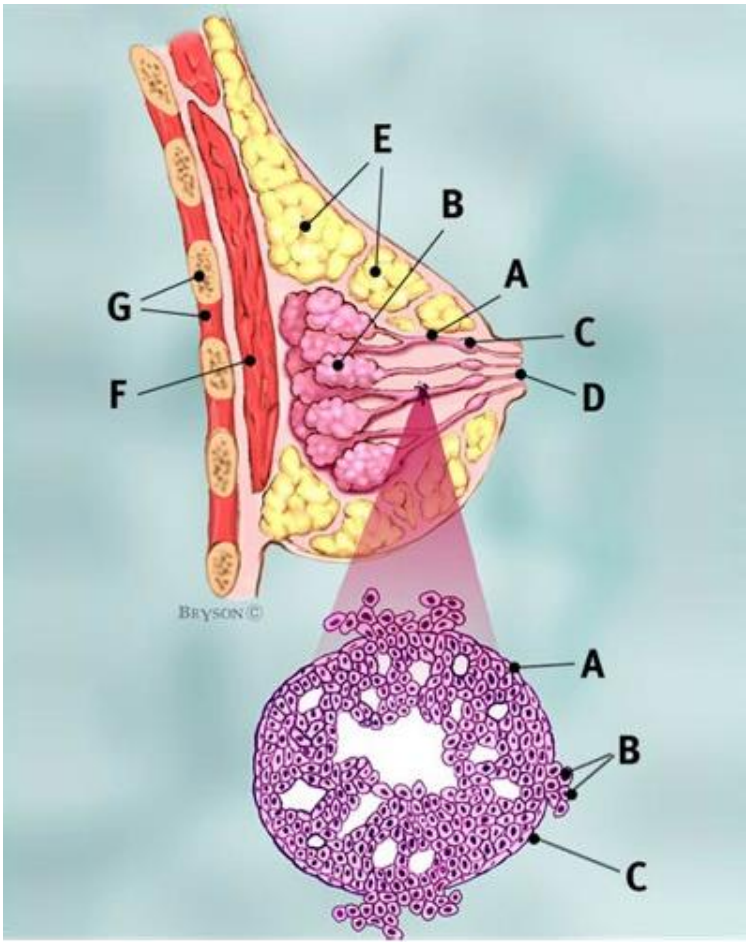
Using Deep Learning in Breast Cancer Detection

Invasive Ductal Carcinoma or IDC is the most common form of breast cancer making up nearly 80% of all breast cancer diagnoses. It is a cancer that breaks through the walls of the milk duct and spreads throughout the breast tissue where it can eventually access the lymph nodes and then metastasize throughout the body. More than 180,000 women in the United States are diagnosed with IDC every year. Although IDC can affect women (and even men to a lesser extent) at any age, ⅔ of all diagnoses are women over the age of 55.



(Source: breastcancer.org)

The population of the United States is aging rapidly. The number of Americans ages 65 and older will more than double over the next 40 years, according to the US Census Bureau. Currently, there are 3.5 working age adults for every american aged

65+. That ratio will fall to 2.5 by 2060.[1] This means that not only will we have a much larger population of older americans, there will be fewer young people capable of taking care of them. Optimizing medical care will be an even higher priority than what it is now.

Diagnosing IDC is currently done in a multitude of ways including physical exam, mammogram, ultrasound, MRI, and biopsy. The biopsy is done by gathering breast tissue from the patient and is examined under a microscope by a pathologist to look for cancer cells. If we could use deep learning to automate this time consuming process, the doctor that is normally tasked with detection could be freed up for treatment.

Breast cancer runs in my family. My grandmother and aunt have both been diagnosed with it in the past. In addition, my mother was diagnosed with melanoma and I, myself am a survivor of Hodgkin's Lymphoma. This is a subject that hits home for me and I love having the opportunity to use what I've learned to work on a subject that's so important to me.

The Data

The dataset is from a study that was done back in 2014 entitled "*Automatic detection of invasive ductal carcinoma in whole slide images with Convolutional Neural Networks* " which you can see here. Their best performing model got an F1 of 71.8% and accuracy of 84.23%. 6 years is nearly an eternity in the Data Science world and the massive advancements in this field over that time make me believe that even with my novice experience, that we can outperform those metrics.

