Histopathologic Cancer Detection

Springboard-DSC Program
Capstone Project 1
May 2020

By Morgan Fry

Introduction

- *The Problem*
- Cancer detection is commonly done by human visual inspection of biopsy samples treated with contrast dye.
- *The Client*
- Healthcare providers and patients both have an interest in faster and more accurate analysis of these, as more accurate and earlier detection can ultimately improve treatment outcomes.
- *The Goal*
- A sufficiently accurate model for detection may save analysts time by reducing their workload to mostly edge cases or may improve on human results altogether.

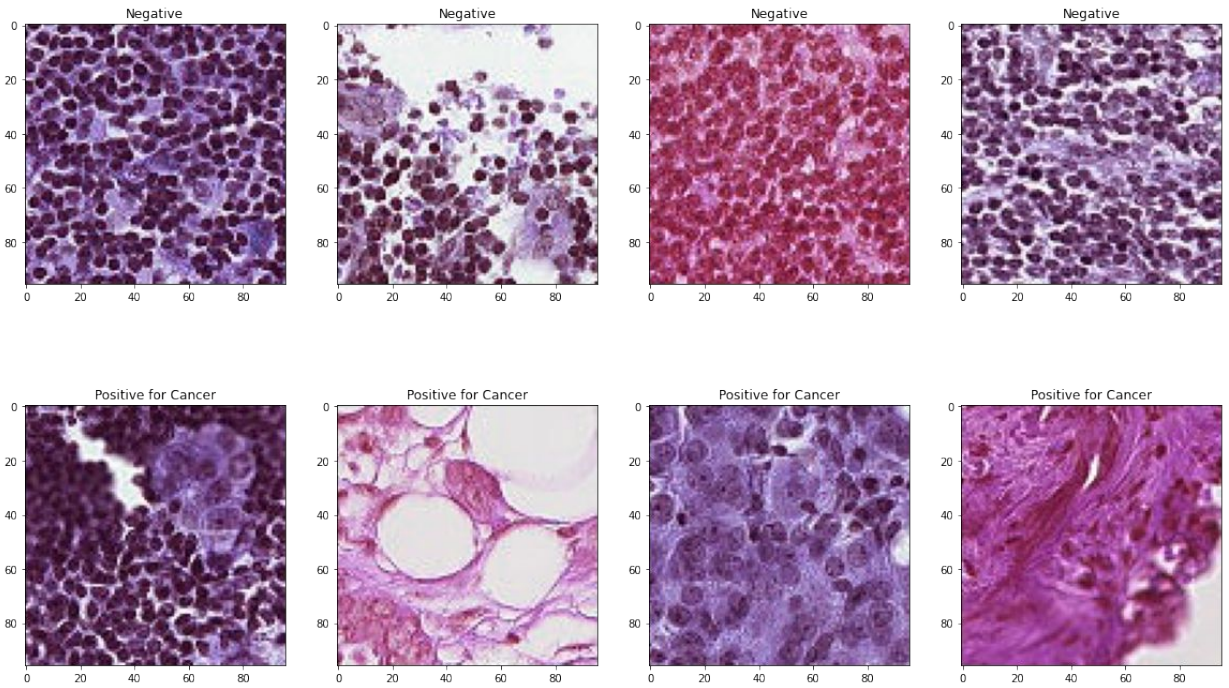
Data Science Problem and the Data

- This is a supervised learning, binary image classification problem.
- The dataset is available from Kaggle Histopathologic Cancer Detection competition, and is a subset of the PCam dataset for image classification benchmarking.

