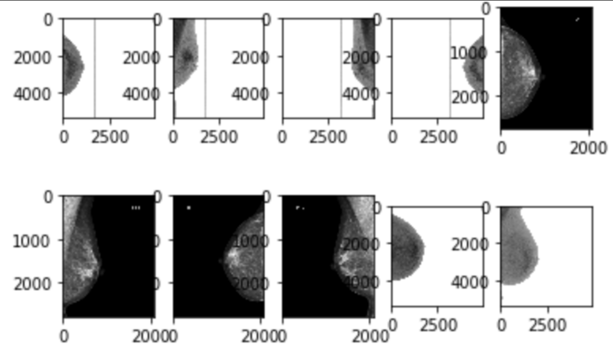
Mammography breast cancer detection

Phuc Nguyen

April 2023

- As many as half of women experience false positive mammography screening, leading to costly medical procedures.
- We would like to automate breast cancer detection using ML.

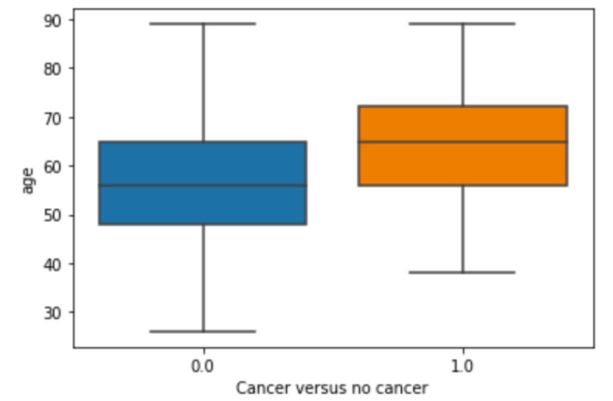
- Advantage: might improve false positive rate (FPR), thereby improving patient care.

- 

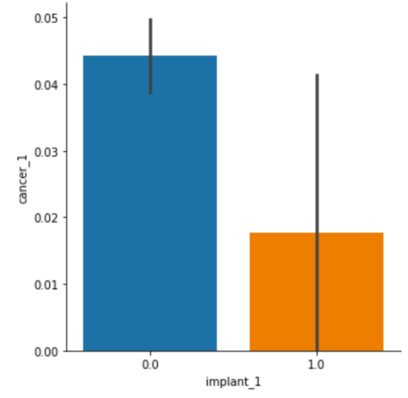
  2 types of mammograms: different backgrounds, different sizes.

- Converted to white background, size 512 x 512.

- Cleaned up view:
  - 'ML', 'LM', 'LMO' typos for 'MLO' (medio-lateral oblique)
  - 'AT' deleted
- Dropped rows with missing data
- One-hot encoding

- 

Median age is higher among patients with cancer.

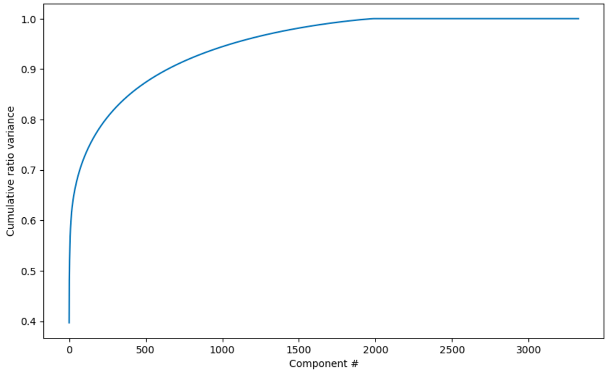
- 

Interestingly, cancer is less likely among patients with breast_implant !

- We found the distribution of breast_density to be:
  - Density A: 10 % of patients
  - Density B: 43 % of patients
  - Density C: 42 % of patients
  - Density D: 5 % of patients

  Same as breast density distribution in the general population.
- Mammograms evenly distributed between laterality L (49.8 %) and laterality R (50.2 %)
- Mammograms evenly distributed between view CC (48.6 %) and view MLO (51.4 %)

- Feature engineering with HOG (histogram of oriented gradients), instead of using pixel values directly.
- Undersampling and oversampling with SMOTE.
- Scaling with MinMaxScaler.
- PCA transformation with 100 components.

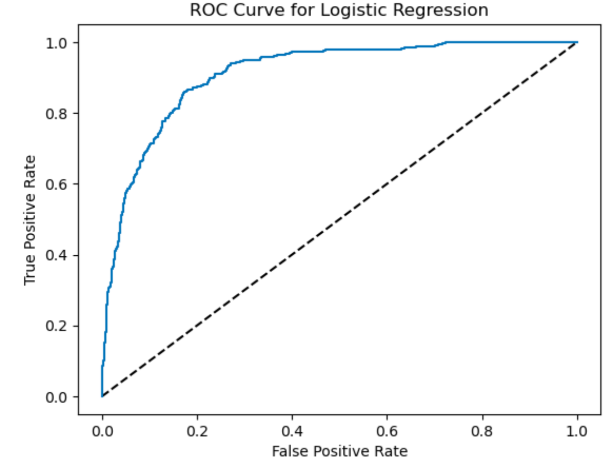


- 100 PCA components explain around 0.6-0.7 of the cumulative ratio variance

- Trained 3 models:
  - Logistic Regression
  - Random Forest
  - Boosted Gradient
- Performance metric: F1 score.
- Hyperparameter search with 5-fold cross-validation.

- Best model: Logreg
- Bayesian hyperparameter search for C over the range 0.01-200.
- F1-score on training data: 0.96
- F1-score on testing data: 0.64

- For random forest, we did hyperparameter search for min_samples_split, max_depth, criterion, max_features, bootstrap.
- F1-score on testing data only 0.26.
- For gradient boosting, we did hyperparameter search for learning_rate, min_samples_split, max_depth, criterion, max_features.
- F1-score on testing data only 0.35.

- 

  ROC-AUC for log reg: 0.91.
- The eblow occurs at a FPR below 0.25, which is what we want.

- By deploying the logistic regression model, we hope to bring down the FPR to below 25 percent within the next 5 years.
- If we had tried to use more images, Kaggle Kernel would have run out of memory. In the future, we would like to use more images.
- Use more PCA components ? Search hyperparameter more thoroughly ?
- Use neural nets ?