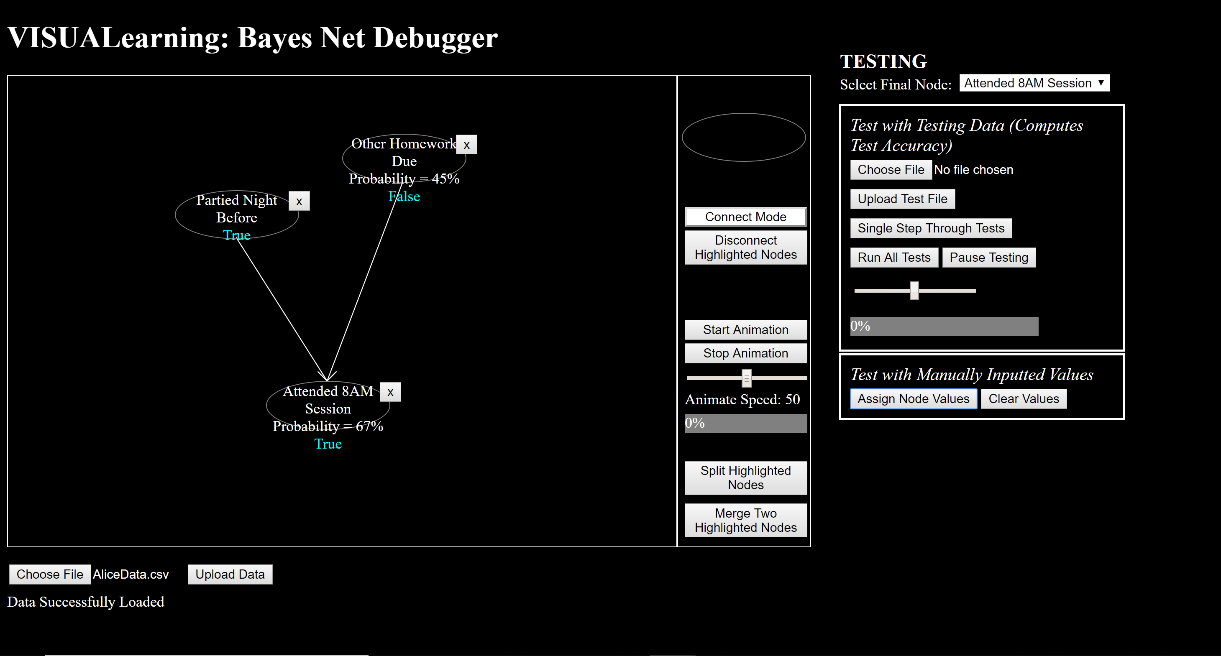


* Once finished manually assigning node values, click the “Assign Node Values” button to automatically assign values to unassigned nodes.



The program assigns values to nodes by randomly selecting a value based on the probability that the node’s feature is true.