Here is a description of the data direct from kaggle:
"The original dataset consisted of 162 whole mount slide images of Breast Cancer (BCa) specimens scanned at 40x. From that, 277,524 patches of size 50 x 50 were extracted (198,738 IDC negative and 78,786 IDC positive). Each patch's file name is of the format: uxXyYclassC.png — > example 10253idx5x1351y1101class0.png . Where u is the patient ID (10253idx5), X is the x-coordinate of where this patch was cropped from, Y is the y-coordinate of where this patch was cropped from, and C indicates the class where 0 is non-IDC and 1 is IDC."

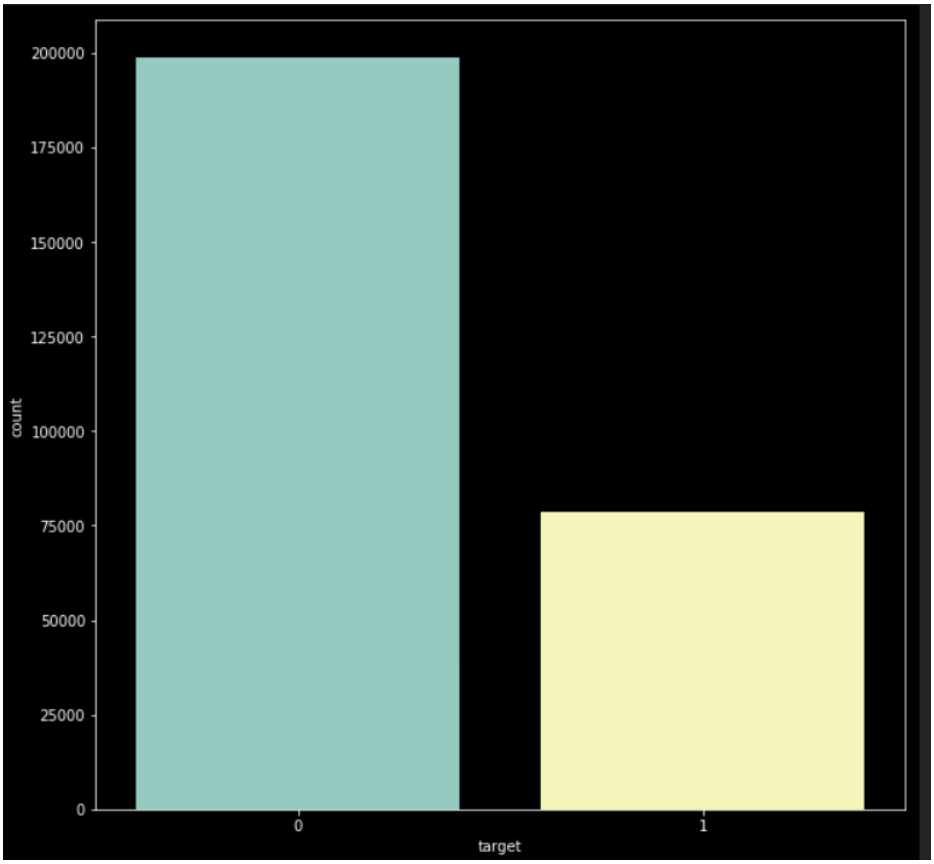
[1] https://www.census.gov/prod/2014pubs/p25-1140.pdf

Furthermore, each patient has their own folder which is further subdivided into either positive or negative for cancer. This was a very clean dataset and I'd like to give a quick shout-out to whoever put in all the hard work of putting this together and Paul Mooney for uploading it to Kaggle.

