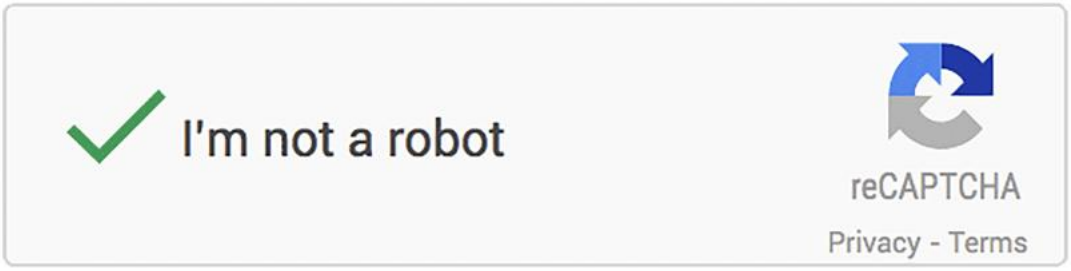
I’m Not a Robot

vs.

Data Scientist?



By. Alejandro Pesantez

Define The Questions

Throughout the data science community, image classification methods have been developing and adjusted to become even better. This led us to ask the question of how effective image classification tests are against an automated machine, and more specifically CAPTCHA tests, where many websites prompt users to select a certain type of image to prove that they are not a robot. With machine learning algorithms, we wanted to test the integrity of these CAPTCHA tests and wondered if an automated machine can pass the test with as great or even greater of an accuracy as a human. According to an article on CAPTCHA by the Baymard Institute in 2018, humans fail tests ranges from 8-29% of the time, which means humans on average tend to fail or misclassify 18.5% of the time. With that being said, given the provided CIFAR-10 image dataset, we wanted to see if we could create a machine learning algorithm that had an accuracy of above 81.5% to do better than the average failure rate of a human on these CAPTCHA tests. Now, let's look at the dataset and see how we will actually test this.

Get The Data

As previously mentioned above we will optimize an algorithm to correctly identify images with a low error rate using the CIFAR-10 image data set, which is a widely used dataset for machine learning and image recognition. The data consists of 10 classes of images: airplanes, cars, birds, cats, deer, dogs, frogs, horses, ships, and trucks. Each of the images are 32x32 color pixels with 6,000 images per class.

When prompted with the computer validation or CAPTCHA, a user will typically be presented with a variety of nine blurry images. A user will then be prompted to select all the images within a 3x3 screen that contain a type of object or animal. For this

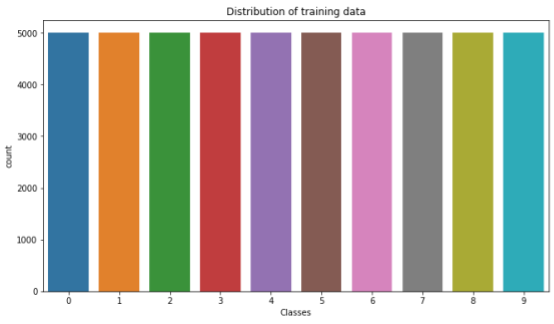
example, the team attempted to see how well an automated machine can decipher animals from objects as there is a 50/50 split of their presence in the dataset. If we were to focus on a 3x3 matrix, the machine would be prompted to choose all the animals. If accurate, the machine will select 6 of the 9 cells to be animals thus beating the website protector.

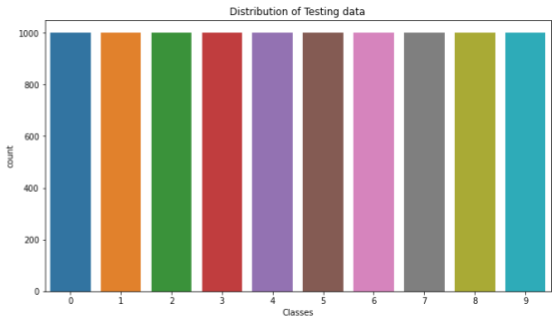
Based on the label classification on the data, the corresponding labels of object-oriented images such as airplane, automobile, ship, or truck were 0, 1, 8, and 9 respectively. The remaining labels corresponded to non-object-oriented, or animal, images such as bird, cat, deer, dog, frog, and horse which represented 2-7 respectively.

