Using Machine Learning Classification Techniques on Breast Cancer Dataset

Phalguni Rathod, Student Number: R00183770, PML Assignment 2

Abstract In this project we are building end-to-end machine learning model on breast cancer dataset[1] to classify each data point into malignant or benign based on features given. We are starting by preprocessing the data using techniques like outlier detection (box plots), looking for missing values, checking categorical data, scaling and handling imbalance. Then we are applying variety of classification algorithms and comparing them to select top 3 algorithms. We will be tuning the hyperparameters of top algorithms and calculating the score. Further, we have chosen, handling imbalance data as our research area to understand alternative ways to deal with it.

1 Introduction

Breast cancer is one of the most frequently occurring cancer in women. It has the second highest mortality rate. The good news is that the mortality rate from breast cancer has progressively and steadily declined over the years [2]. Diagnosis of breast cancer is performed when an abnormal lump is found (from self-examination or x-ray) or a tiny speck of calcium is seen (on an x-ray). After a suspicious lump is found, the doctor will conduct a diagnosis to determine whether it is cancerous and, if so, whether it has spread to other parts of the body [1]. We have taken data of cancerous lumps obtained from the University of Wisconsin Hospitals, Madison from Dr. William H. Wolberg. The data taken is clean data having 569 data points. The data consist of only numeric values. And it doesn't have any outliers and missing values but it's highly imbalanced, approximately 37% - 63%. The dataset consists of 5 features and a target.

1. **mean_radius:** mean of distances from center to points on the perimeter
2. **mean_texture:** standard deviation of gray-scale values
3. **mean_perimeter:** mean size of the core tumor
4. **mean_area:** mean area of the core tumor
5. **mean_smoothness:** mean of local variation in radius lengths
6. **diagnosis:** The diagnosis of breast tissues (0 = malignant, 1 = benign)

This dataset has been work upon rigorously for various problem statements. Specific to classification many papers are published comparing various algorithms under a number of conditions.

2 Research

The breast cancer data is skewed towards the classifying the datapoint as benign (1), and it's about 63% of the data. While 37% of data points are classified as malignant (0). Along with this issue, it is also observed that the data is very small, i. e. just 569 rows.

When the data is imbalance and we are not using proper evaluation metric, it might show us over optimistic results, e.g : when using accuracy it might give us result high accuracy in favour of the majority class.

There are various techniques of handling data imbalance broadly saying, 2 major approaches are : Over-sampling and Under-sampling.

1. Over Sampling : Here, the data points from the minority class is randomly duplicated to balance with majority class. This is generally used when dataset is small and losing data points will lead to smaller dataset. While the con of using this is that the model can be overfitting.

2. Under Sampling : Here, the data points from the majority class is randomly removed to balance with minority class. This is generally used when dataset is huge and removing data points will not create much effect on the expected output.

As we observed that our dataset is very limited and small, hence, doing under sampling is out of the option.

Hence, we are researching various ways to over sample the data. One of the ways is to use SMOTE (Synthetic Minority Oversampling Technique) that is used in data preprocessing step ahead.

2.1 Techniques for Over Sampling

As we are using SMOTE for over sampling in preprocessing steps, it is a good idea to use variants of SMOTE to research their effect and analyse them. We have shortlisted three variants: Borderline Smote, K Means Smote and SVM Smote.

2.1.1 Borderline SMOTE

This variant also work on the similar lines of SMOTE i.e by finding the nearest neighbours using KNN but the only difference is that, it focuses on borderline data and over sample them. It is such, because borderline data points are more apt to be misclassified than the ones far from the borderline. [5]
Not like the existing over-sampling methods, our methods only oversample or strengthen the borderline minority examples. First, we find out the border-line minority examples; then, synthetic examples are generated from them and added to the original training set. [5]

2.1.2 K Means SMOTE

K-Means SMOTE is an oversampling method for class-imbalanced data. It aids classification by generating minority class samples in safe and crucial areas of the input space. The method avoids the generation of noise and effectively overcomes imbalances between and within classes. [4]

K-means SMOTE works in three steps:
1. Cluster the entire input space using k-means
2. Distribute the number of samples to generate across clusters:
   i. Filter out clusters which have a high number of majority class samples.
   ii. Assign more synthetic samples to clusters where minority class samples are sparsely distributed.
3. Oversample each filtered cluster using SMOTE [4]

2.1.3 SVM SMOTE

SVM-SMOTE focuses on generating new minority class instances near borderlines with SVM so as to help establish boundary between classes.[6]
It is in line with borderline SMOTE by replacing KNN with SVM.