Exploratory Data Analysis

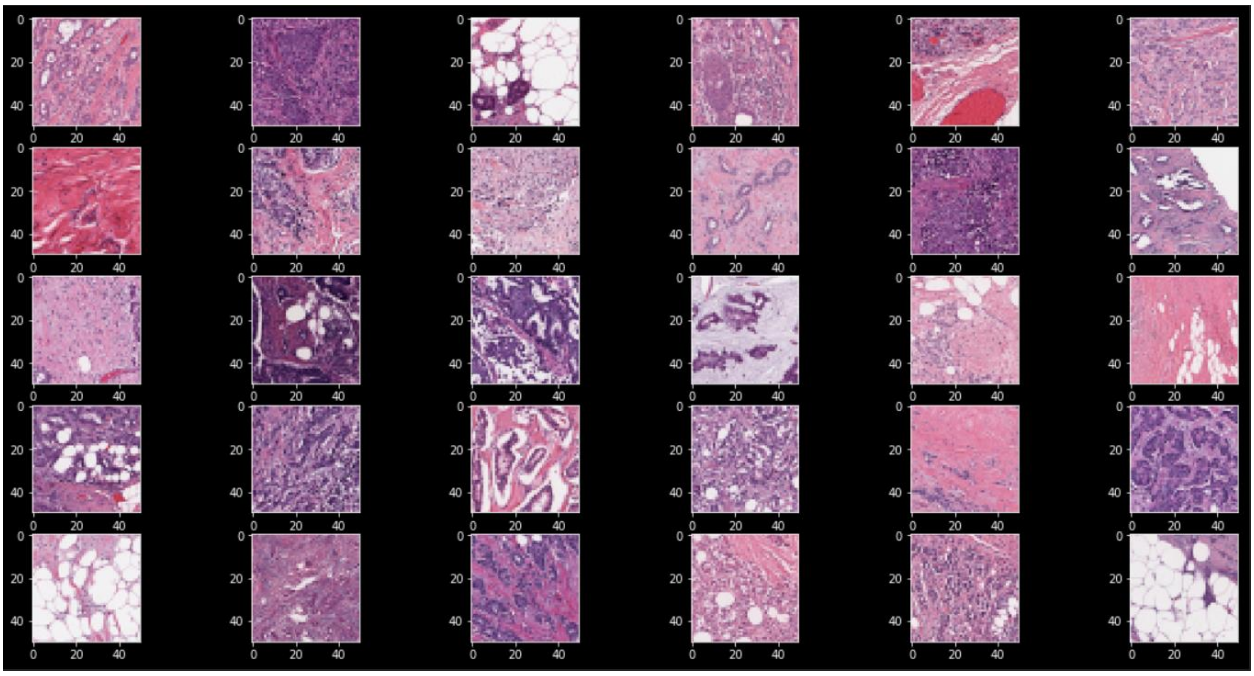
For performing EDA on images, one must get creative. Thankfully, the meticulous labelling and sorting that was performed by some hero in the past makes this task much easier for us. I use a bit of string manipulation in a function to take our filenames and convert them into a Dataframe. Each row is an image and the columns consist of the patient id, the x coordinate of the image, the y coordinate of the image, our target feature, and the filename).

First, let's see the split between IDC (cancerous) and non-IDC (non-cancerous) images:

