Sea Turtle Conservation AI Project

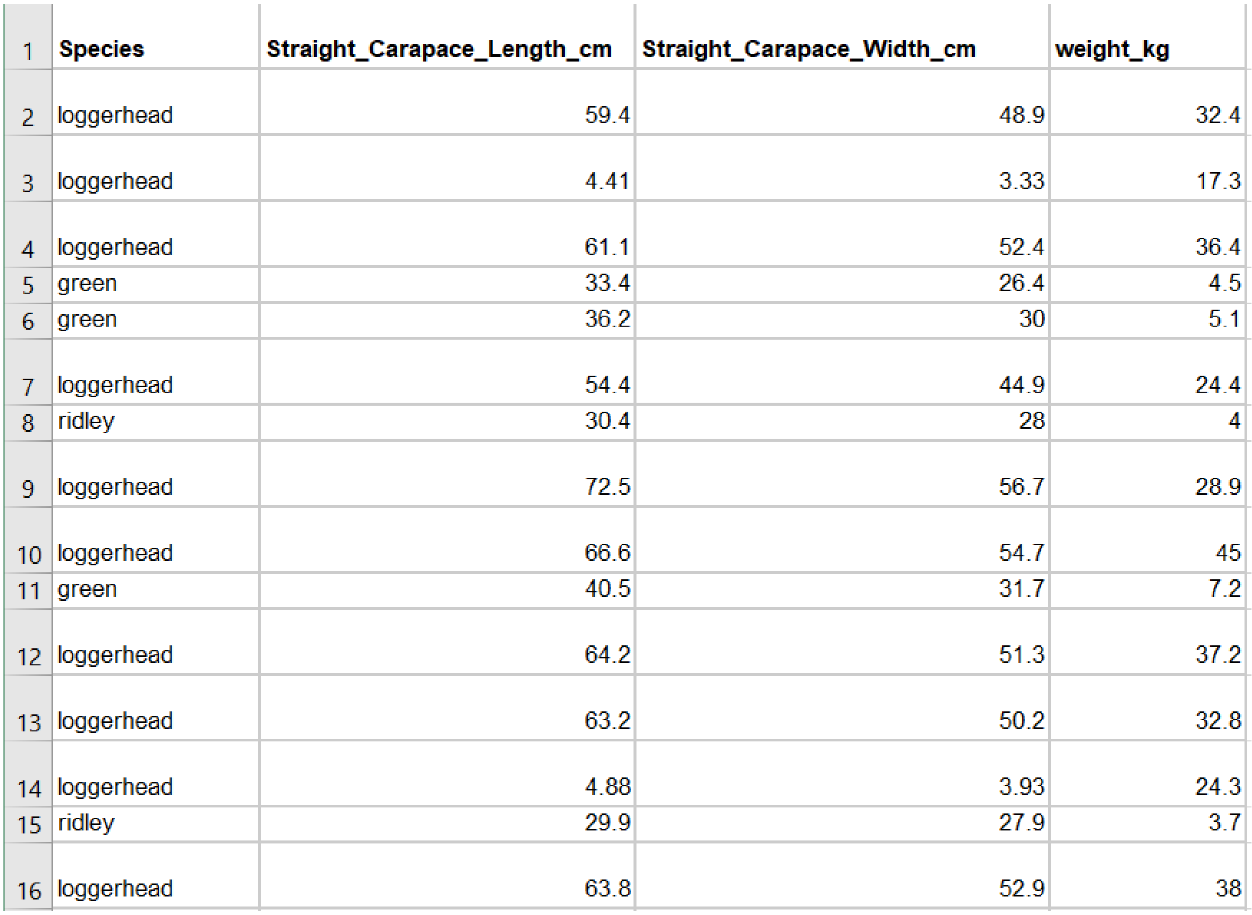
Part #2

EEL3872

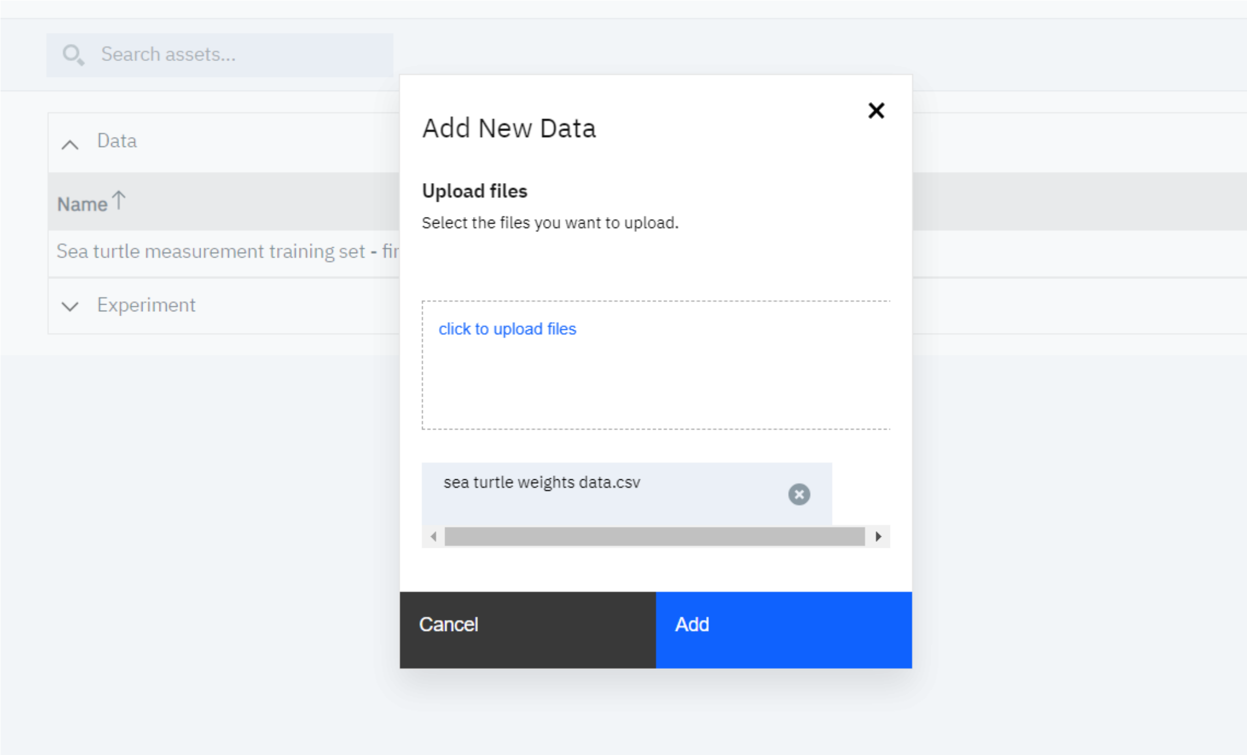
Jay Rosen

Experiment 2 – AutoKeras Regression

For this study, we analyzed a dataset of 5700 sea turtles' measurements, with datapoints for width(cm), length (cm), weight (kg), and species of sea turtles (green, loggerhead, hawksbill, and ridley). The csv file was uploaded to CogBot University for training Cammy, the AI Chatbot, using HiPerGator CPU. The AutoKeras Regression algorithm was used for training, which is automated machine learning built on top of TensorFlow library. AutoKeras Regression was setup with the prediction target set to