3 Methodology

Using machine learning to solve a problem is not just about taking data and piping through algorithms. There are a number of steps taken to make sure everything works fine. These steps start with data preprocessing where we deal with inconsistencies in data and try to make it as consistent as possible. Post this, we either split the data in train and test, or use cross validation techniques for the same. Once we have our train-test dataset ready, we are ready to pipe them through model (we are using pre-defined models from sklearn package). We can evaluate the output on various parameters like accuracy, precision, recall, f1-score or, simply, confusion matrix. Yet for the purpose of this project, we will be using f1-score as the data is skewed. Once done with everything we are then going to select top 3 models based on scoring (f1-score) and tune their hyper-parameters.

3.1 Data Preprocessing

3.1.1 Detecting Outliers

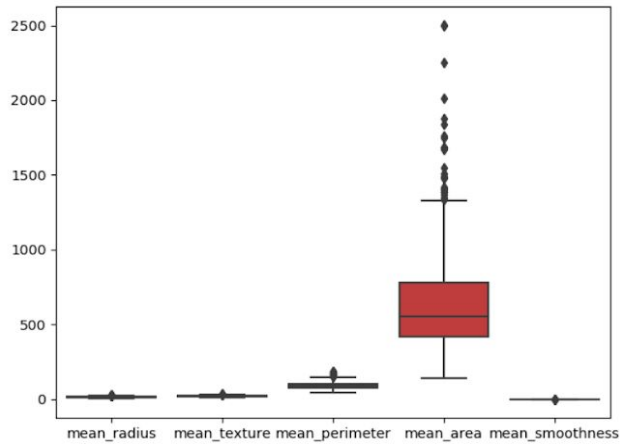
Outlier is a data point which differs a lot from the majority of data sample. There could be many reasons for a data point to be outlier some of them can be:

1. Data transcription may be erroneous due to human or machine error
2. Rarest possibility of that event/data-point to occur.

To check the existence of outliers in dataset we typically use box plot visualization which shows the data points lying outside of 1.5 IQR. Data outside 1.5 IQR doesn't always have to be outlier sometimes when data which

isn't sparse rather it's clustered together it becomes a really important data and can impart crucial properties too. While using box plot[Fig 1] on breast cancer data we saw that a lot of points are lying outside the whiskers yet they all seemed to be clustered together. And it's similar to above discussed exception. Hence, we didn't remove ant data-point as outlier.

Fig 1



3.1.2 Missing Values

The data obtained is totally clean with no missing values. Hence, we simply checked the existence of null values in each column to verify the authenticity of the clean dataset and it turns out to be true.

3.1.3 Handling Categorical Data

The breast cancer dataset is only consisting of numerical columns of various measurements of lump in breast. Hence, we didn't have to deal with categorical data.
In any case, to deal with categorical data we can convert it into numeric columns by using one-hot encoding or label encoder.

3.1.4 Scaling Data

As it's visible from box-plot above, values like area is at very different scale then values like smoothness. For ML algorithms to work better & efficiently we have to bring all the numeric values on the same scale and this is known as scaling. The majority of ML and optimization algorithms behave better if features are on the same scale.[Lecture Slides] There are 2 techniques:

1. Normalization: Data is rescaled in the range of 0 to 1.
2. Standardization: It facilitates the transformation of features to a standard Gaussian(normal) distribution with a mean of 0 and a standard deviation of 1.

We are using standardization technique for scaling our data, reason being that it is not so sensitive to outlier and as we didn't remove any outlier it is good to use this technique to avoid unforeseen effect due to non-removal of outliers.

3.1.5 Handling Data Imbalance

As mentioned, our dataset is very small, under sampling is not an option for us. Hence, we are using an over sampling technique called SMOTE.
SMOTE stands for Synthetic Minority Oversampling Technique. It is recursive flow of certains steps :

1. Selection of datapoint from minority class
2. Finding nearest neighbours using knn
3. Picking up one of the neighbours at random

4. Take difference of the selected data points and multiply with random number between 0 & 1.
5. The data created is our new sample of minority class.
6. Continue until balance achieved.