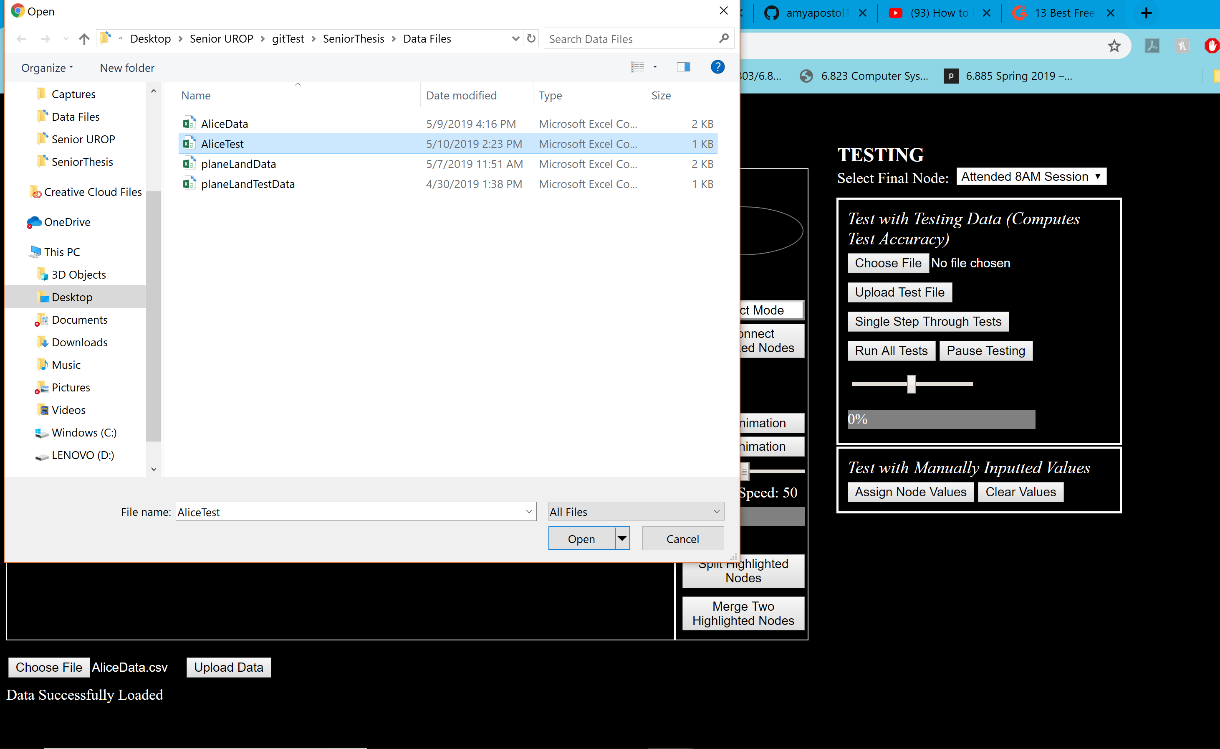


* Select the “Clear Values” button to remove the set values from all nodes.

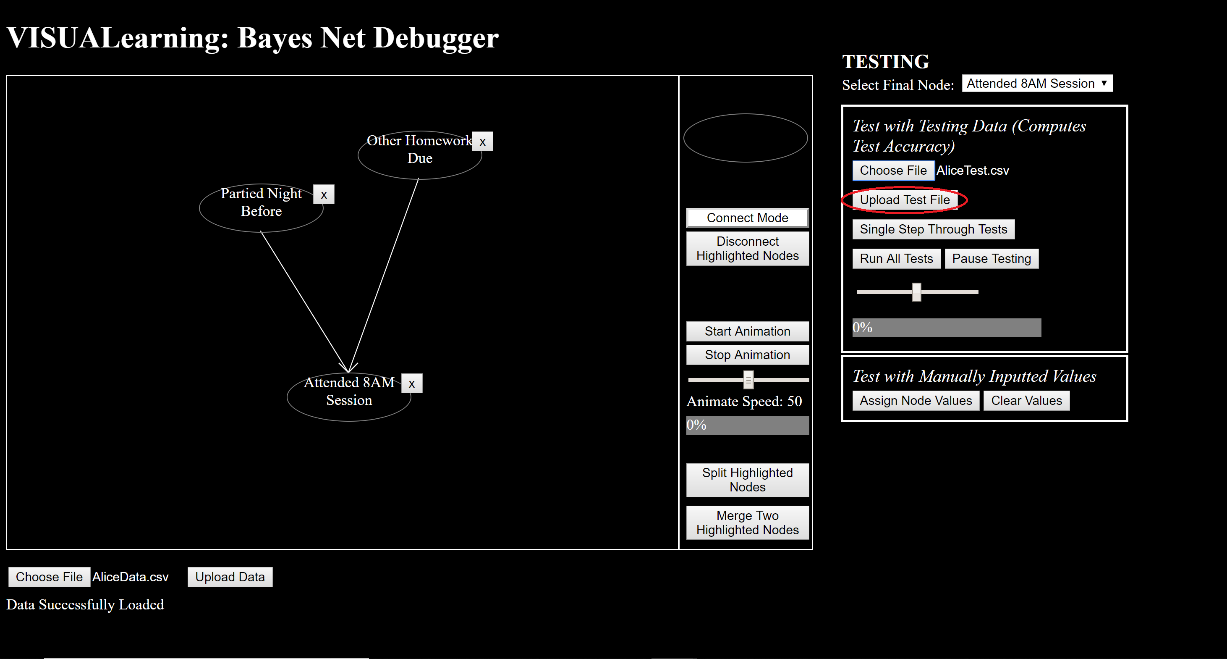
Use the “Testing with Manually Inputted Values” module to see how the model behaves with different inputs.