weight (kg), and uses the characteristics of width, length, and species for calculating the predicted weight.



1	Species	Straight_Carapace_Length_cm	Straight_Carapace_Width_cm	weight_kg
2	loggerhead	59.4	48.9	32.4
3	loggerhead	4.41	3.33	17.3
4	loggerhead	61.1	52.4	36.4
5	green	33.4	26.4	4.5
6	green	36.2	30	5.1
7	loggerhead	54.4	44.9	24.4
8	ridley	30.4	28	4
9	loggerhead	72.5	56.7	28.9
10	loggerhead	66.6	54.7	45
11	green	40.5	31.7	7.2
12	loggerhead	64.2	51.3	37.2
13	loggerhead	63.2	50.2	32.8
14	loggerhead	4.88	3.93	24.3
15	ridley	29.9	27.9	3.7
16	loggerhead	63.8	52.9	38

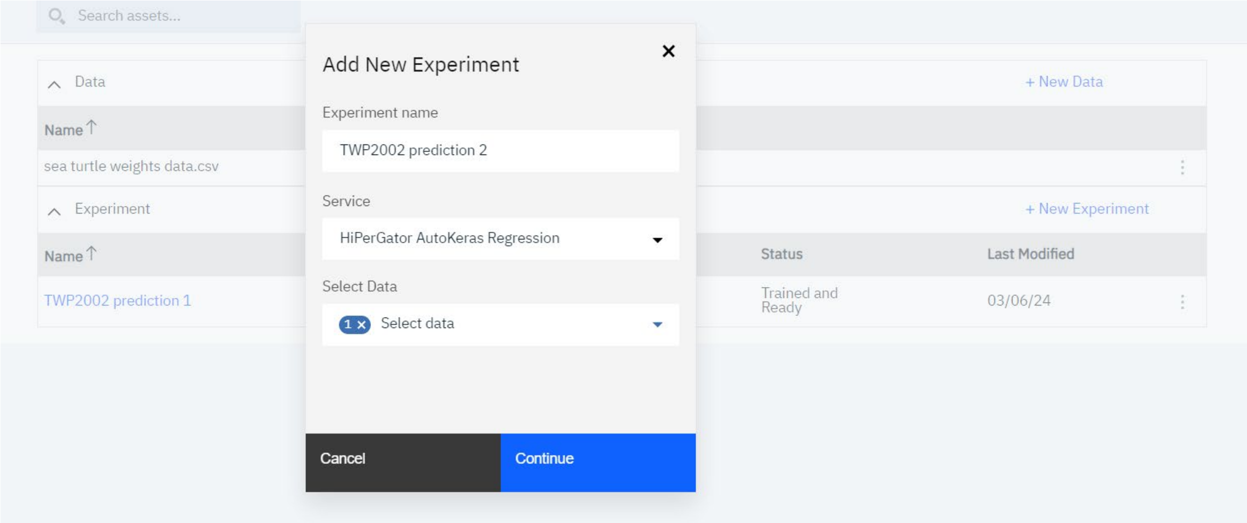


Add New Data

Upload files

Select the files you want to upload.