Data Acquisition and Wrangling

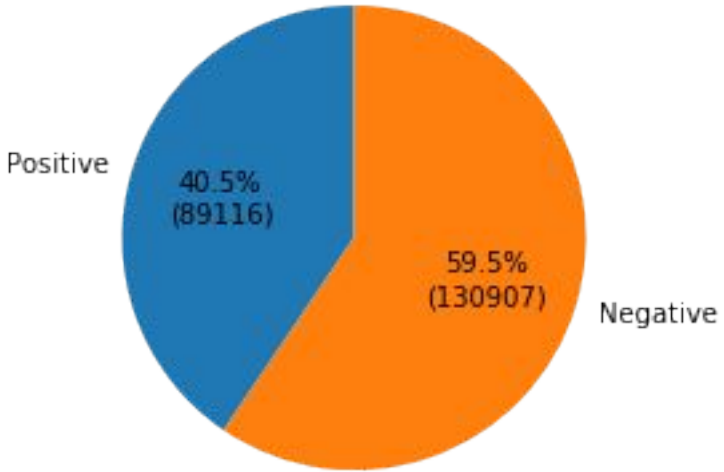
- The images and labels were downloaded from the PCam Github repository.
- The images were loaded into a python dictionary with their ID values as keys and 3-d numpy arrays as values.
- Their labels were loaded into a similar dictionary, IDs as keys and labels (1 or 0) as values.
- The dataset was clean as provided, it was only necessary to flatten the 3d arrays to 1 dimension for use in scikit-learn.

The Dataset (example)



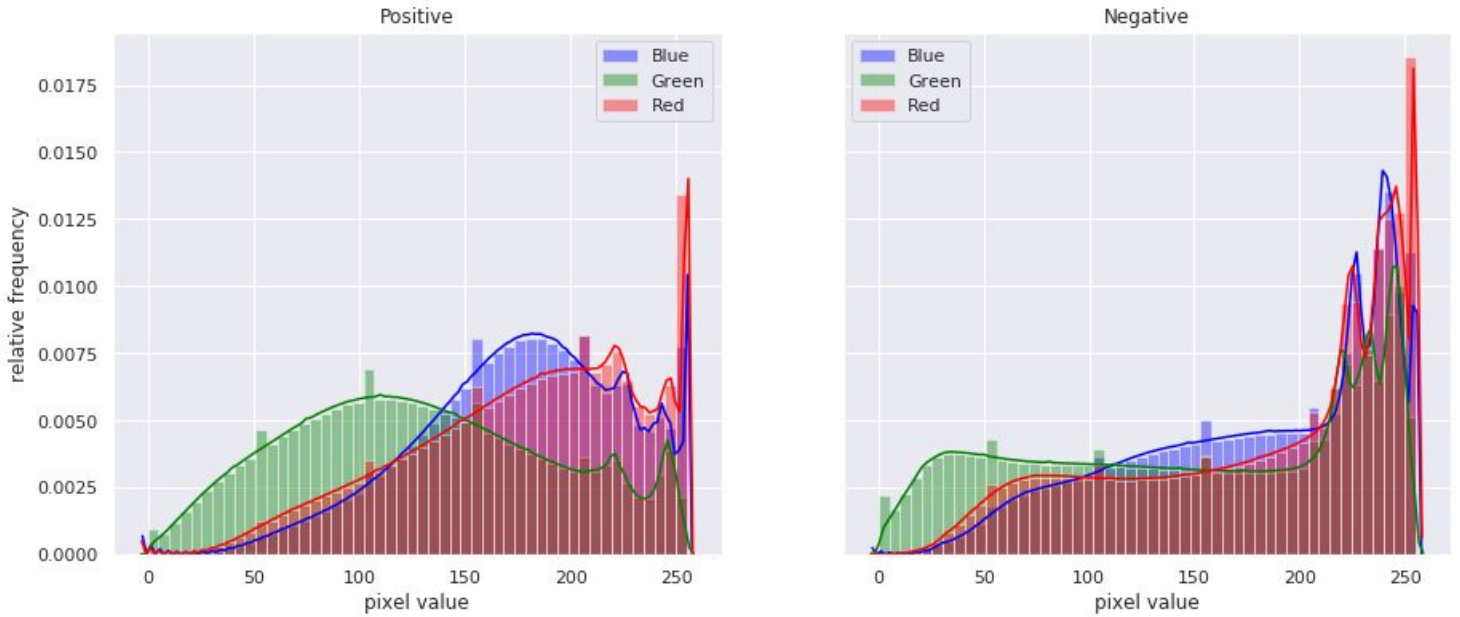
Data Storytelling and Inferential Statistics

- During the preliminary data exploration, a few key observations were made:
- The dataset is somewhat imbalanced, with 40.5% positive and 59.5% negative images. This imbalance will have some implications for model training.



Data Storyetlling (cont.)