Since this is a binary classification problem, we relabeled the classifications of object-oriented images as 1, while animals are 0 for our target variable in both datasets.

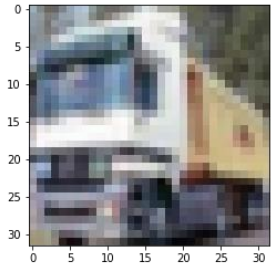
Explore The Data

When doing the EDA for our data we looked at a bunch of different things. According to the shape of the outputs, it appeared that each type of image had 6000 records in the entire CIFAR-10 dataset. Per object, there were 5000 images in the training dataset and 1000 images found within the test dataset. We can see this visually below.



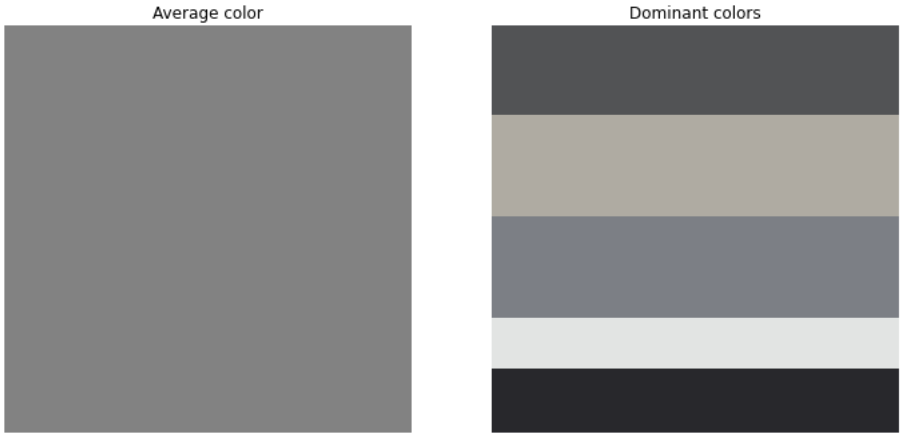


Once we figure out the distribution of our data, we wanted to look at how an image appears. We did so by pulling the first image and looking at whether it was classified as an object or an animal.