sea turtle weights data.csv



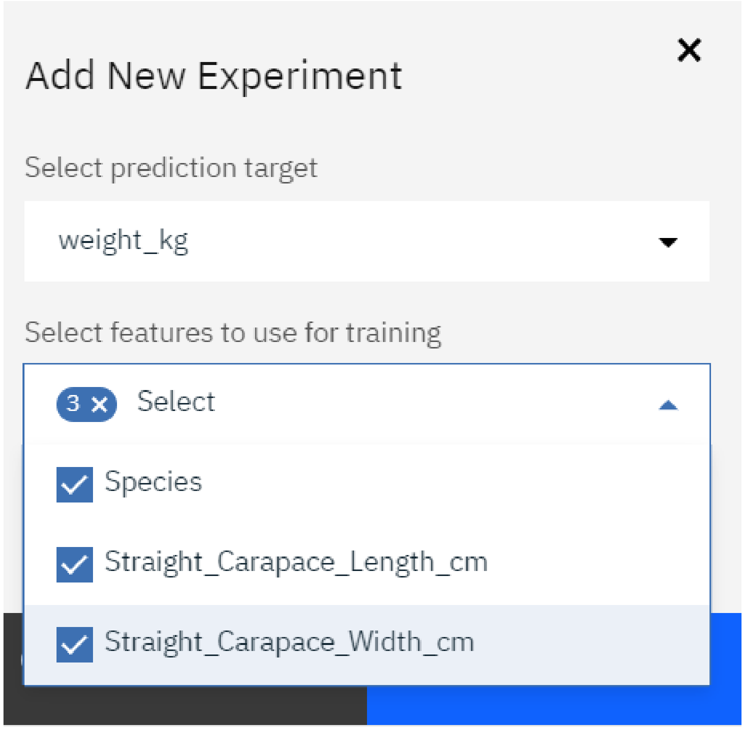
Add New Experiment

Experiment name: TWP2002 prediction 2

Service: HiPerGator AutoKeras Regression

Select Data: 1 × Select data

Name	Status	Last Modified
TWP2002 prediction 1	Trained and Ready	03/06/24



Add New Experiment

Select prediction target

weight_kg

Select features to use for training

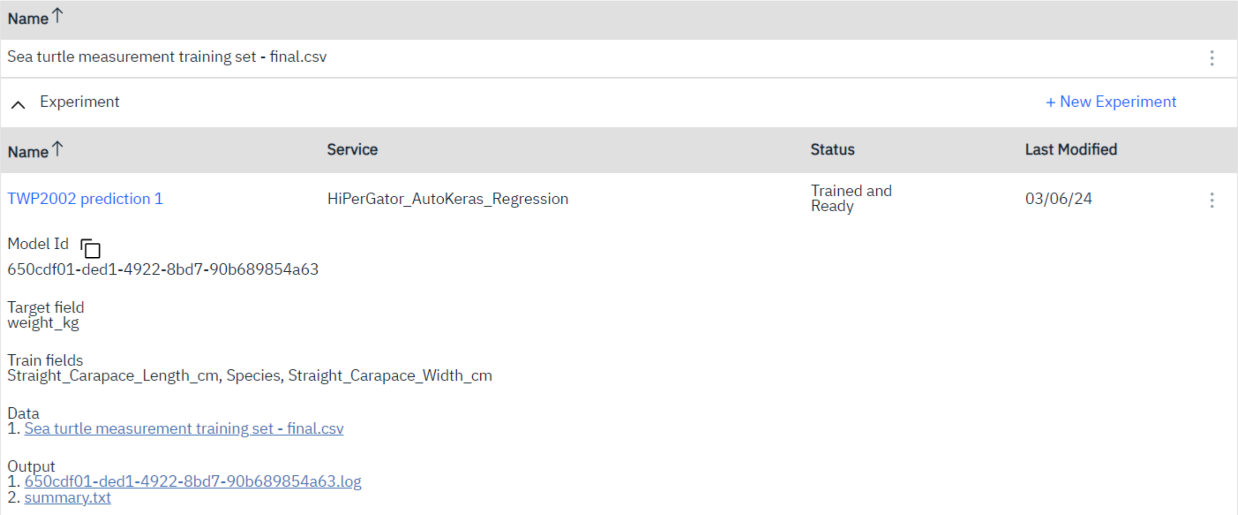
3 × Select

☑ Species

☑ Straight_Carapace_Length_cm

☑ Straight_Carapace_Width_cm

The training time took about 1 minute, and the output files showed the AutoKeras Regression was trained for 20 epochs. The 1st epoch had a high loss value of 22.7916, and the 2nd epoch had loss value of 5.54, which demonstrated substantial reduction in loss in this 1st cycle. Epoch 3 had loss value of 3.27, which continued to show learning with loss cut in half. From Epoch 4 through 20, the loss and loss validation showed stabilization, staying in the 2.5 – 1.93 range. The lower loss demonstrated the model could recognize pattern of species and dimension correlating to weight. The output summary showed that the model was trained on 1,224 total parameters, and only had 7 non-trainable parameters.



Name ↑

Sea turtle measurement training set - final.csv

^ Experiment   + New Experiment

Name ↑	Service	Status	Last Modified
TWP2002 prediction 1	HiPerGator_AutoKeras_Regression	Trained and Ready	03/06/24

