Give the html time to load up and then it runs the javascript…

Best practice is to run the javascript at the bottom of the body tag

Document.ready guarantees that the JS is not going to load before the options populate in the select menu. Like a promise…

Convert to .py file and wrap in a function that returns an output. Kind of like the mission to mars HW…

If it was just displaying pre run model results, then I could do a query on CSV file. Like a .filter()

You can send JavaScript stuff to a PY file

Wrap the entire machine learning NB get it into a PY file and wrap it into a function, then have that function require a parameter.

In the route, have the user send back a parameter to evoke the machine learning file I created calling the function and feeding the input back to the front end

The route returns the result back to the front end. Route has a request and a response. Request processes the machine learning in function and the response sends output back to end user

1. Get machine learning loaded in then create a route that feeds it static data to make sure it gives output and doesn’t error out
2. Then, create a dynamic route to see if I can dynamically get user input and run it through the machine learning (print the output to debug)
3. Then, figure out how to get that output back to the front end

-Biggest part is getting the machine learning into the PY file and working…. The other stuff is just literally transferring data.

Make a new JNB with the saved model to make it simpler….

python -m http.server --bind 127.0.0.1