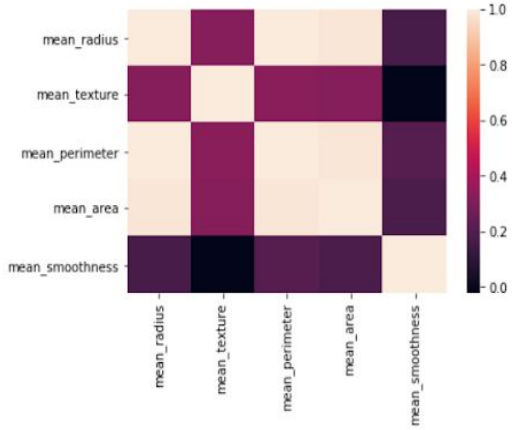
3.1.6 Feature Selection

It is the process of identifying the interdependency among each feature and target. The feature which is more relevant for prediction of the classes are selected. There are various ways of feature selection. Yet we often start with finding correlation among features and move ahead.

Fig 2

	mean_radius	mean_texture	mean_perimeter	mean_area	mean_smoothness
mean_radius	1.000000	0.323782	0.997855	0.987357	0.170581
mean_texture	0.323782	1.000000	0.329533	0.321086	-0.023389
mean_perimeter	0.997855	0.329533	1.000000	0.986507	0.207278
mean_area	0.987357	0.321086	0.986507	1.000000	0.177028
mean_smoothness	0.170581	-0.023389	0.207278	0.177028	1.000000

```
In [5]: sns.heatmap(corr)
Out[5]: <matplotlib.axes._subplots.AxesSubplot at 0x2ba5e985400>
```



From the correlation matrix and heatmap visualization [Fig 2] we can say that mean_radius, mean_perimeter and mean_area are highly correlated. Yet we can't reduce our feature space to just three features. Hence, we are not extracting these features for classification.

3.2 Baseline Models

We are using a number of classification algorithms to find the best model which gives the maximum score for the breast cancer dataset. We have undertaken basic algorithms to ensembles to find the best possible model for the dataset. For all the algorithms considered we are running them on default parameters.

1. KNN
2. Logistic Regression
3. Gaussian Naïve Bayes
4. SVM
5. Random Forest
6. Decision Tree
7. Ada Boosting
8. Gradient Boosting

For the evaluation of these models we have taken a common evaluation metric: F1- score. The models generated are the baseline models. We will select the top 3 best models based on scores and tune them on their hyper parameters.

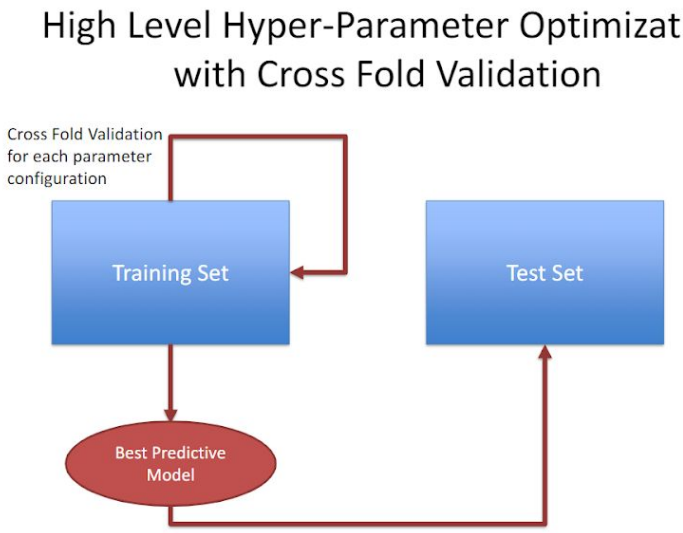
In our range of models, we detected the best performing models with highest f1-scores are:

1. Logistics Regression : 0.9454841049024744
2. Ada Boosting: 0.9394529849864597
3. Gradient Boosting: 0.9395901972234133

3.3 Hyper-Parameter Optimization

These are the parameters which are not learned or automatically adjusted by algorithms and needs to be experimented with a variety of values to get the best possible result. These parameters are tuned manually over range of values and then compared to get the best set of parameters having highest performance score.

Fig 3



The above illustrations[Fig 3] show the flow used for hyper-parameter selection.

We are using GridSearchCV[7] for the same purpose. Here we pass param_grid dictionary, which have parameters as keys and range of values for parameters as values.
We have taken no_of_splits as 10 because the dataset is small and we need more data for training, so, with the increase in number of splits the training data increases. The way is to use leave-one-out CV but it is computationally expensive, hence, we are using this.
As our data is imbalance we have to use SMOTE on training data along with the top algorithms and make a pipeline which inturn is passed to the GridSearchCV. Again for evaluation we are using F1-Score.