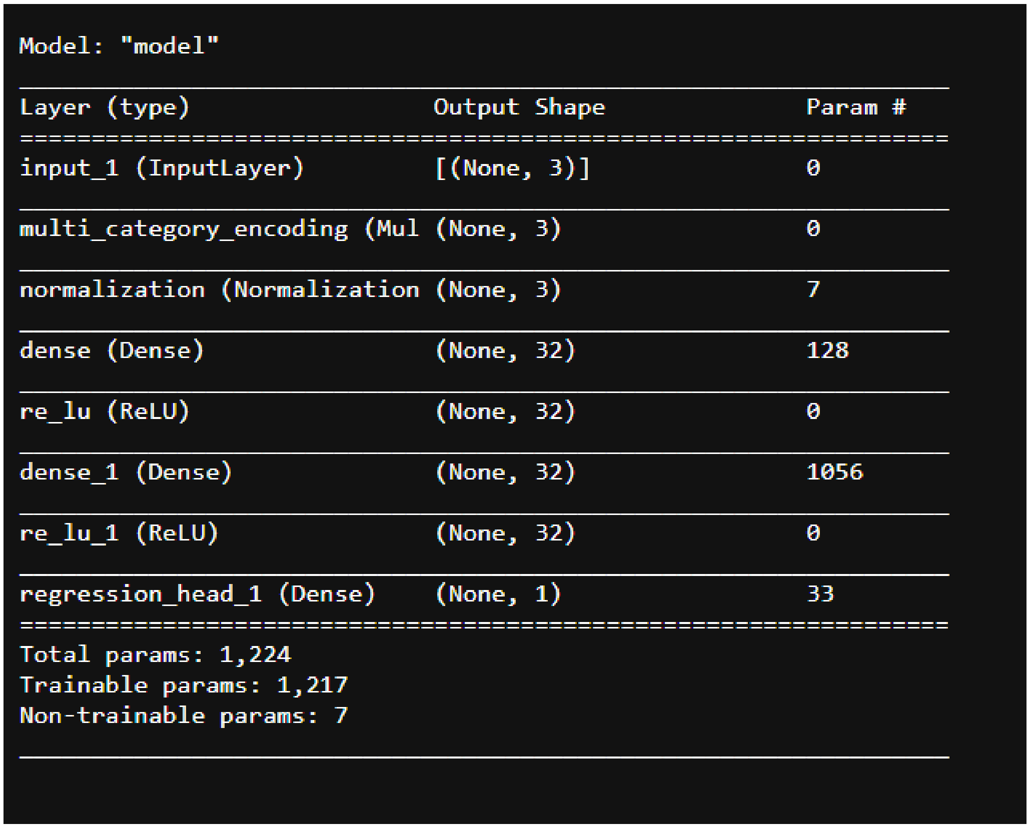
Model Id
650cdf01-ded1-4922-8bd7-90b689854a63

Target field
weight_kg

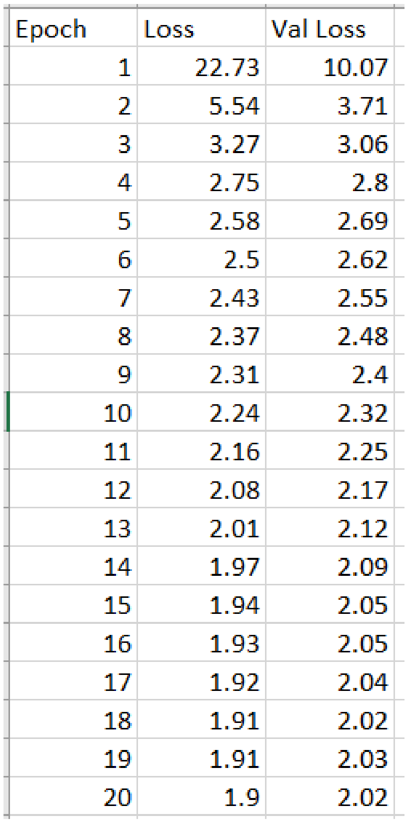
Train fields
Straight_Carapace_Length_cm, Species, Straight_Carapace_Width_cm

Data
1. Sea turtle measurement training set - final.csv

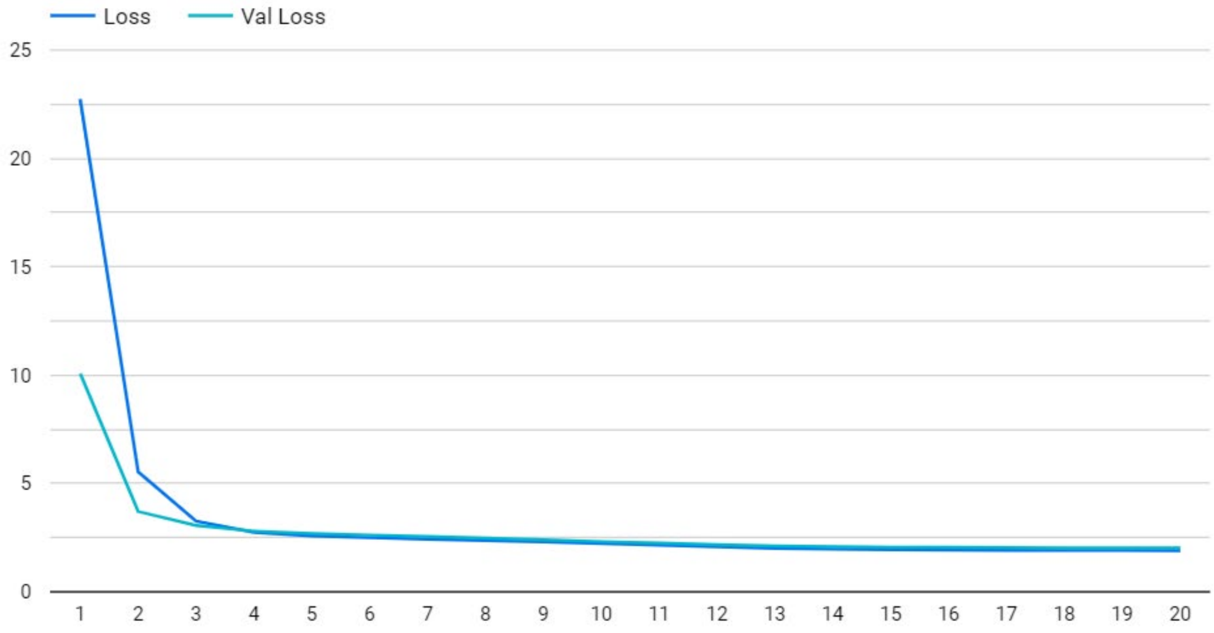
Output
1. 650cdf01-ded1-4922-8bd7-90b689854a63.log
2. summary.txt



