

# Bare Demo of IEEEtran.cls for IEEE Computer Society Journals

Michael Shell, *Member, IEEE*, John Doe, *Fellow, OSA*, and Jane Doe, *Life Fellow, IEEE*

**Abstract**—The abstract goes here.

**Index Terms**—Computer Society, IEEE, IEEEtran, journal, L<sup>A</sup>T<sub>E</sub>X, paper, template.

## 1 INTRODUCTION

## 2 METHODOLOGY

As the amount of generated per day grows at an exponential rate, brand new technologies have to be developed to cope up with the copious exabytes of data. Machine learning tools provide us with the capabilities to handle both structured and unstructured datasets. These tools can be configured to analyze patterns inherent in the data and make accurate predictions based on the information obtained. This concept is a reality for almost all sectors today. As per a 2020 Stanford study, the amount of healthcare data generated will be around 2,314 exabytes with a steady growth of 48%. The pipeline developed for this project has been depicted in Figure 1. The remainder of the section describes each individual step in detail.

### 2.1 Obtaining Raw data

In order to obtain distinct yet comparable subjects, a cohort dataset of 89 patients was selected in this study. The dataset consisted of four intrinsic molecular subtypes of breast cancer which are contrasted on the genes a cancerous cell expresses. The dataset has been described in Table 1.

For each of the patient, a CT scan was conducted to obtain cross-sectional images of the hypothesised tumor location. CT scans provide a more detailed description of

the patients condition by increasing the radiation level the patient is exposed to. Once the scan is completed three views are obtained namely, Axial, Sagittal and Coronal. DICOM (Digital Imaging and Communication in Medicine) images were obtained after the scan. For each patient 323 new studies were conducted with each study have 384 series which corresponded to 466 instances or images of the scan. Even though DICOM files are a standard format for medical imaging, NRRD (Nearly Raw Raster Data) files are anonymized and contain no sensitive patient information. Moreover NRRD store the entire information in a single file as opposed to DICOM imaging.

### 2.2 Convert to a suitable format

As mentioned previously, NRRD provides a more insightful approach to understanding medical imaging and recognizing inherent patterns in a concised format. The conversion was done with the help of the Plastimatch tool which is an open source software for image computation. Plastimatch takes the DICOM image which is described in a polyline vectorized format, and converts it into a series of pixels which is more prominently known as rasterization. The subroutine for rasterization of a DICOM image set with coordiantes  $x$  and  $y$  is shown below.

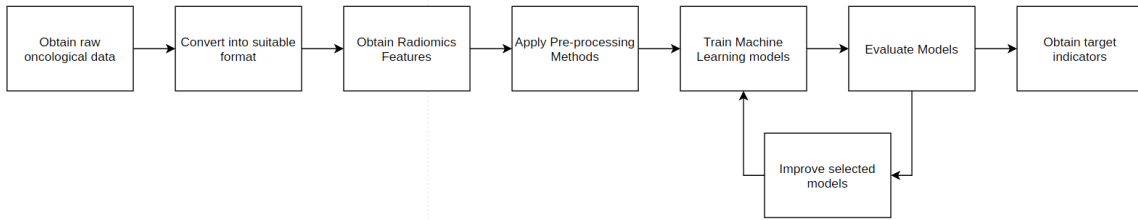


Fig. 1. Project Pipeline

TABLE 1  
Data Description

Subtype	Number of paitents	Estrogen Receptor	Progesterone Receptor	HER2	KI67 range
Luminal A	29	+	+/-	-	[5,20]
Luminal B	36	+	+/-	+/-	[25,80]
Triple Negative(TN)	19	-	-	-	[20,90]
HER	5	-	-	+	[30,50]

```
def rast(x, y, shape):
    nx, ny = draw.polygon(x, y, shape)
    nrrd = np.zeros(shape, dtype=np.bool)
    nrrd[ny, nx] = True

    return nrrd
```

Once this step is conducted, our image is in a compressed format, rife with information. Information extraction can be conducted through multiple means such as using neural networks, OCR recognition or pattern recognition algorithms.

### 2.3 Obtaining Radiomics Features

Information extraction from images directly has certain drawbacks. For eg, consider tumor classification using a standard Convolutional Neural Network (CNN). The CNN might be extremely successful in determining the existence of a blob of mass and its exact location. However diagnosing the exact nature and feature set of the tumor is extremely difficult for a CNN. This is because a CNN views the image as simply a collection of pixels without any regard to the information embedded in all the views of the data.