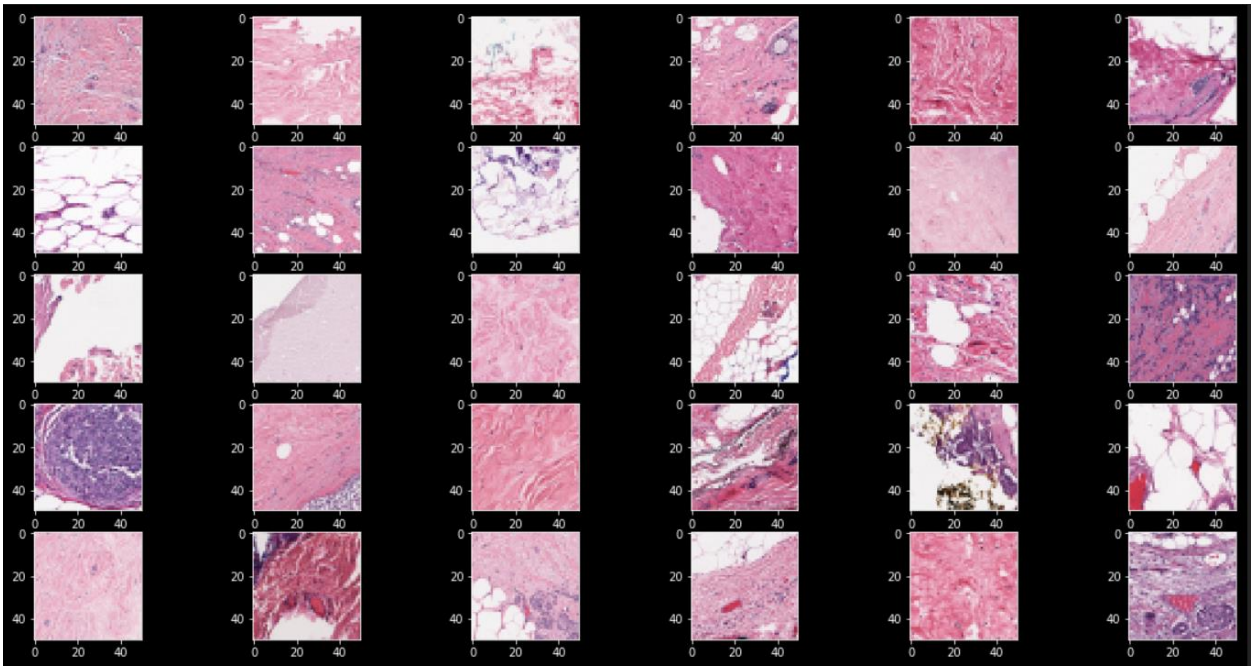


Green is cancerous and yellow is not. This data is highly imbalanced. Cancerous slides only make up about 28% of the total. Something to keep in mind.

What do the images themselves look like? Here are the positive cases:



And the negative cases:



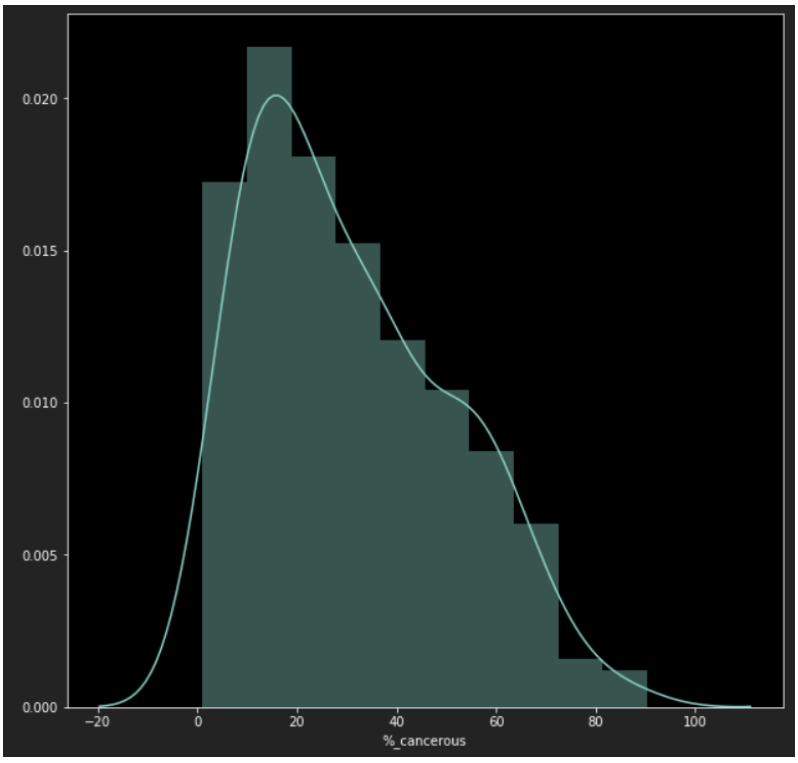
The positive cases appear to be a lot more purple but there are several purple images in the negative dataset as well. The model should do a better job than myself at reading the images and being able to differentiate the two.