3.4 Parameter Tuning & Best Performing Model

3.4.1 Logistic Regression

1. **C:** Inverse of regularization strength; must be a positive float. Smaller values specify stronger regularization.[3]. **Range**: np.logspace(0, 4, 10)
2. **penalty:** Used to specify the norm used in the penalization. **Range:** [‘l1’,’l2’]

3.4.2 Ada Boosting

1. n_estimators: The maximum number of estimators at which boosting is terminated. In case of perfect fit, the learning procedure is stopped early.
Range: list(range(50, 101))
2. learning_rate: Learning rate shrinks the contribution of each classifier by learning_rate.
Range: [0.1, 0.3, 0.5, 0.7, 0.9, 1]

3.4.3 Gradient Boosting

1. n_estimators: The number of boosting stages to perform. Gradient boosting is fairly robust to overfitting so a large number usually results in better performance.
Range: list(range(100, 200))
2. learning_rate: Learning rate shrinks the contribution of each tree by learning_rate.
Range: [0.01, 0.03, 0.05, 0.07, 0.09, .1]
3. max_depth: The maximum depth limits the number of nodes in the tree. Tune this parameter for best performance; the best value depends on the interaction of the input variables.
Range: [1, 3, 5, 7, 9]

4. Evaluation

Machine learning algorithms are incomplete until and unless they are properly evaluated. The most crucial part of preparing end-to-end ML model is deciding the right evaluation metric.

There are a number of metrics available to judge the algorithm. The selection of evaluation metric depends on the data and how we are processing it. Some of the frequently used evaluation metrics are:

1. Accuracy: We compare the predicted target values with actual labels and find out what percentage of values are correctly predicted.

2. Confusion Matrix: It gives us deeper insight of the predicted & actual classes. It tells us about how many True Positives, True Negative, False Positives & False Negatives are found in the prediction.

3. Recall: It shows us how confident we are that all of the instances that belong to a specific class have been classified correctly using our model.

4. Precision: It shows us how confident we are that all any instance predicted as belonging to a certain class actually belongs to that class.

5. F1-Score: Rather than having Recall & Precision as 2 different metrics, they can be combined to form the F1-Score. It is harmonic mean of precision & recall. For F1 to be high both precision & recall has to be high. F1 doesn't get skewed on just a single value.

4.1 Selecting Evaluation Metric

4.1.1 Why not to use Accuracy?

Accuracy hides lot of details for basic classification. For example, an ML algorithm might be doing well are predicting one class but may be poor are predicting another class. This type of behaviour can often be masked when simply looking at classification accuracy.

Accuracy Paradox: When dealing with highly imbalanced datasets. The accuracy of the model will appear high but the model is just predicting the majority class. The situation is often referred to as the accuracy paradox. It occurs when your algorithm reports a very high level of accuracy (such as 95%), but the accuracy is only reflecting the class distribution within the dataset. [Lecture Slides]

4.1.2 What to use, Precision, Recall, F1-Score?

Precision: It will give us very low value when algorithm is not correctly predicting the values to be of actual class.

Recall: It will highlight if we do very poorly on predicting any specific class. In the imbalanced dataset it would clearly show that if our algorithms perform very poorly on minority class [Video Lecture].

F1-Score: When dealing with imbalanced data we actually need both precision & recall for evaluation but rather than using 2 different metrics, we considered using F1-Score because we need a model having high precision & recall and F1 score works on similar lines, i.e F1 is high only when both are high.

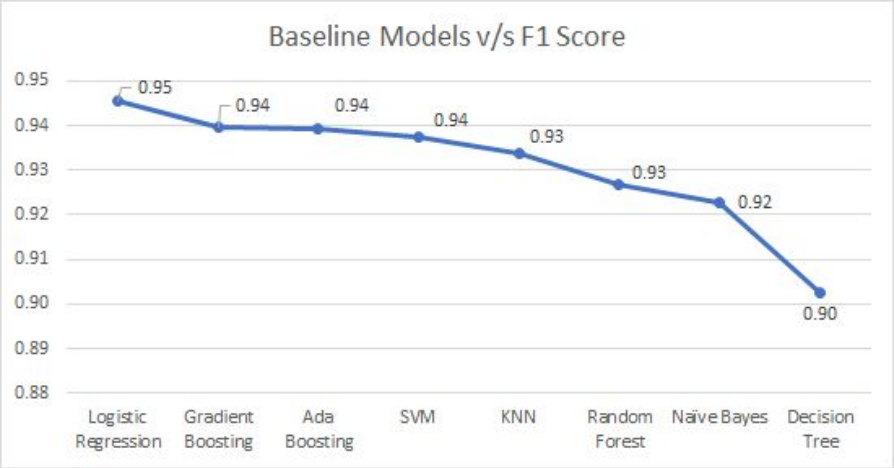
Alternatively, we can also use classification report where we get all three i.e precision, recall, & F1-Score.