To tackle this issue, we have utilized radiomics algorithms to extract feature sets from the medical images to reveal characteristics which are not captured by trained networks. The open-source Python library, PyRadiomics was used to mine out the required feature set. Before the actual extraction could be performed, a set of filters were applied on the NRRD to provide a comprehensive view of the data. The filters applied are listed in table 2.

PyRadiomics obtains radiomics features from the CT scan results in a stagewise manner. Initially the images are loaded into the platform by using SimpleITK which supports a gamut of image types along with basic image processing techniques. In the next step, the filters described in 2 are applied using SimpleITK, PyWavelets, and Numpy. Finally, statistical and texture classes are used for feature extraction. The features so obtained, are stored in a dictionary format which suitable labels.

To define a Region of Interest (ROI) and to check the dimensional constraints of the data, a mask file is utilized. The mask file contains the tumor's location demarcated by a radiologist. The features extracted are described by the Imaging Biomarker Standardization Initiative (IBSI) and have been shown in tables 3 and 4.

Therefore for each patient, the total number of features obtained are number of filters  $\times$  number of features i.e,  $17 \times 100 = 1700$  features. Once the entire feature set has been collected, the classification task can be started.

### 2.4 Applying Pre-processing Techniques

From the 1700 features collected, not all of the features will contribute equally in the classification function. The process of preparing the input data for pattern learning by removing redundant characteristics, reducing noises and normalizing, selecting, and extracting features is termed as Data Pre-Processing. Multiple data pre-processing techniques have been applied to the feature set. These techniques have been described in Table 5.

Since the number of test subjects for each class is not similar, a threshold confidence level must be specified during the hypothesis testing phase. A 'P-value' is utilized in hypothesis testing to test the hypothesis under observation. A lower p-value corresponds to a higher confidence level in the predictions. The number of features selected after the pre-processing step is directly proportional to the p-value as a higher p-value will be more accommodating of even unimportant features. A grid for different p-values was created and the corresponding number of features were obtained.

### 2.5 Model-based Predictions

Once the features have been narrowed down, we use a flavour of Ensemble Learning called SFORCE to classify test subjects into the predicted classes. SFORCE establishes a symbiotic relation between a predictive model (Random Forest) and an Ensemble model (AdaBoost). Both these models work on the presented data simultaneously, aiding