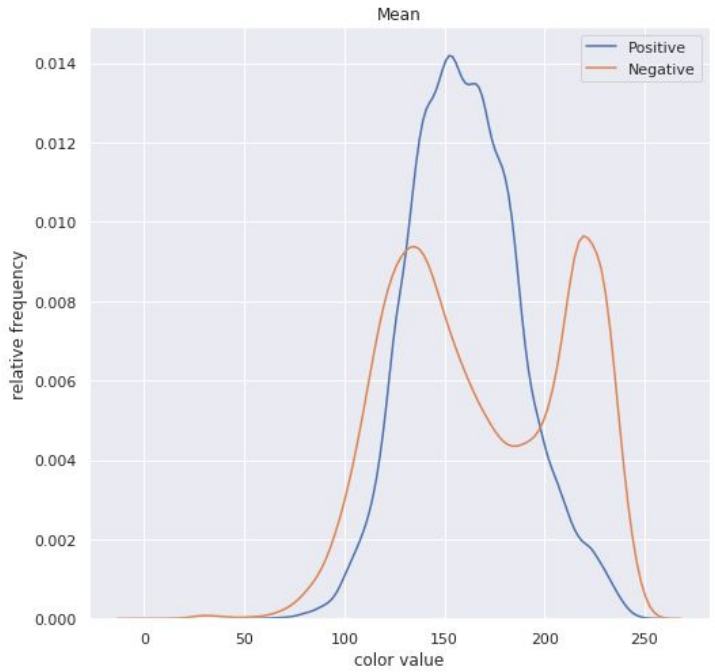
- The values of the color channels were examined, and there was a noticeable difference between the plots of the values of the positive and the negative sets:

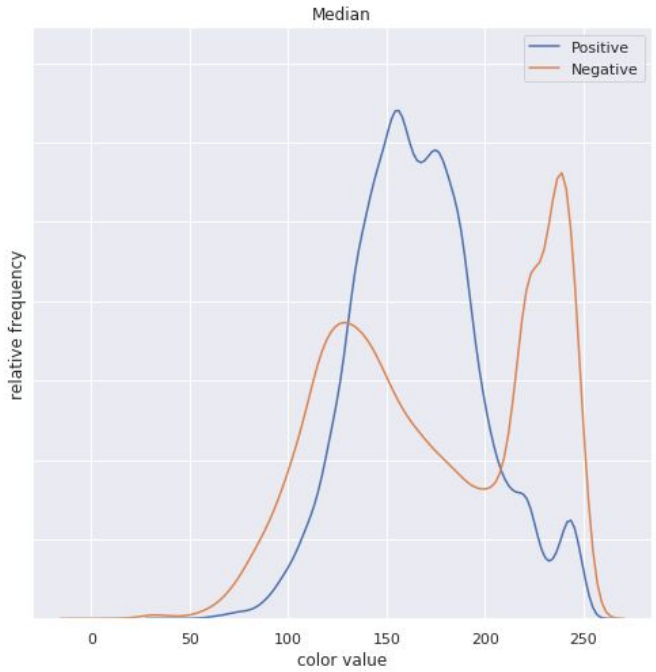


The relative frequencies of the color channel values.

Data Storytelling (cont.)

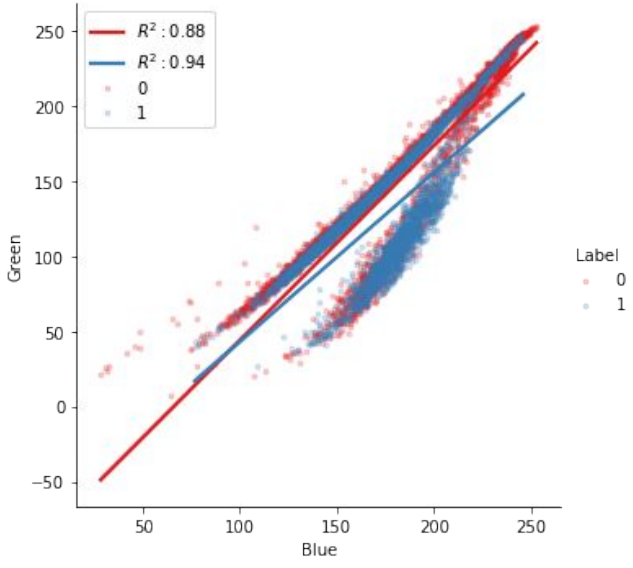
- There was also a distinct difference in the distribution of the aggregate values of the images (example below of the mean and median values for the image sets):