We chose F1-Score only for ease of working through.

4.2 Baseline Models

We are evaluating our models on F1-Score over 10 iterations of Stratified K Fold Cross Validation and then we are averaging the F1 Score over no. of iterations.

Fig 4



Observation:
We can observe[Fig 4] that Logistic Regression, Gradient Boosting and Ada Boosting are the top performing models for breast cancer dataset. While Decision is the least performing algorithm.

4.3 Hyper Parameter Optimization on Top Models

1. Logistic Regression
Best Parameters: {'model__C': 464.15888336127773, 'model__penalty': 'l1'}
Best Score: 0.9367155095052364

2. Ada Boosting
Best Parameters: {'model__learning_rate': 0.5, 'model__n_estimators': 83}
Best Score: 0.9386796482294708

3. Gradient Boosting
Best Parameters: {'model__learning_rate': 0.9, 'model__n_estimators': 90}
Best Score: 0.9409046592076727

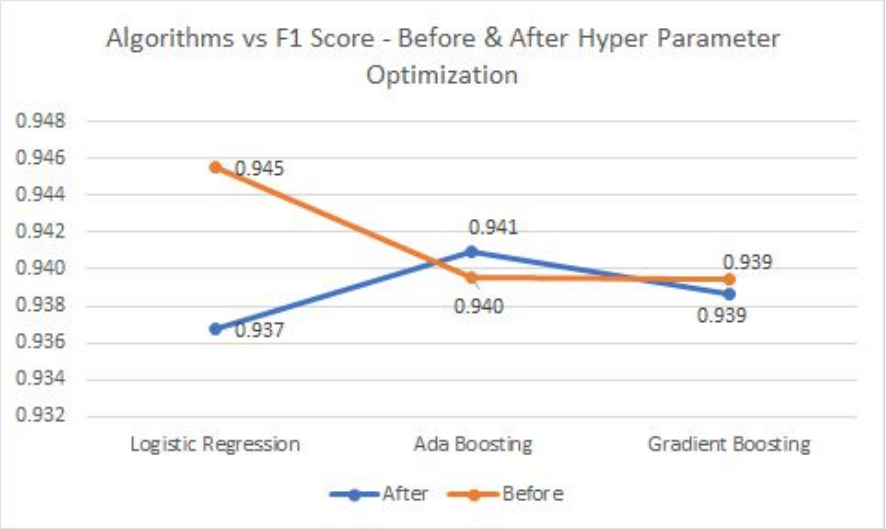


Fig 5

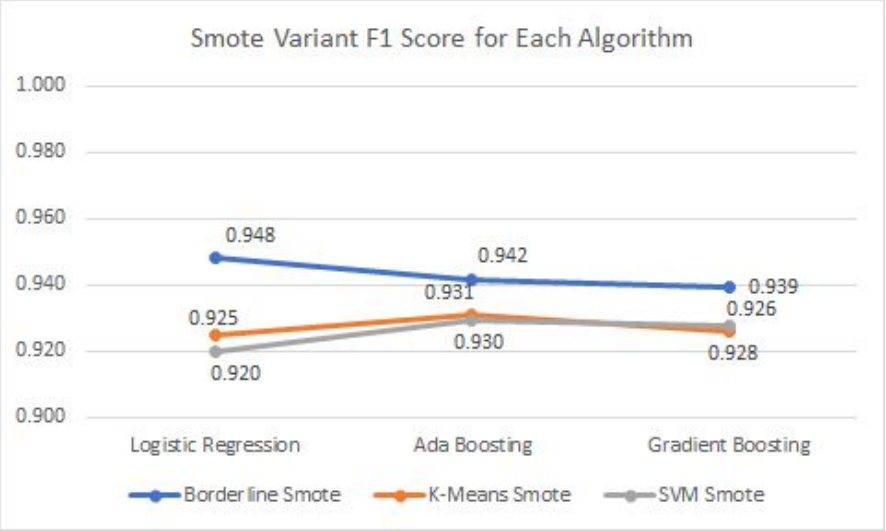
Observation:

It is very evidently visible [Fig 5] that there is a great increase in F1-Score of Logistic Regression while there is hardly any difference in F1-Score of all the other algorithms.