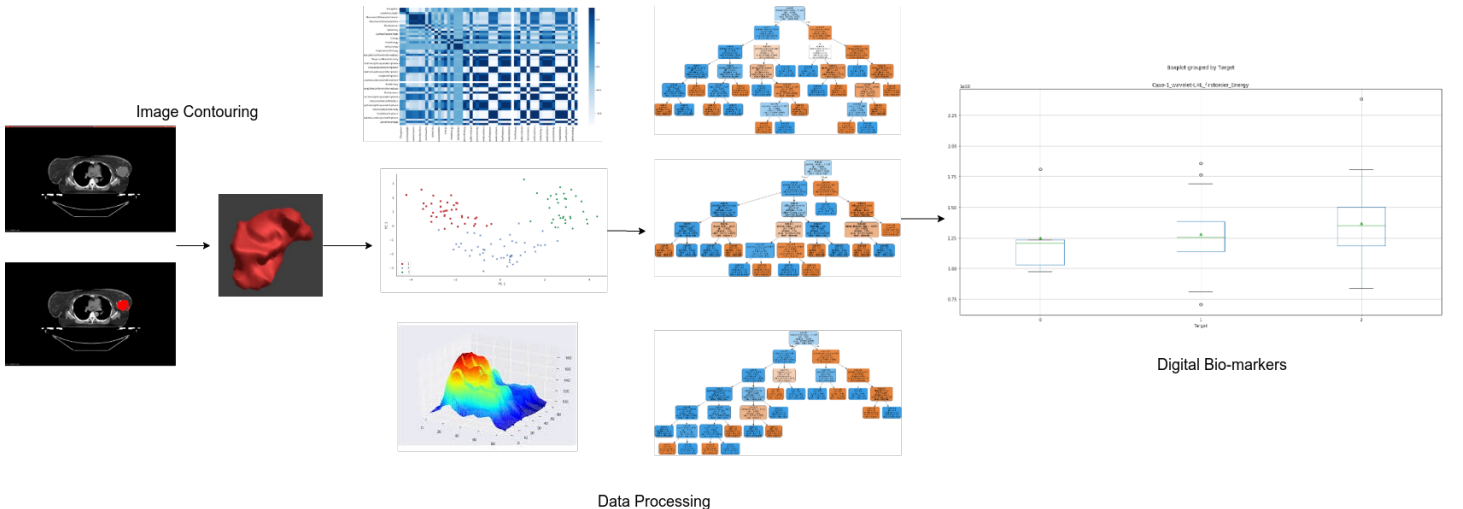


Fig. 2. untitled

each other in the prediction process. Random Forests provides a strong learning system with the occasional pitfall of overfitting. The data is classified based the features which contrast the classes with the highest information content. The process of data classification using Random Forest is shown in Algorithm 1. AdaBoost solves the problem of overfitting by presenting the system with the misclassified data and forcing it to improve the overall performance. The two flavours of AdaBoost i.e, SAMME and SAMME.R have

been described in Algorithms 2 and 3. SFORCE combines the strength of Random Forests and takes care of the drawbacks by using a Boosting algorithm to make the search process more concentrated as shown in Algorithm 4.

To obtain digital bio-markers, two cases studies were conducted from the available cohort dataset. The first study involved classifying test subjects as TN or non TN subjects. In the second study, the Luminal-B dataset was set aside as the test dataset due to the close resemblance of it's

TABLE 2  
Applied Filters

Filter	Description	Equation
Wavelet (9)	Selective emphasizing or de-emphasizing of image in selected spatial frequency domain	-
Square	Square the image intensities	$x := (cx)^2$
Square Root	Compute root of image intensities	$x := \sqrt{cx}$
Laplacian of Gaussian $\sigma = 1, 2, 3$	Applies a Laplacian of Gaussian filter to the input image and yields a derived image for each sigma value specified	$\frac{1}{(\sigma\sqrt{2\pi})^3} e^{-\frac{x^2+y^2+z^2}{2\sigma^2}}$
Logarithm	Computes the natural logarithm of image intensities	$\text{clog}(x + 1)$
Exponential	Computes the exponential of the original image	$e^{cx}$
Gradient	Computes the gradient of the image	-

TABLE 3  
Features-I

Feature Class	Feature	Feature Class	Feature
Shape	Elongation	Grey Level Co-occurrence Matrix	Autocorrelation
	Flatness		Cluster_Prominence
	Least_Axis_length		Cluster_Shade
	Major_Axis_Length		Cluster_Tendency
	Max_2D_Diameter_Column		Contrast
	Max_2D_Diameter_Row		Correlation
	Max_2D_Diameter_Slice		Difference_Average
	Max_3D_Diameter		Difference_Entropy
	Mesh_Volume		Difference_Variance
	Minor_Axis_Length		Inverse_Variance
	Sphericity		Joint_Average
	Surface_Area		Joint_Energy
	Surface_Volume		Joint_Entropy
	Voxel_Volume		MCC
			Maximum_Probability
First Order Statistics	10 Percentile		Sum_Average
	90 Percentile		Sum_Entropy
	Energy		Sum_Squares
	Entropy		Id
	Interquartile_Range		Idm
	Kurtosis		Idn
	Maximum		Idmn
	Mean_Absolute_Deviation		Imc1
	Mean		Imc2
	Median	Grey Level Run Length Matrix	Uniformity
	Minimum		Normalized_Uniformity
	Range		Variance
	Robust_Mean_Deviation		High_Run_Emphasis
	Robust_Mean_Squared		Long_Run_Emphasis
	Skewness		Long_High_Run_Emphasis
	Total_Energy		Long_Low_Run_Emphasis
	Uniformity		Low_Run_Emphasis
	Variance		Run_Entropy
			Run_Uniformity
			Run_Uniformity_Normalized
			Run_Percentage
			Run_Variance
			Short_Run_Emphasis
			Short_Run_High_Emphasis
			Short_Run_Low_Emphasis

characteristics with those of Luminal A. The model was trained to place the test subjects into the Luminal-A class with an accuracy of 72.7%. The results for different p-values have been described in Tables 6 and 7. Based on these results, box-plots have been obtained for the selected features which act as bio-markers for future reference.

## REFERENCES

- [1] H. Kopka and P. W. Daly, *A Guide to L<sup>A</sup>T<sub>E</sub>X*, 3rd ed. Harlow, England: Addison-Wesley, 1999.