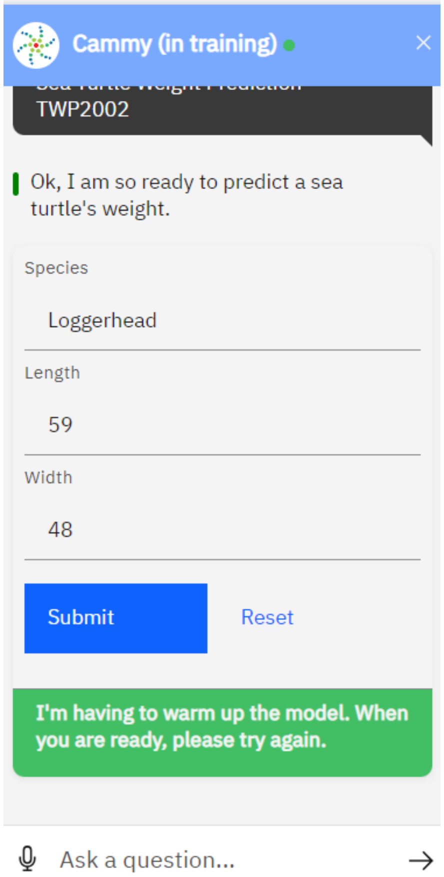
```
Model: "model"
_________________________________________________________________
Layer (type)                Output Shape              Param #
=================================================================
input_1 (InputLayer)        [(None, 3)]               0
_________________________________________________________________
multi_category_encoding (Mul (None, 3)                0
_________________________________________________________________
normalization (Normalization (None, 3)                7
_________________________________________________________________
dense (Dense)               (None, 32)                128
_________________________________________________________________
re_lu (ReLU)                (None, 32)                0
_________________________________________________________________
dense_1 (Dense)             (None, 32)                1056
_________________________________________________________________
re_lu_1 (ReLU)              (None, 32)                0
_________________________________________________________________
regression_head_1 (Dense)   (None, 1)                 33
=================================================================
Total params: 1,224
Trainable params: 1,217
Non-trainable params: 7
_________________________________________________________________
```



Epoch	Loss	Val Loss
1	22.73	10.07
2	5.54	3.71
3	3.27	3.06
4	2.75	2.8
5	2.58	2.69
6	2.5	2.62
7	2.43	2.55
8	2.37	2.48
9	2.31	2.4
10	2.24	2.32
11	2.16	2.25
12	2.08	2.17
13	2.01	2.12
14	1.97	2.09
15	1.94	2.05
16	1.93	2.05
17	1.92	2.04
18	1.91	2.02
19	1.91	2.03
20	1.9	2.02



The trained model had an ID 650cdf01-ded1-4922-8bd7-90b689854a63, and was given to Cammy chatbot for it to reference. Once Cammy had the model loaded to memory, it made a refence to The Matrix film, declaring Cammy knows Jiujitsu. Cammy then displayed an input form in the conversation window, to input species, width, and height. When trying this several times, I ran into issues, where Cammy would respond "I'm having to warm up the model, when you are ready, please try again.". This happened over the span of a few days, so I tried to re-upload the data and performed the experiment again of training a new model with AutoKeras on HiPerGator, using same setup.



Sea Turtle Weight Prediction
TWP2002

Ok, I am so ready to predict a sea turtle's weight.

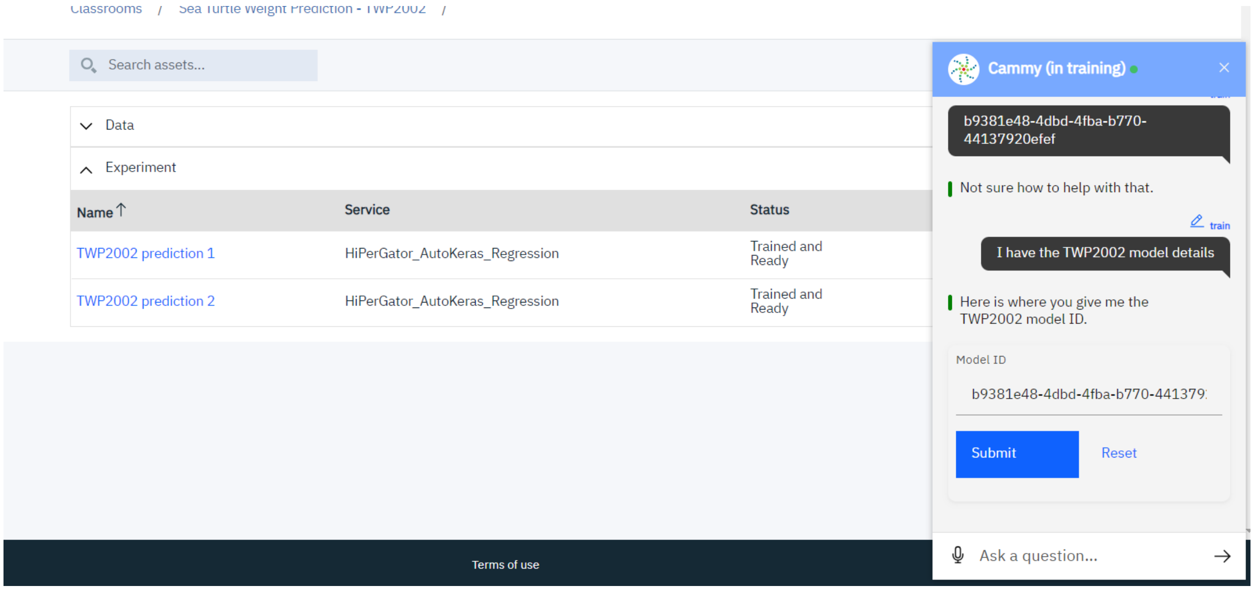
Species
Loggerhead

Length
59

Width
48

I'm having to warm up the model. When you are ready, please try again.