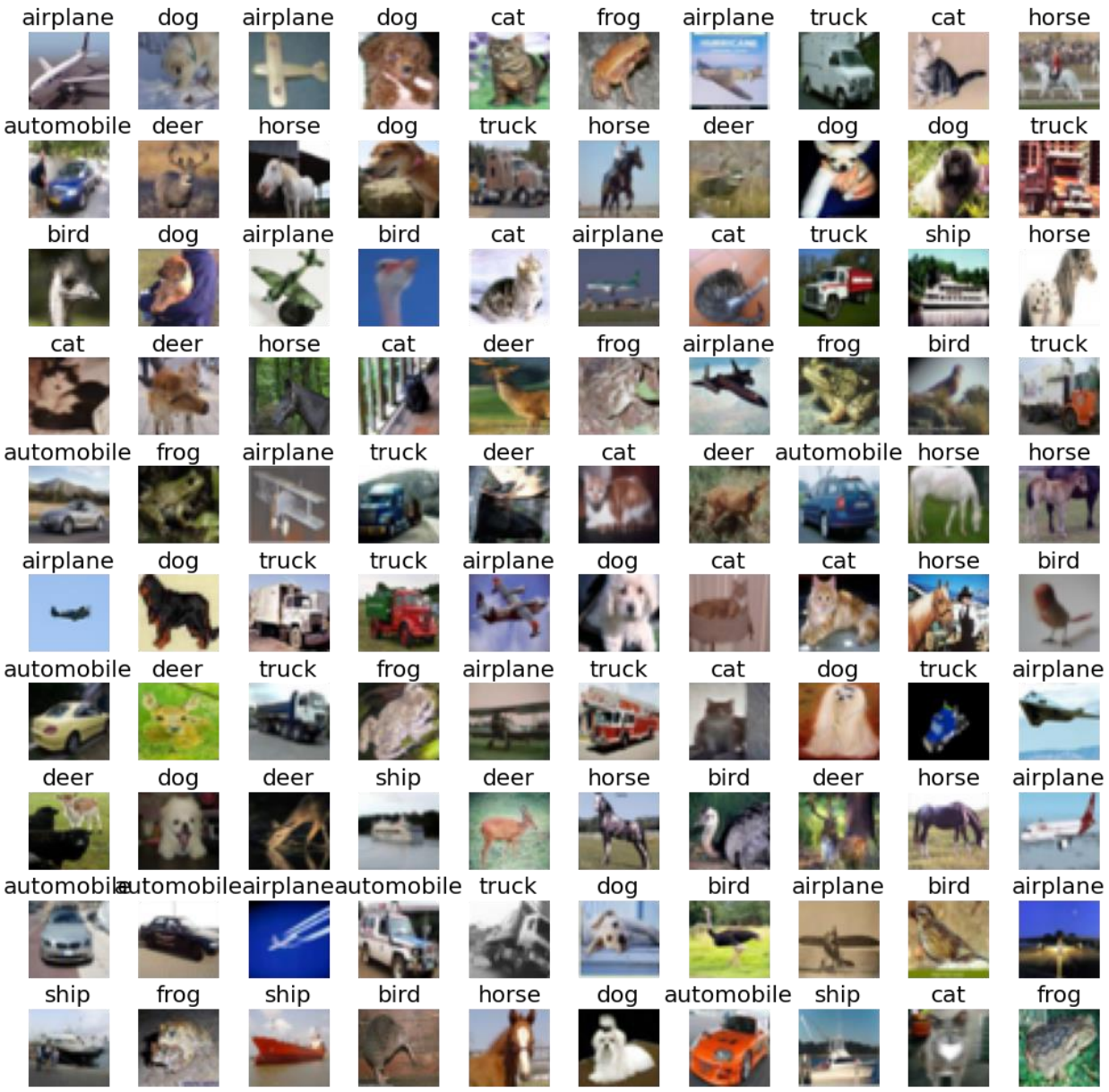


As a result, the object appeared to be a truck which meant the image was classified as a 0 or an object.

Based on the example we used above, we wanted to examine the color palette summary of the image. The code written allowed us to find the top 5 most prominent colors found among the image. For this example, we can look at the image above and its most used colors.



For a better visual of the overall data and the labels that are applied to it, we can see variations of the images and how they are classified below**.**



Preprocessing

After we did the EDA for our data, we started to begin our data preprocessing. The first step in our data preprocessing was to classify our images as either an animal or an object. We did so by creating a response variable for our models, which is what we will predict, that was 0 for animals and 1 for objects.

Next, we reshaped the input image data. Our data contained RGB color images which have 3 bands, so the raw image data is stored as a matrix with 32 rows, 32 columns, and a depth of 3.

This data was then flattened to 1 dimension to easily couple information across the dimensions. Without flattening, the data is viewed as disconnected slices of an image, which we can't put through a model. For example, information can be learned on a row-by-row basis but cannot be combined across different rows.

Next, the data was split into training, testing, and validation datasets with ⅔ of the original data in our training set and ⅙ of our data in each the testing and validation set. This had provided data to evaluate our models and prevent overfitting to the training data.

Lastly, the data was standardized so each feature, or data point, in the images had an average value of 0 and standard deviation of 1. This ensured each feature in the data will have an equal weight when modeling or in other words, to maintain a uniform range among the pixels of all the images.

Initial Model

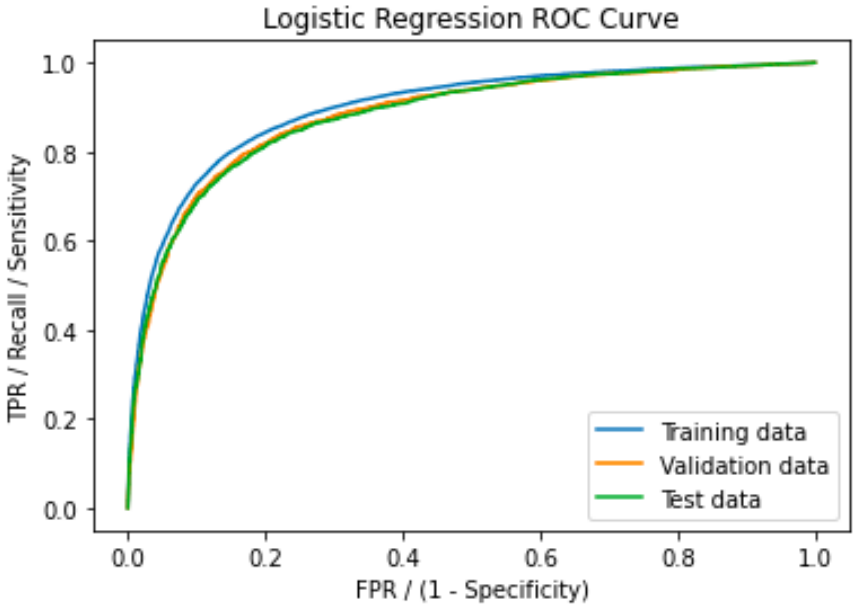
Once we were done with the data preprocessing step, we began to create our first model. To start modeling, we created a baseline logistic regression model. Logistic regression models, model the probability of an event from a combination of predictors. Given our data, the image type is predicted from a linear combination of image features.