**Section 7: Testing the Model with Testing Data**

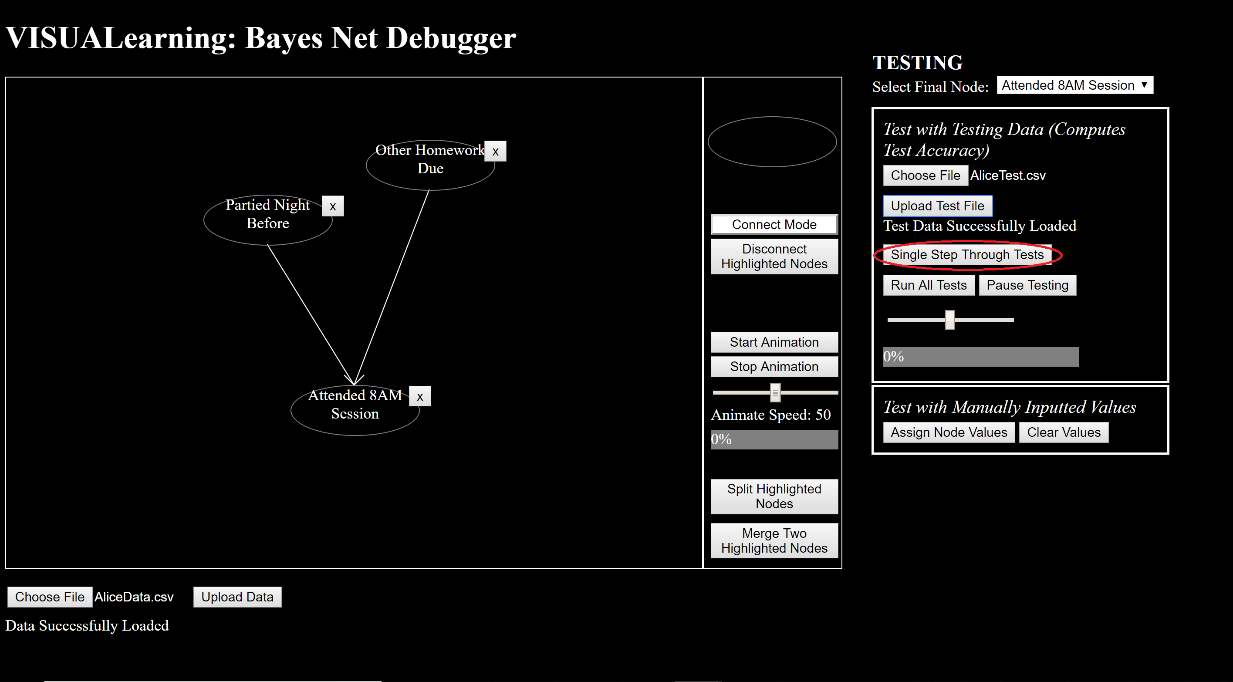
* Move to the “Test with Testing Data” section. Upload the testing data, “AliceTest.csv” by clicking the “Choose File” button.