The 2nd trained model had ID b9381e48-4dbd-4fba-b770-44137920efef. To give Cammy the new model, I had to type "I have the TWP2002 model details". This process of giving Cammy a new model was not intuitive, since the default behavior of starting the training session would auto-populate the "I have the TWP2002 model details" to the chatbot, and no other option to change model unless say that again, without instructions of the usable commands. Typing /help did not give set of phrases to say to reset the model.



Name ↑	Service	Status
TWP2002 prediction 1	HiPerGator_AutoKeras_Regression	Trained and Ready
TWP2002 prediction 2	HiPerGator_AutoKeras_Regression	Trained and Ready

b9381e48-4dbd-4fba-b770-44137920efef

Not sure how to help with that.

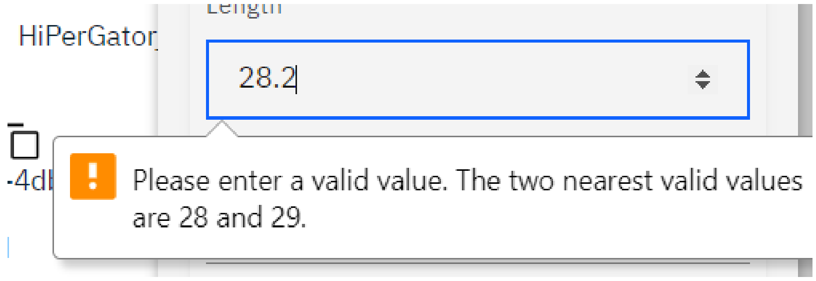
I have the TWP2002 model details

Here is where you give me the TWP2002 model ID.

Model ID

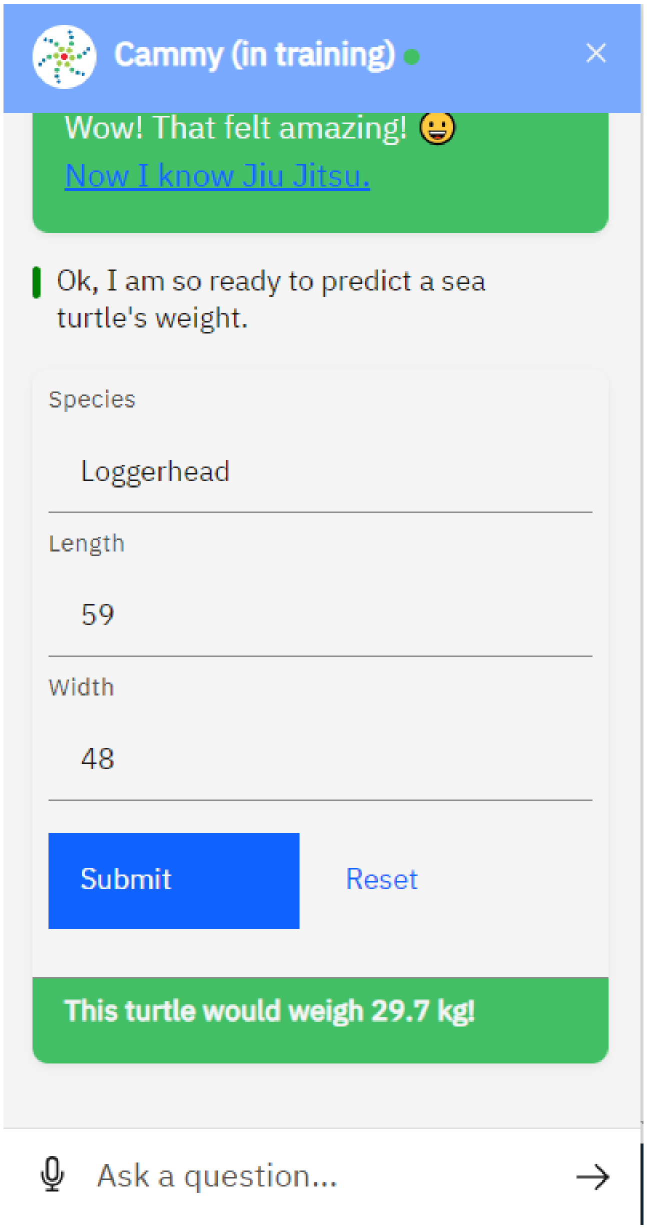
b9381e48-4dbd-4fba-b770-441379

The 2nd model trained did work, and I was able to see predicted weights by entering dimensions and species. I referenced the original dataset, testing 10 different rows of data, or 2-3 different sizes of the 4 turtle species. The answers were close to the actual weight of the turtle being measured. For example, a loggerhead that is 40 x 40 cm would be predicted to have a weight of 13.2 kg. One of the issues I see with the Cammy input, is that it would only accept whole integer numbers, and would not allow floating decimal numbers for input, even though it would output floating decimal numbers for weight. The model was trained on floating decimals in the dataset, so it seemed odd to be unable to use decimal input. This could show loss of data and loss of precision for the experiment, however, does make it easier for the end-user interacting with chatbot by just entering simple numbers.



28.2

Please enter a valid value. The two nearest valid values are 28 and 29.