I grouped the patients by their total number of images and calculated the % of their images that were cancerous. Here's the top 20:

```
patient_id
14209    90.35
9262     85.11
12873    82.56
9077     77.82
12241    75.66
8957     74.77
15633    74.72
10275    71.90
13694    70.73
15514    69.12
Name: %_cancerous, dtype: float64
```

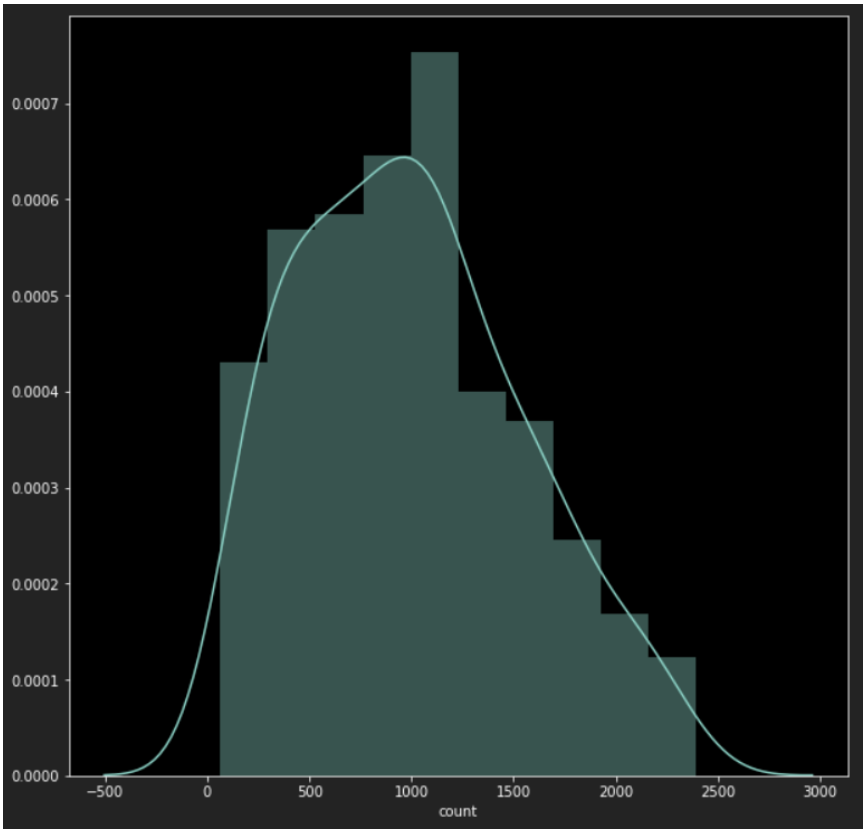
Incredible, one patient has cancer in 90% of their images. 61 of our 279 patients have cancer in a majority of their images. On the other hand, 43 of our patients have cancer in less than 10% of their images. The dataset varies quite a bit in terms of cancer percentages.

Here's a distribution plot of the % of cancer in each patient's images.