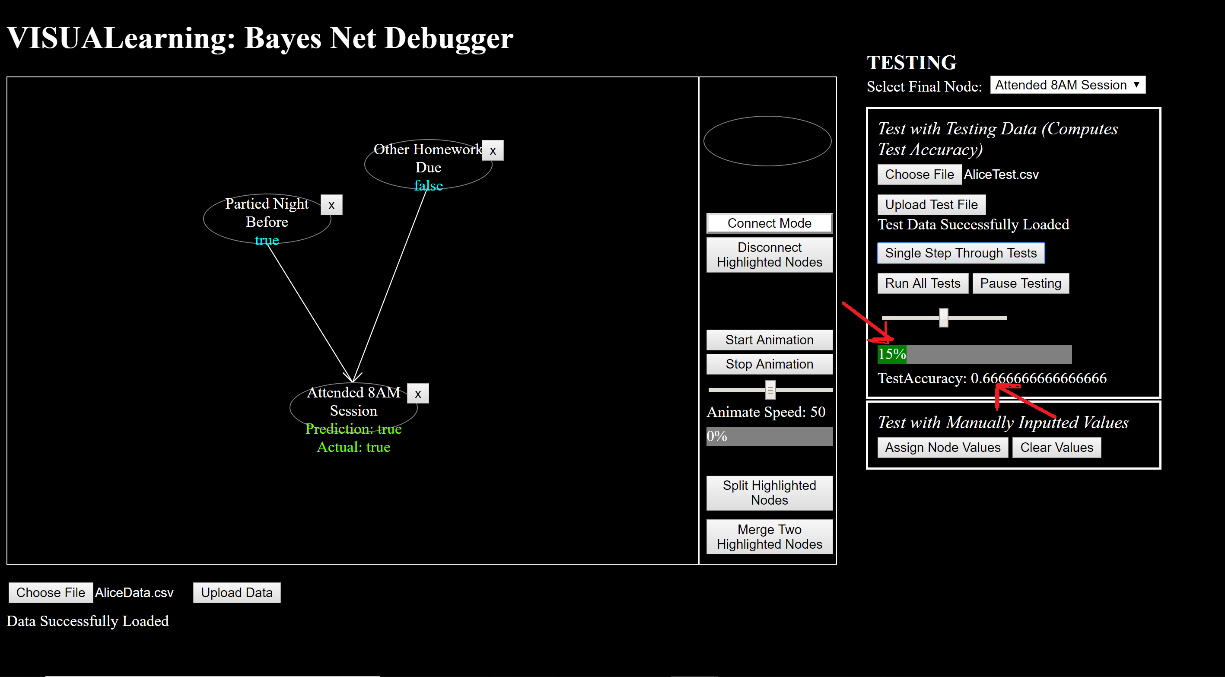
Then click “Upload Test File”.