Wow! That felt amazing! 😀
Now I know Jiu Jitsu.

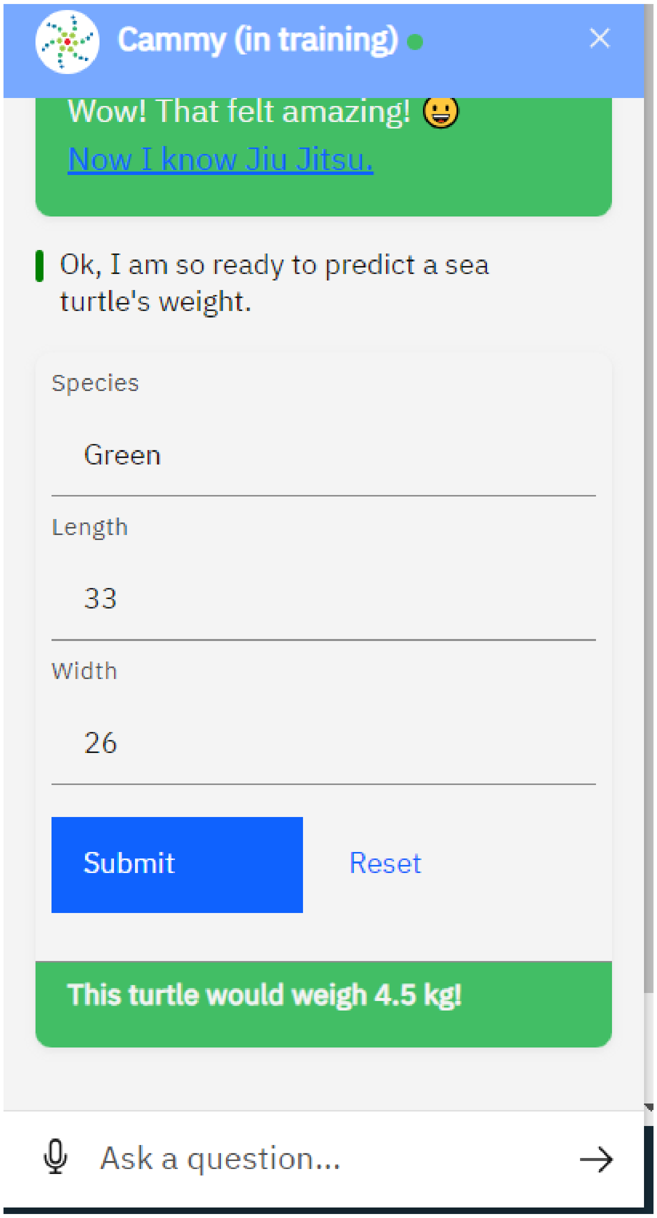
Ok, I am so ready to predict a sea turtle's weight.

Species
Loggerhead

Length
59

Width
48

This turtle would weigh 29.7 kg!



Wow! That felt amazing! 😃

Now I know Jiu Jitsu.

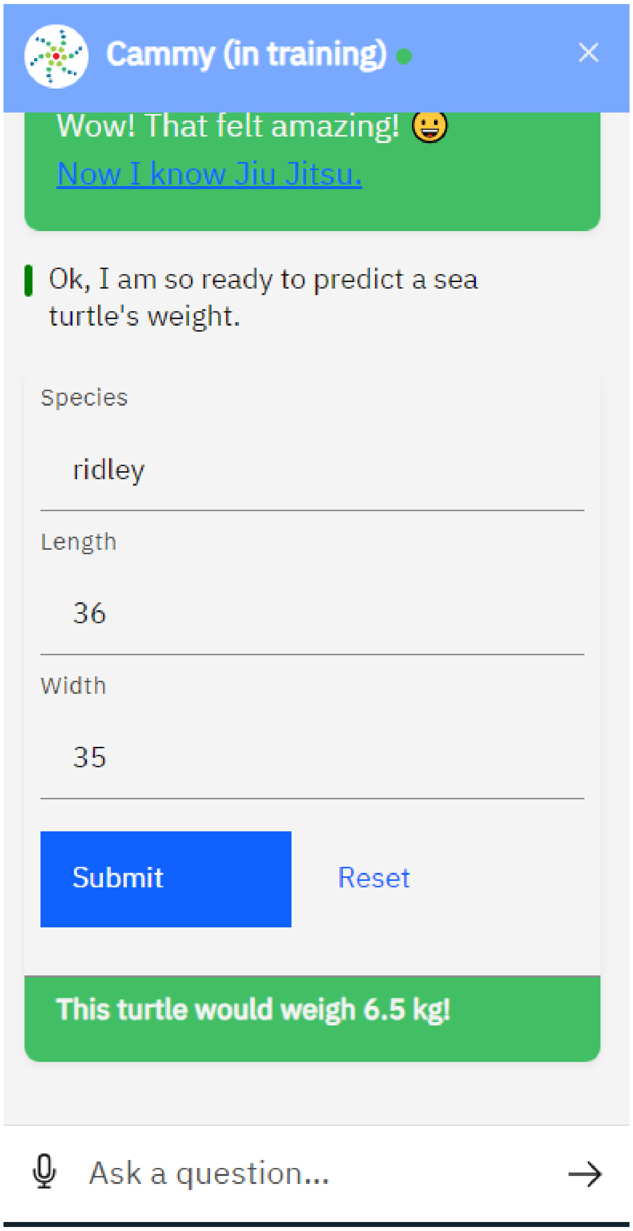
Ok, I am so ready to predict a sea turtle's weight.

Species: Green

Length: 33

Width: 26

This turtle would weigh 4.5 kg!