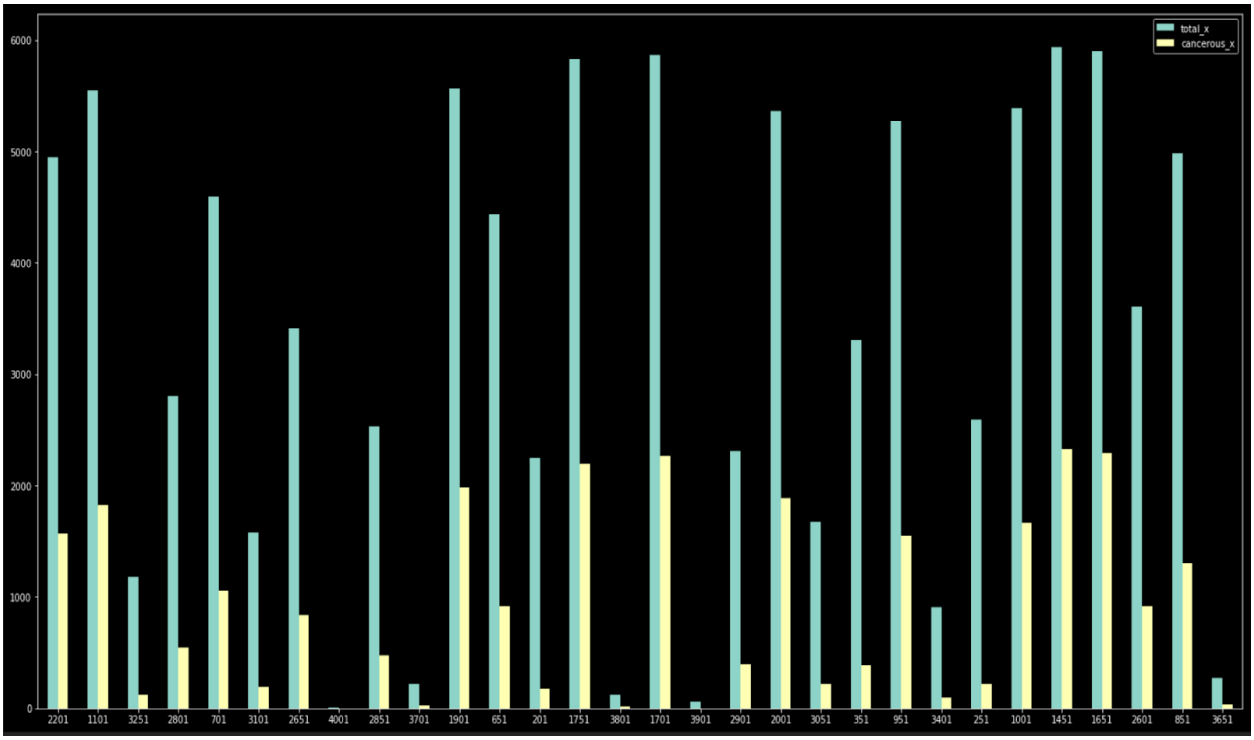
Looks like most are between 0% and 30% cancerous in their images.

Here's a distribution plot of the number of images each patient has.

