Diego Betanzos
Mariana Pérez

Bayes Network Report

The Bayes network designed by the team is based on the recent events that caused controversy with the security of the planes Boeing 737 MAX 8, two less than one year old planes have crashed recently, in a span of less than 5 months, both killing all the passengers and crew members.

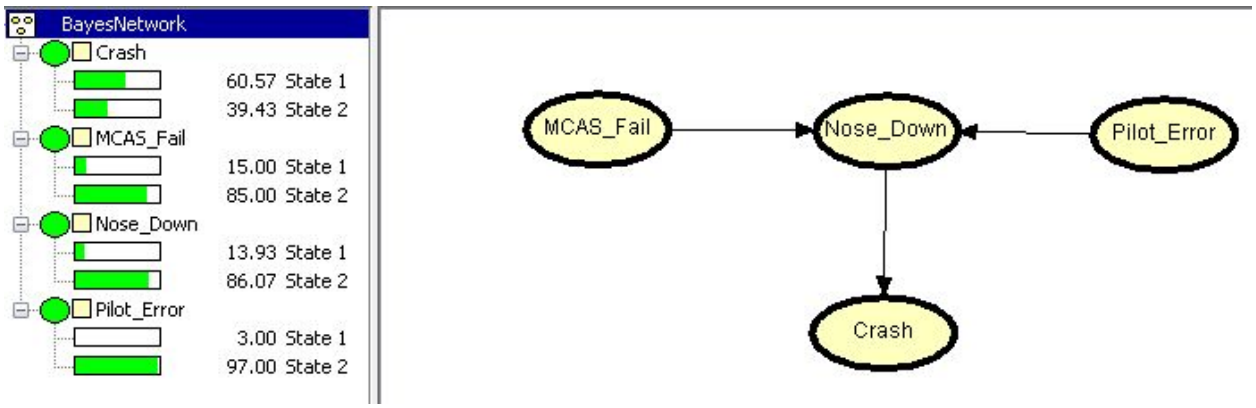
The first crash was determined to be caused by a failure in the MCAS system and failure of the pilots to detect the bad readings, this system is in charge of preventing a “stall” by changing the angle of attack of the plane and making its nose go down, but it depends on a sensor that can fail in certain circumstances, making the plane’s nose go down more than it should and crashing. The second crash is still in investigation but the early results from the investigation point to the same failure with the MCAS system.

Currently, all the 737 MAX in the world are grounded (parked in an airport without flying) until the aviation regulatory associations determined the plane is again safe to fly commercial flights, and one airline already canceled the order of 49 more planes, since this model lost the customer’s thrust to fly in it.

For more information visit:

- https://www.theverge.com/2019/3/22/18275736/boeing-737-max-plane-crashes-grounded-problems-info-details-explained-reasons
- https://www.cnbc.com/2019/03/22/indonesias-garuda-canceling-its-order-for-49-boeing-737-max-jets.html

Based on the information above, the team members decided to create the following Bayes network (Disclaimer: the probability values were arbitrary decided by the team, they are not real):