TABLE 4  
Features-II

Feature Class	Feature
Grey Level Size Zone Matrix	Non_Uniformity
	Non_Uniformity_Normalized
	Variance
	High_Zone_Emphasis
	Large_Area_Emphasis
	Large_Area_High_Level_Emphasis
	Large_Area_Low_Level_Emphasis
	Low_Zone_Emphasis
	Zone_Non_Uniformity
	Zone_Non_Uniformity_Normalized
	Small_Area_Emphasis
	Small_Area_High_Level_Emphasis
	Small_Area_Low_Level_Emphasis
	Zone_Entropy
	Zone_Percentage
	Zone_Variance
Gray Level Size Zone Matrix	Dependence_Entropy
	Dependence_Non_Uniformity
	Dependence_Non_Uniformity_Normalized
	Dependence_Variance
	GL_Non_Uniformity
	GL_Variance
	High_Emphasis
	Large_Dependence_Emphasis
	Large_Dependence_High_Emphasis
	Large_Dependence_Low_Emphasis
	Low_Emphasis
	Small_Dependence_Emphasis
	Small_Dependence_High_Emphasis
	Small_Dependence_Low_Emphasis
Neighbouring Gray Tone Difference Matrix	Busyness
	Coarseness
	Complexity
	Constrast
	Strength

TABLE 5  
Preprocessing techniques

Method	Description
Missing Value Ratio	Removal of data columns where the ratio of missing values is greater than a set threshold
Low Variance Filter	Removal of normalized data columns where the variance is lesser than a set threshold
Highest correlation filter	Removal of data columns which are highly correlated leading to redundancy
Principle Component Analysis	Transformation of data to maximize variance under constraints
Fast Independent Component Analysis	Decomposition of signals to focus on mutual independence of data
Factor Analysis	Generating a common feature by reducing number of common variables

TABLE 6  
TN vs Non-TN

P-Value	Number of features	SAMME Accuracy	SAMME.R Accuracy
1	21	81.25	90.39
0.5	16	90.39	93.25
0.1	6	75	81.25

**Algorithm 1 : Ensemble Learning: Random Forest**


---

```

1: // Input: Data Set  $D = \{(x_1, y_1), (x_2, y_2), \dots, (x_m, y_m)\}$ , Feature Set
  F, Randomization Factor R, Number of trees T
  // Output: Root node of  $i^{\text{th}}$  tree
2: -----
3: for  $\forall i \in \{1, 2, \dots, T\}$  do
4:    $N_i \leftarrow$  Root node of  $i^{\text{th}}$  tree
5:   if All targets belong to same class i.e  $y_i$  or  $F \in \emptyset$  then
6:     Return  $N_i$ 
7:   end if
8:    $D_i \leftarrow$  bootstrapped sample from D
9:   for Each node do
10:     $f \leftarrow$  Randomly selected  $R$  features from  $F$ 
11:     $N_f \leftarrow$  Best Feature from  $f$  features
12:     $N_p \leftarrow$  Best Split based on  $N_f$ 
13:   end for
14: end for
15: return  $N_i$ 

```

---

**Algorithm 2 : Stagewise Additive Modeling: SAMME**


---

```

1: // Input: Data Set  $D = \{(x_1, y_1), (x_2, y_2), \dots, (x_m, y_m)\}$ , Number
  of Learning Rounds T, Learning Algorithm  $\epsilon$ 
2: // Output:  $\text{sign}(\sum_{t=1}^T \alpha_t C_t)$ 
3: -----
4:  $D_1(x) = 1/m$  {Initialize the weight distribution}
5: for  $t = \{1, 2, \dots, T\}$  do
6:    $C_t = \epsilon(D, D_t)$  {Create classifier  $C_t$ }
7:    $e_t = P_{x \sim D}(h_t(x) \neq f(x))$  {Calculate error  $e_t$ }
8:    $\alpha_t = \log \frac{1 - e_t}{e_t} + \log(K-1)$  {Calculate the weight  $h_t$ }
9:    $D_i(x) \leftarrow D_i(x) \cdot \exp(\alpha_t \cdot P(C_i \neq f(x)))$  {Update the distribution
     $D_t\}$ ,  $i = \{1, 2, \dots, m\}$ 
10:   Renormalize  $D_t(x)$ 
11: end for