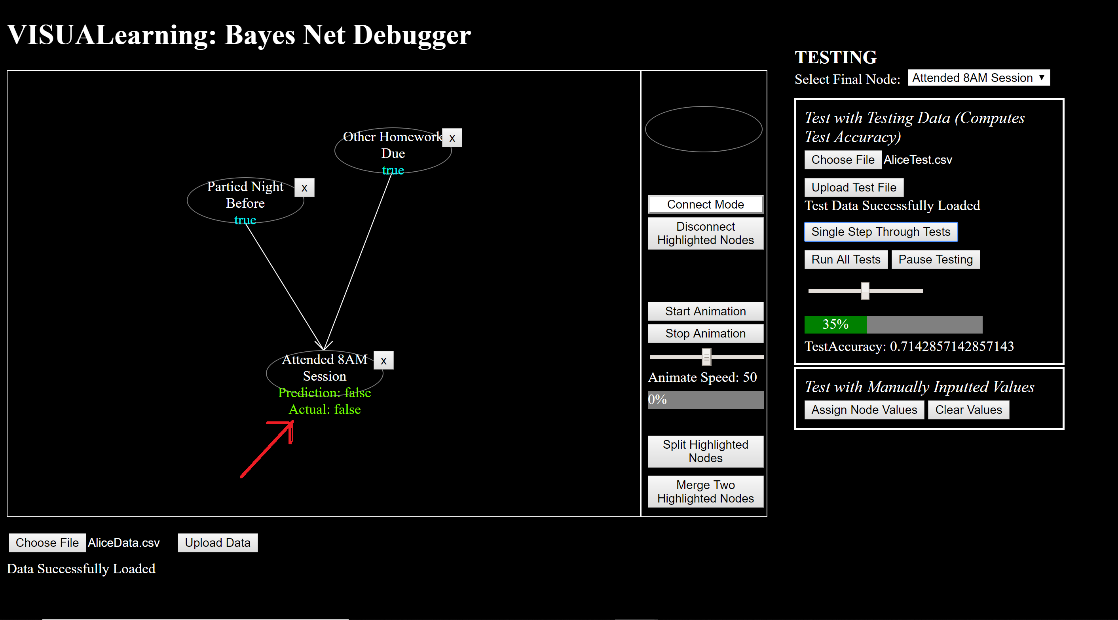
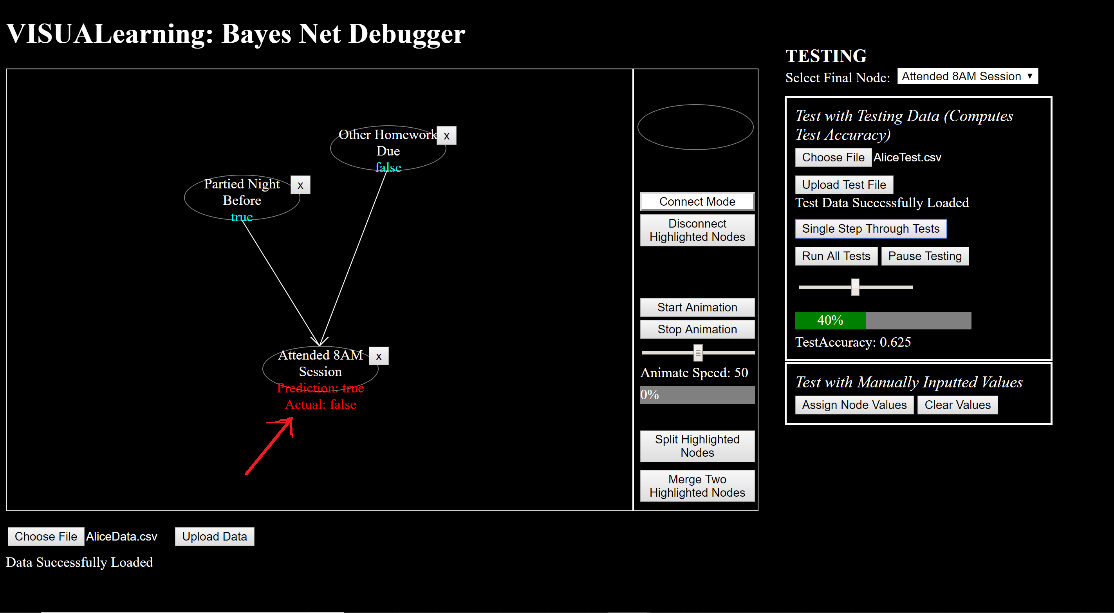
* The test file contains pre-assigned values for each node as well as what the outcome should be for the final node. It measures the model’s prediction for a feature against what actually occurred to determine the testing accuracy (how often the model makes correct predictions).
* The user has the option to look at each individual test, or automatically run through all tests at once. To view an individual test, click the button “Single Step Through Tests”.



The progress (number of tests completed) is displayed by the green bar. The testing accuracy value is displayed below the green bar.



* Correct predictions are displayed on the final node in green, while incorrect predictions are displayed on the final node in red.

* To run all tests at once, click the button “Run All Tests”. You can select the speed at which tests are run by moving the slider. To pause testing, click the “Pause Testing” button.