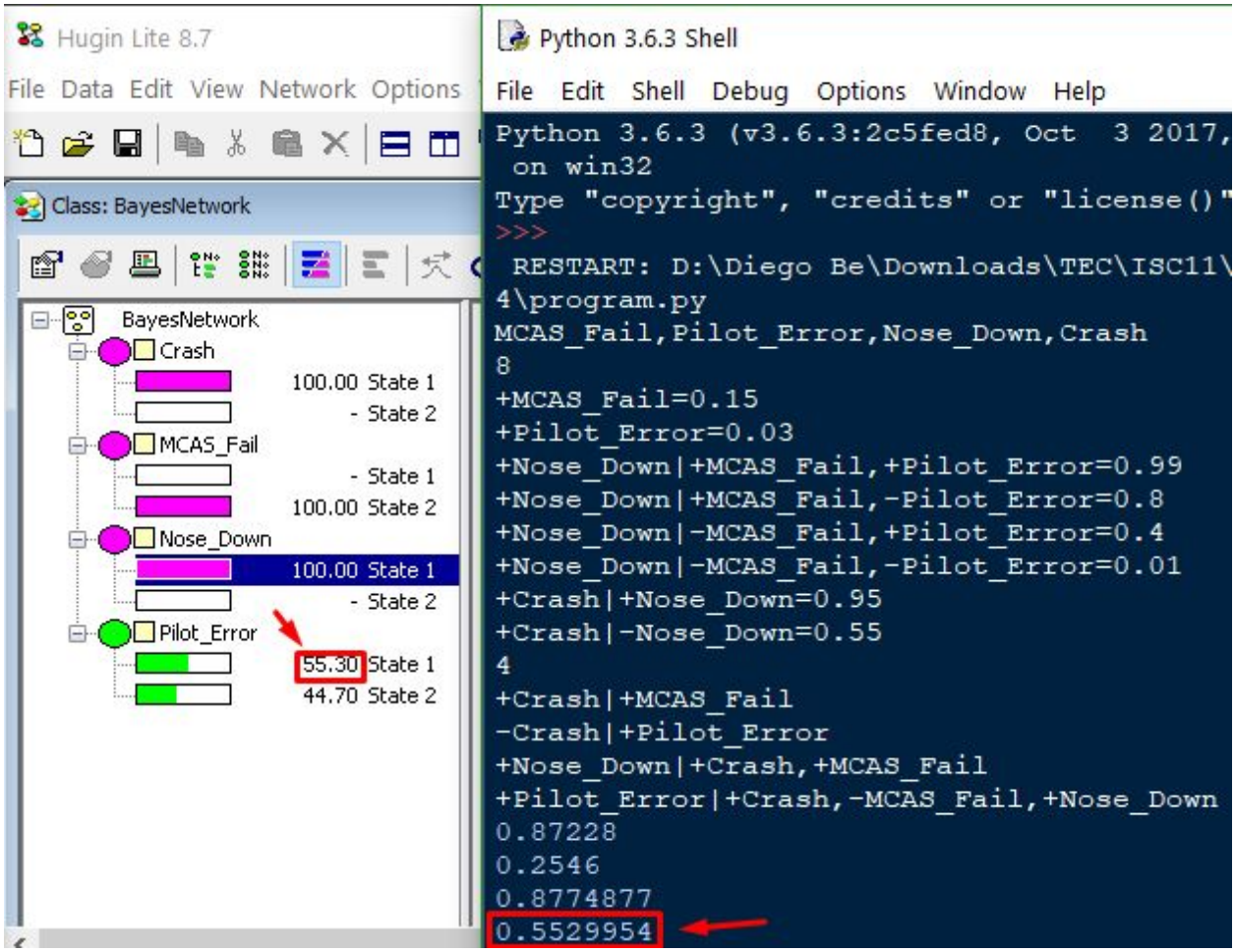


Diego Betanzos
Mariana Pérez

The following queries were tested on both Hugin Lite and the developed application, the results are:

Query	Hugin Lite	Python Program
A plane crashes given that the MCAS system failed.	87.23	87.228
A plane doesn't crash given that there was a pilot error.	25.46	25.46
The nose of the plane is down given that the plane crashed and the MCAS system failed.	87.75	87.74877
There was a pilot error given that the plane crashed, the MCAS system didn't fail and the nose of the plane was down.	55.3	55.29954

Proof, showing the last query in Hugin:



Diego Betanzos
Mariana Pérez

Reflection

The results from both, Hugin and the team's implementation are "identical", the quotation marks are there because Hugin rounds the result to the closest decimal, while the Python implementation doesn't, but the results are the same if both did the same rounding. Regarding the algorithm used by Hugin, it is difficult to say but most likely it is equal to the one implemented, since there's no noticeable difference in the results.

For real life applications the team considers that any of the two tools will be a good option, since both give the same results, but in general, it would be better and recommended to use Hugin Lite, mostly because it is a tool that has been in the market for more time than the team's implementation and it has been proven to work properly in different situations. Other good reasons to pick Hugin Lite over the Python program are:

- Proper, detailed and public documentation on how to use it, what it's for, different things that you can do.
- There are blogs from the community where you can ask questions regarding your issues using it.
- It includes more functionalities that can be useful for the user.
- It has more users from a long time ago, which means that the user has better chances to find the solution for a problem in the internet, from someone who had the same issue earlier.
- Possibility to print, import, export Bayes Networks and have a consistent and safe way to store them.
- Support from the developers.

The only situation in which the team would recommend using the Python program instead of Hugin Lite is when the user is used to work with the terminal and prefers this type of solutions over the ones which have a graphical interface and the user is familiar with the way of constructing the based network and doing queries in the program. This is mentioned because the team members had a hard time building the bayes network in Hugin Lite, the documentation is a bit confusing because it explains more things than what is required to build it and start querying, the controls are not intuitive, and the interface is not as friend as it could. The most confusing parts:

- Trying to add the probability values to each node.
- Determining which "State 1, State 2" belongs to which node.
- Doing the queries (specifying which is true and which is false for the givens).