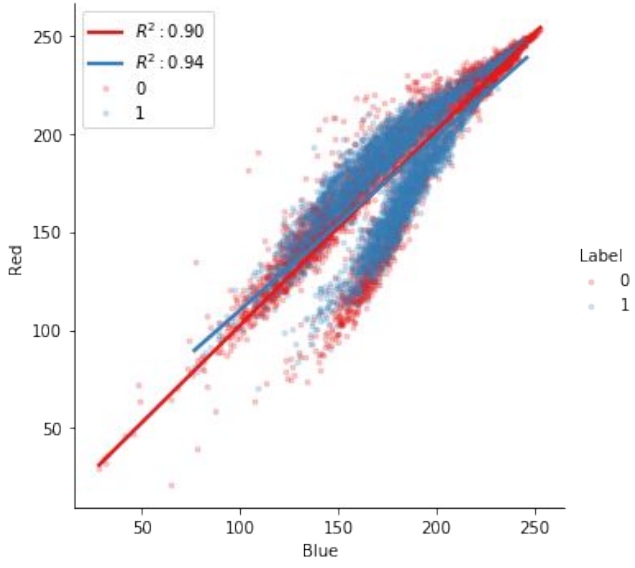


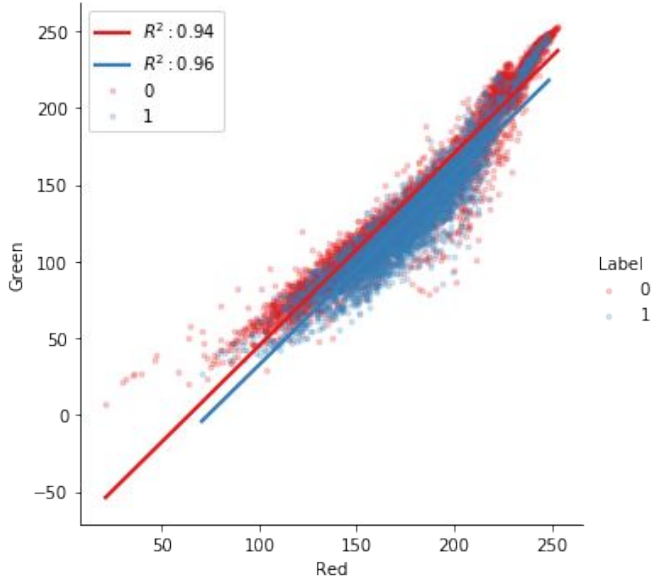


Inferential Statistics (cont.)

- It was also noted that there was a difference in the regression lines between positive and negative sets for each of the color groups, for instance:







Inferential Statistics (cont.)

- I observed that the difference between the positive and negative groups across the blue/red and blue/green planes is greater than across the red/green plane. An OLS Regression of the blue channel vs green and red channels combined shows a R^2 of 0.818 in the positive set and 0.93 in the negative set.

Baseline Modeling

- Scikit-learn was used for all modeling.
- For the baseline model, a Logistic Regression was chosen.
- The initial results were as follows:

Class	Precision	Recall	F1 Score	Support
0	0.72	0.83	0.77	32727
1	0.67	0.52	0.59	22280

Extended Modeling

- Random Forest and MLP neural net were attempted
- MLP Classifier failed but Random Forest showed some improvement over Logistic Regression

Class	Precision	Recall	F1 Score	Support
0	0.76	0.88	0.82	32727
1	0.77	0.59	0.67	22280

Extended Modeling (cont.)