```

---

**Algorithm 3 : Stagewise Additive Modeling for Real Value Predictions: SAMME.R**


---

```

1: // Input: Data Set  $D = \{(x_1, y_1), (x_2, y_2), \dots, (x_m, y_m)\}$ , Number
  of Learning Rounds T, Learning Algorithm  $\epsilon$ 
2: // Output:  $\text{sign}(\sum_{t=1}^T \alpha_t C_t)$ 
3: -----
4:  $D_1(x) = 1/m$  {Initialize the weight distribution}
5: for  $t = \{1, 2, \dots, T\}$  do
6:    $C_t = \epsilon(D, D_t)$  {Create classifier  $C_t$ }
7:    $p_{kt}(x) = \text{Prob}(y = k|x)$ ,  $k = \{1, 2, \dots, K\}$ 
8:    $h_{kt}(x) \leftarrow (K-1)(\log p_{kt}(x) - \frac{1}{K} \cdot \sum_{k'} \log p_{k't}(x))$ 
9:    $D_i(x) \leftarrow D_i(x) \cdot \exp(\frac{1-K}{K} \cdot y_i^T \cdot \log(p_t(x_i)))$  {Update the distri-
    bution  $D_t$ ,  $i = \{1, 2, \dots, m\}$ 
10:   Renormalize  $D_t(x)$ 
11: end for

```

---

**Algorithm 4 : Ensemble of Ensemble: SFORCE**


---

```

1: // Input: Data Set  $D = \{(x_1, y_1), (x_2, y_2), \dots, (x_m, y_m)\}$ , Feature Set
  F, Randomization Factor R, Number of trees T, Number of Learning
  Rounds  $T'$ , Learning Algorithm  $\epsilon$ 
2: // Output: Root node of  $i^{\text{th}}$  Boosted Tree
3: -----
4: Random Forest
5: for  $\forall i \in \{1, 2, \dots, T\}$  do
6:    $N_i \leftarrow$  Root node of  $i^{\text{th}}$  tree
7:   if All targets belong to same class i.e  $y_i$  or  $F \in \emptyset$  then
8:     Call SAMME.R with  $N_i$ 
9:   end if
10:   $D^i \leftarrow$  bootstrapped sample from D
11:  for Each node do
12:     $f \leftarrow$  Randomly selected  $R$  features from  $F$ 
13:     $N_f \leftarrow$  Best Feature from  $f$  features
14:     $N_p \leftarrow$  Best Split based on  $N_f$ 
15:    Call SAMME.R with  $N_i$ 
16:  end for
17: end for
18: return  $N_i$ 
19: -----
20: SAMME/SAMME.R
21:  $D_1(x) = 1/m$  {Initialize the weight distribution}
22: for  $t = \{1, 2, \dots, T\}$  do
23:    $C_t = \epsilon(D, D_t)$  {Create classifier  $C_t$ }
24:    $p_{kt}(x) = \text{Prob}(y = k|x)$ ,  $k = \{1, 2, \dots, K\}$ 
25:    $h_{kt}(x) \leftarrow (K-1)(\log p_{kt}(x) - \frac{1}{K} \cdot \sum_{k'} \log p_{k't}(x))$ 
26:    $D_i(x) \leftarrow D_i(x) \cdot \exp(\frac{1-K}{K} \cdot y_i^T \cdot \log(p_t(x_i)))$   $\{i = \{1, 2, \dots, m\}\}$ 
27:   Renormalize  $D_t(x)$ 
28:   Call Random Forest with  $(\sum_{t=1}^{T'} \alpha_t C_t)$ 
29: end for

```

---

**Algorithm 5 untitled**


---

```

1: //Input Image dataset  $D_n$  and masks  $D_m$ 
2: //Output Predicted Class
3: for Each image  $i$  in  $D_n$  do
4:   Convert image to a suitable format using conversion software
5:   Call the pre-processing techniques on the formatted images
6:   Using mask  $j$  for corresponding  $i$ , extract radiomics features
7:   Create a grid of p-values
8:   for EACH value in grid do
9:     Call Algorithm 4 with related feature set
10:   end for
11:   Obtain accuracy levels and digital bio-markers
12: end for

```

---

TABLE 7  
HER vs Luminal-A vs TN

P-Value	Number of features	SAMME Accuracy	SAMME.R Accuracy
1E-5	17	72	63.63
1E-6	15	70	72.7
17-5	13	72.7	70