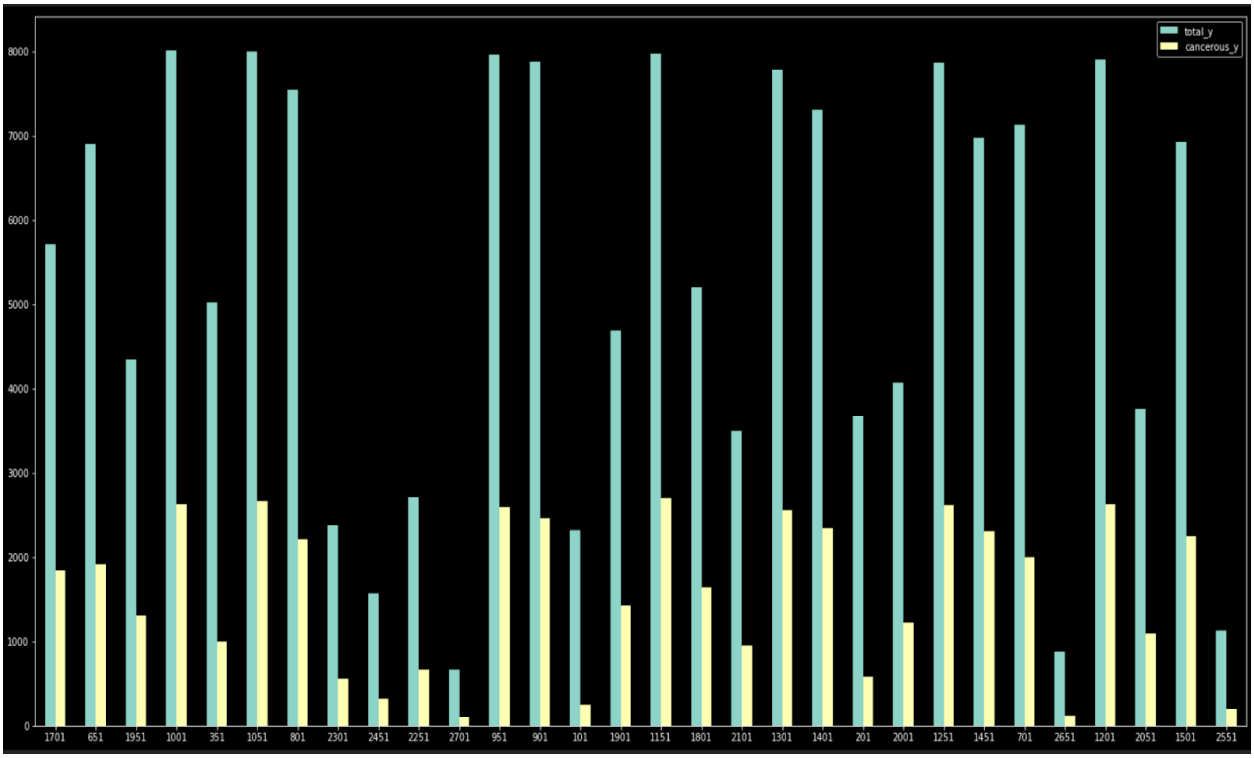


There's a surprisingly wide range on these as well. I guess whenever these biopsies are performed there can be quite a bit of variance in how much tissue is extracted and put onto a slide.

Finally, let's see if there is a relationship between the coordinates of the images and being positive for cancer. We will start with x coordinates and grab a random subset of 30. Green is total number images for that coordinate and yellow is the number of images of that coordinate that were cancerous:



There are only 81 distinct Xs and some coordinates clearly have a higher proportion of cancer than others. How about Ys? Again, we take a random subset of 30:



This is half of all of our y coordinates and again, the proportion of cancer definitely seems to vary between them.

Takeaways from EDA:

-Our data is imbalanced 71.6% to 28.3% so we will need to look at possibly resampling or adding more complexity to our model if the model has difficulty classifying due to this imbalance.

-The positive cases appear to be more purple than the negative images but there is almost certainly more at work than that.

-There is a wide variance between how many images each patient has in their folder.

-Some Xs and Ys seem more prone to having cancer in their images than others. If our model has difficulty classifying, we could possibly look at integrating the actual coordinates for each image into our model.

Preprocessing

As great as the file structure of these images is, I am going to have to sort them differently. I found a Python package called split_folders that will split my data into training, testing, and validation sets. I like this package because I can adjust the split ratios and sort images differently on the fly. The catch is that split_folders requires that your images of the same class must be in the same folder. Currently, the two different classes are in folders for each patient.

I run a simple script that filters the images based on their classification and then sort them into either the 0 or 1 folder. From there, we just run split_folders and split our data into 70% training, 15% testing, and 15% validation.

Now that we're sorted, we're going to use Keras's built in ImageDataGenerator function to preprocess our images. This package is great because it both allows us to preprocess our images to make augmentations and it also acts as an image loader into our model. I promise that I'm not sponsored by Keras.

We initialize the generator and implement height & width shifts, zooms, rotations, and a horizontal flip. We run this generator on our training, testing and validation sets.

Building and testing the Model

As this is my first Deep Learning project, the model will not be too complex. I make a sequential model that uses a lot of relu activation along with maxpooling, 2d convolutional layers, a dropout, and some dense layers.

I trained the model over 35 epochs and capped out at ~88% accuracy on my validation set. This simple model exceeded my expectations by far.

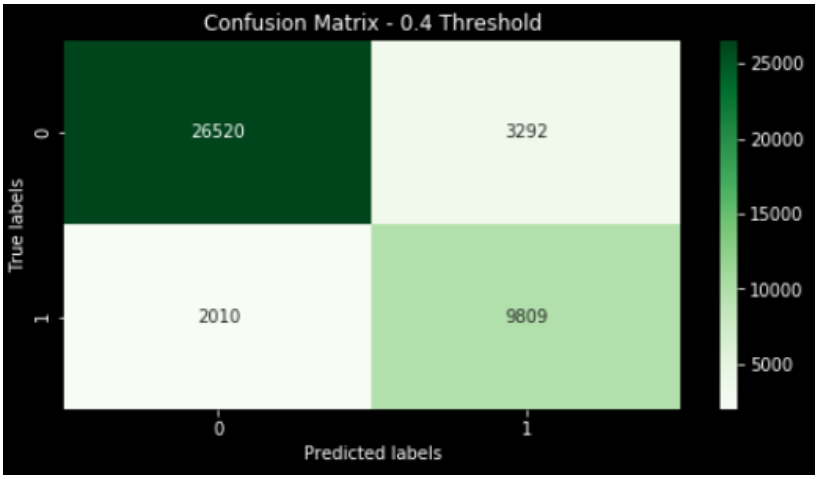
The real results are how we do on the test set though. After loading in the test set, the model evaluated with 87.85% accuracy! I am ecstatic with these results.