The advantages of this type of model are that it's simple to implement and efficient to train. Additionally, it is easy for a person to interpret the effect of each predictor, or image feature, had in the model.

Unfortunately, there are some disadvantages as well. Logistic regression models tend to overfit when using high dimensional data. Each of our images had 3,072 data points so we needed to be cautious of this. Logistic regression models also assumes that the data is not multicollinear, and since our pixels are next to each other, they tend to be correlated due to items in images spanning multiple pixels.

However, with all the being the case, our logistic regression model accurately classified 81.6% of the images in our test data set. With this model alone we already reached our goal of 81.5% accuracy.

The plot below shows a ROC curve which illustrates the true positive rate, or rate that objects were classified as object, vs the false positive rate, or rate that animals were classified as objects. The training, testing, and validation, which is used as an additional testing data, lines show very similar performance, so this tells us that the model did not overfit to the training data. This model will be used as our baseline when creating a more complex model.



Although our logistic regression model performed well and already reached our goal of 81.5% accuracy we still wanted to push further and see if we could be even more accurate using a different and optimized model.

Model Optimization

The model we decided to use as our final and optimized model was a neural network. A neural network is just a method in AI that teaches computers to process data in a way that is inspired by the human brain. It's a part of a machine learning process, called deep learning, which uses interconnected nodes or neurons in a layered structure that resembles the human brain. A neural network is one of, if not the best model to use when it comes to image classification and since it resembles the human brain and is really good for image classification we wanted to see if this model's accuracy could beat an actual human's accuracy.

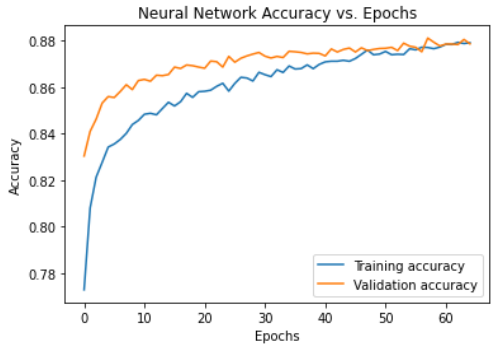
One advantage of a neural network is its efficiency, because unlike humans, a machine doesn't get tired if it runs within well-specified limits. They can also work continuously, saving a lot of time producing more remarkable results. They are also very good at multi-tasking and can produce multiple metrics and results while it runs, which is super helpful when trying to adjust features to optimize the model.

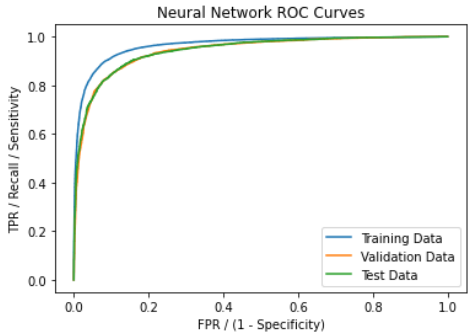
There are some disadvantages however, such as its data dependency. Whatever data is fed to the machine, it acts accordingly, which can be an issue since it might overfit to the given data and not produce a universal model. Another disadvantage of a neural network model is its "black box" nature, which basically means that we don't exactly know why or how a particular model came up with a certain output.

With all that being said, we still went ahead and used the neural network model. To begin we used one hidden layer with 29 neurons and a ReLU activation function, since when we ran the model these two features gave the best accuracy. We then decided to add a dropout layer since our model was overfitting to the training data and made the dropout rate 36% which again gave us the best accuracy for our model. Lastly, we added an output layer with a sigmoid activation function since we our doing a binary classification problem.