Cammy (in training)

Wow! That felt amazing! 😀
Now I know Jiu Jitsu.

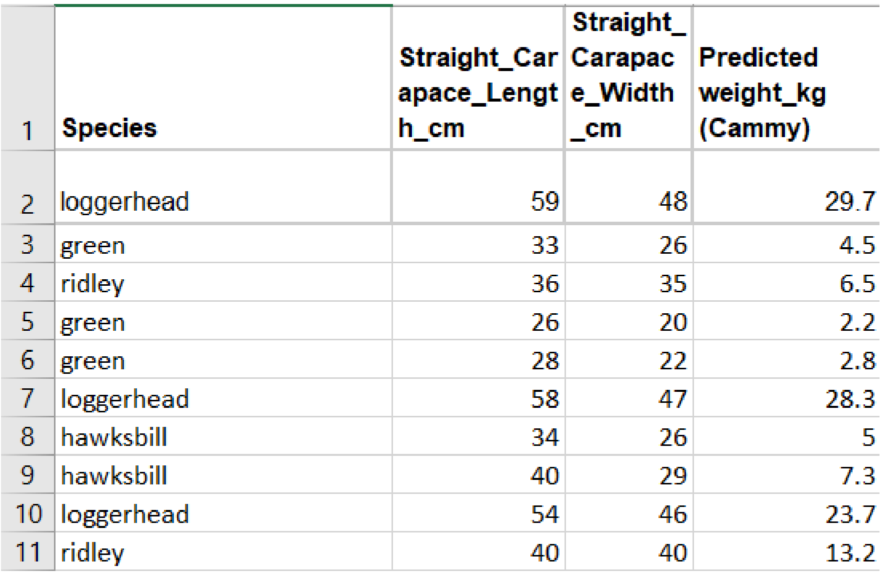
Ok, I am so ready to predict a sea turtle's weight.

Species: ridley

Length: 36

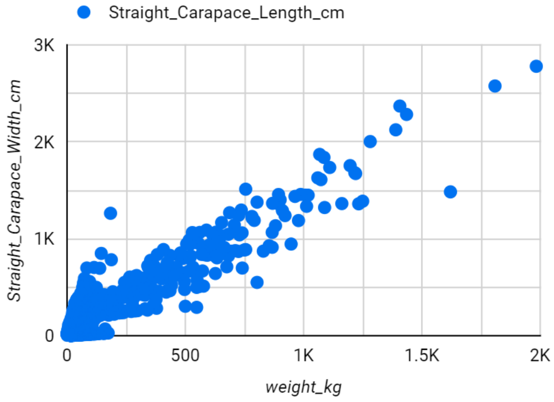
Width: 35

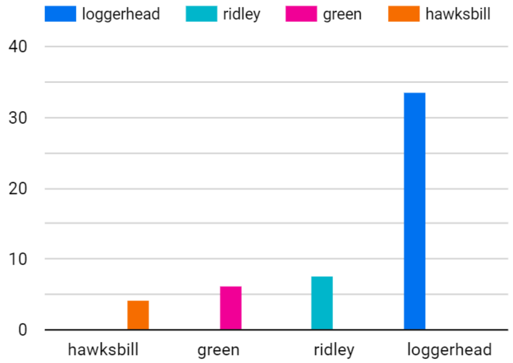
This turtle would weigh 6.5 kg!



1	Species	Straight_Carapace_Length_cm	Straight_Carapace_Width_cm	Predicted weight_kg (Cammy)
2	loggerhead	59	48	29.7
3	green	33	26	4.5
4	ridley	36	35	6.5
5	green	26	20	2.2
6	green	28	22	2.8
7	loggerhead	58	47	28.3
8	hawksbill	34	26	5
9	hawksbill	40	29	7.3
10	loggerhead	54	46	23.7
11	ridley	40	40	13.2

Google Looker Studio was used to chart the dataset of the 5700 turtles from the experiment. Scatter plot charts illustrated the correlation of dimensions and weight, and bar chart demonstrated the different sizes of species of turtle with Loggerheads appearing much larger than other species. Additional tables were added to chart the AutoKeras learning curve, and the experiment was logged as a csv of input integer dimensions and the predicted decimal weights for the 10 tests with Cammy.





Reflections and Future Use Cases

Tools like AutoML does the job well at learning a dataset to predict outcome with minimal setup required. This is great to perform more predictive experiments in shorter amounts of time. I have used AutoKeras in past projects when I wanted to quickly setup an AI project to demonstrate a concept for a new app, without having to learn everything about Machine Learning. This makes it easy to approach building with AI for non-technical folks and felt like a no-code tool – just bring your data, and the computer does the work.

Google Looker Studio is a great tool for analytics and displaying data. The UI and buttons feel more intuitive and inviting to use than Excel, which requires knowledge of formulas to make charts. Looker Studio can connect to many apps to for data, which also makes it usable in many scenarios.

I plan to use AutoKeras and LookerStudio in the future for AI and Analytics projects. As a web developer at UF, I use Looker Studio to share simplified reports of Google Analytics for webpage statistics, and I'm curious how I can use AutoKeras to make predictions for web traffic as we develop new websites. Another project working on is with the Florida Museum of Natural History, an app for Museum visitors to learn more about the exhibits like the Butterfly Garden. My Butterfly AI app was trained using Google Thinking Machines, an AutoML like AutoKeras. This algorithm labeled the large data set of butterfly images with names, so that visitors could identify the butterflies in the garden using their smartphone cameras. Looker Studio could be used to map geographic data about the species of butterflies, and possibly predict varying migration patterns based on changing environmental conditions.