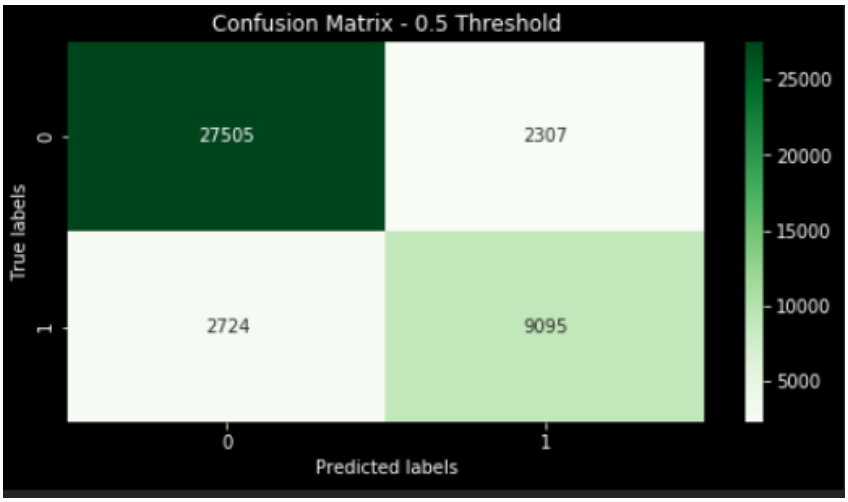
Determining the best threshold for Cancer

When you evaluate the model on the test set, it returns an array of floats between 0 and 1 that represent the likelihood of cancer being present. What number should we use as our "cancer" threshold? The rules of math state that .5 should always be rounded up to 1—a cancer prediction. However, we need keep our problem in mind when making this decision.

A false positive on a cancer diagnosis can cause panic with a patient and their family. But a false negative on a cancer diagnosis can very easily lead to their death depending on how fast the cancer spreads. It is my belief that we ought to prioritize eliminating false negatives as best as we can. We must strike a balance between reducing false negatives, while at the same not reducing our threshold so much that our false positives get too high to the point of the model no longer being useful.

After examining multiple thresholds, the final two were .4 and .5.





The bottom left corner of each matrix is the false negatives we're trying to avoid. Reducing these though affects every other cell in the matrix though. .4 has the lower false negatives but it also has more false positives, less true positives, but more true negatives. Whatever decision we make will have tradeoffs.

```
------------- Classification Report for .4 -------------
              precision    recall  f1-score   support

           0       0.93      0.89      0.91     29812
           1       0.75      0.83      0.79     11819

    accuracy                           0.87     41631
   macro avg       0.84      0.86      0.85     41631
weighted avg       0.88      0.87      0.87     41631
```

```
------------- Classification Report for .5 -------------
              precision    recall  f1-score   support

           0       0.91      0.92      0.92     29812
           1       0.80      0.77      0.78     11819

    accuracy                           0.88     41631
   macro avg       0.85      0.85      0.85     41631
weighted avg       0.88      0.88      0.88     41631
```

.5 has the slightly heigher weighted F1 score and it has better the precision for our target by .05. In my estimation though, Recall is the more important metric here

because it's finding more instances overall of cancer. This will lead to more false positives but I believe it's worth finding more instances of cancer in the long run. A false diagnosis of cancer will scare the patient until the truth comes out, but a missed diagnosis of cancer could possibly lead to death.

.4 should be our classification threshold for cancer.

Takeaways and Future Study

-F1 score and Accuracy aren't not necessarily the most important metrics depending on what is being prioritized in a given problem.

-A deep learning model does not need to be super complex to give good results.

-In the future, I'd like to make a more complex model and see how I can improve the results. I would also be curious to see how the model would perform on other data if I could get my hands on it.