Then in order to compile and train our model we used an SGD (Stochastic Gradient Descent) optimizer, since for image classification this tends to be a better optimizer than the Adam optimizer because although SGD is slower, it generalizes better than the Adam optimizer. We then used a binary cross entropy loss function since we are again solving a binary classification problem. Lastly, we used a batch size of 200 and 65 epochs since after trying other values these two combinations seem to give the best accuracy for the model.

When the model was all set to run and was trained on the data, we then took a look to see if the model was overfitting at all. If you look at the graphs below, to the left you can see that as the model is being trained more and more to the input or “training” data it was given, the accuracy of the validation and training set actually starts to converge together, which is ideally what we want. However, if you look at the graph to the right you can see that the model actually did in fact overfit to the training data just slightly, which ultimately is the biggest difference between the human brain and the neural network, since the model can only learn on the given data.

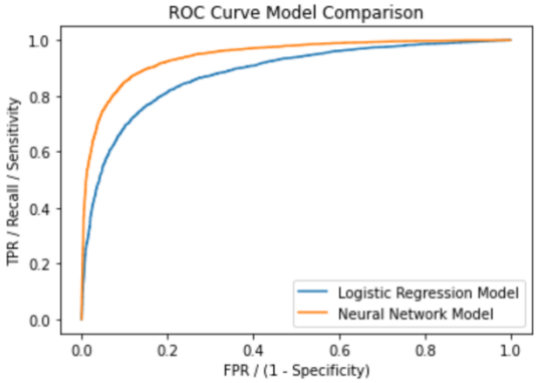




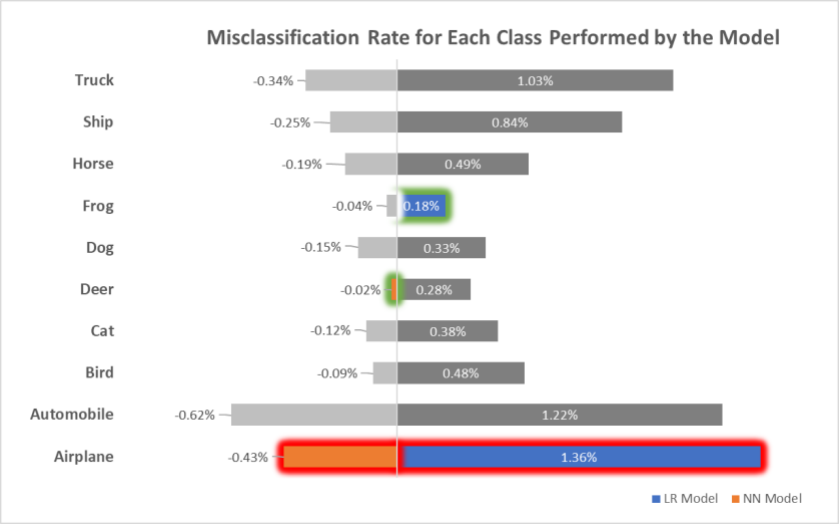
However, with all that still being the case, the model still has a very good accuracy on the test data set of around 87.8%, which is better than what most humans can do and our initial model!

Model Comparison

In order to truly show the comparison of the models we made some graphs which are shown below. We used Receiver Operating Characteristic curves and the Area Under the Curve values to compare visually and numerically the model performances. When comparing the two model's ROC curves we noticed that the Neural Network Model, highlighted in orange, outperformed the logistic regression model, highlighted in blue. Also, the AUC for the Neural Network model (0.942) was higher than the AUC of the logistic regression model (0.803), which means that overall, the Neural network model performed better than the logistic regression model.



Also, to go more in depth we made a plot showing which model was better at predicting specific images. We plotted the distribution of the misclassification rate for each image class based on the model performance. The left-hand side of the plot represents the neural network model, while the right-hand side of the plot represents the logistic regression model.



As you can see from the plot, although the neural network model produced an accuracy of 6.2% more than the logistic regression model, we noticed that the results for both models showed the class Airplane being the most misclassified, while the classes Deer and Frog were the least misclassified images

In addition, we had noticed that both models had misclassified objects, more so than animals. This is probably due to the data possibly being imbalanced since the proportion of objects to living-things are from 2:3.

In conclusion, our goal in our hypothesis was reached and we were able to create a model that was able to accurately decipher between types of images better than an average human can.