- To address the data imbalance 2 strategies were compared:
- Random undersampling
- Image Augmentation

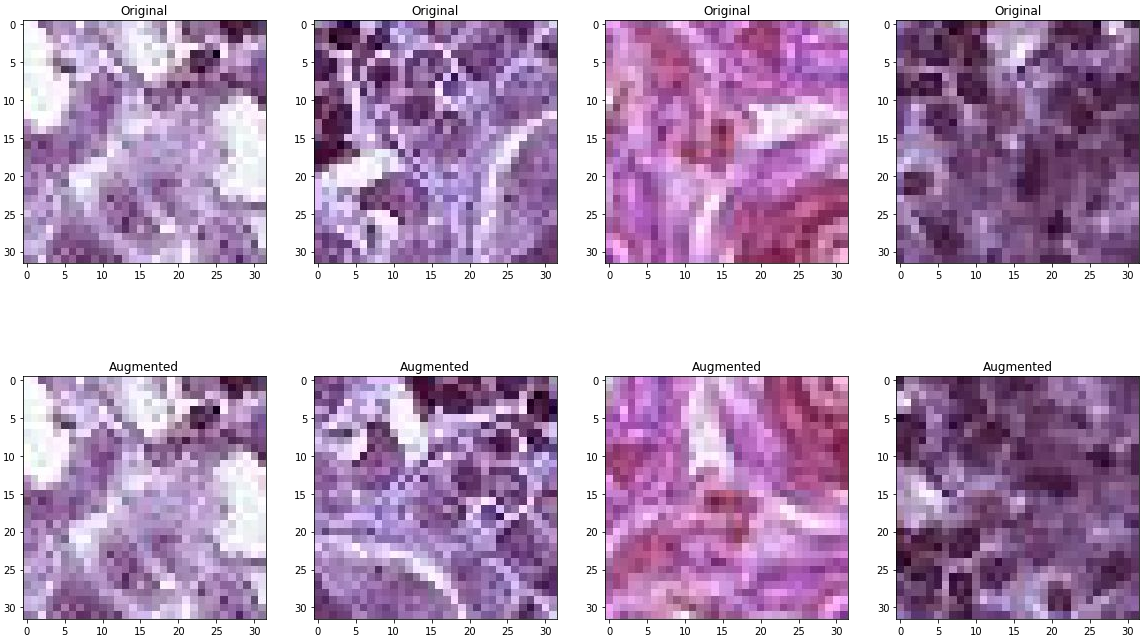
Extended Model (cont.)

- Random Undersampling
- Constructed an ensemble classifier
- Majority vote of several pipelines
- Pipelines contained random undersample of majority class followed by Random Forest classifier
- Improvement over the baseline model was observed:

Class	Precision	Recall	F1 Score	Support
0	0.80	0.81	0.81	32727
1	0.72	0.71	0.71	22280

Extended Model (cont.)

- Image Augmentation
- Random transformations to a random selection of images in the minority class were applied



Extended Model (cont.)

- Image Augmentation
- Improvement in model performance were observed similar to those resulting from random undersampling

Class	Precision	Recall	F1 Score	Support
0	0.80	0.80	0.80	32727
1	0.71	00.70	0.71	22280

Extended Model (cont.)

Hyperparameter Tuning

- Randomized Search CV over random forest parameters
- Modest Improvement

Class	Precision	Recall	F1 Score	Support
0	.081	.080	.081	32727
1	.071	.073	.072	22280

Findings

Model	Class	Precision	Recall	AUC	Support
Baseline Logistic Regression	0	0.72	0.83	0.76	32727
	1	0.67	0.52		22280
Baseline Random Forest	0	0.76	0.88	0.83	32727
	1	0.77	0.59		22280
Undersampling Ensemble	0	0.8.	0.81		32727
	1	0.72	0.71		22280
Image Augmentation	0	0.80	0.81	0.83	32727
	1	0.72	0.70		22280
Tuned Random Forest with Image Augmentation	0	0.81	0.80	0.84	32727
	1	0.71	0.73		22280

Findings (cont.)

- Balancing the dataset by either undersampling or oversampling yields similar improvement in this case.
- As image augmentation is less computationally costly, it should be preferred.
- Hyperparameter tuning shows only modest improvement

Conclusions and Future Work

- Balancing the dataset and using a random forest classifier yielded modest success in identifying samples with cancer and without.
- Results are still less than desired. Given a goal of AUC of .95, .82 was achieved.
- Further improvements may be achieved with hyperparameter tuning of the classifier
- Using a more sophisticated (deep learning) model may be the best way forward. e.g. Convolutional Neural Net

Reccomendations

- For any further modeling use a balanced dataset.
- Proceed with a deep learning model, as the best performance we can achieve in scikit-learn is still short of target.

Resources

- Scikit-Learn
- Imbalanced-Learn
- OpenCV
- Imgaug
- PCam
- Kaggle