4.5 Research Evaluation

As mentioned, we are researching various techniques of over sampling and decided on using three variants on SMOTE. The three chosen variants of smote are Borderline Smote, K-Means Smote and SVM Smote.

Fig 6



	Borderline Smote	K-Means Smote	SVM Smote
Logistic Regression	0.948	0.925	0.920
Ada Boosting	0.942	0.931	0.930

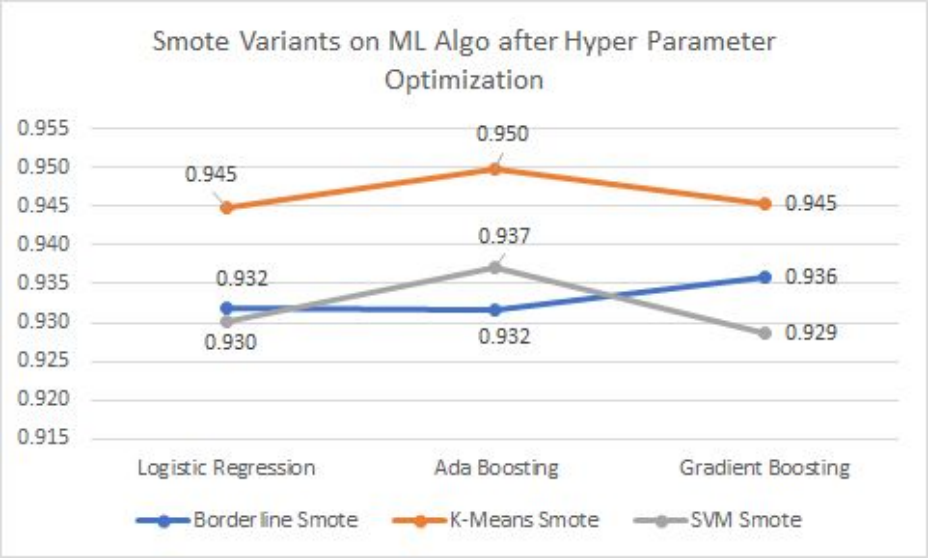
Gradient Boosting	0.939	0.926	0.928

Observation:
We can observe[Fig 6] that Borderline smote is performing the best of all with highest F1-Score as 0.948. While the least performing model is Logistic Regression with least F1-Score as 0.93.

4.5 Hyper Parameter Optimization on Research area

We tried tuning the top 3 algorithm using variants of smote to remove imbalance in the data and we are also tuning the hyper parameters in the similar way as done earlier.

Fig 7



	Borderline Smote	K-Means Smote	SVM Smote
Logistic Regression	0.932	0.945	0.930
Ada Boosting	0.932	0.950	0.937
Gradient Boosting	0.936	0.945	0.929

Observation:
As depicted in the graph (Fig 7), it is observed that K-Means Smote performs the best with a score of around 0.95 for all the top 3 algorithm. While Borderline Smote & SVM Smote are good but not at par with K-Means Smote.

5 Conclusion

We have successfully implement all steps from data pre-processing till research. During the process we have come across various difficulties and resolved them and learned from the mistakes. Our major observation from

the evaluation are: 1) Logistic Regression, Ada Boosting & Gradient Boosting are the best algorithms for breast cancer dataset. 2) Only Logistic Regression has major growth on hyper-parameterization 3) Borderline Smote is the best performing imbalance handling technique 4) On hyperparameter optimization, K-Means smote works the best.

References [1]

1. https://www.kaggle.com/merishnasuwal/breast-cancer-prediction-dataset
2. https://breast-cancer.ca/diag-chnces/
3. https://scikit-learn.org/stable/modules/generated/sklearn.linear_model.LogisticRegression.html
4. https://pypi.org/project/kmeans-smote/#:~:targetText=K%2DMeans%20SMOTE%20is%20an,imbalances%20between%20and%20within%20classes.
5. https://sci2s.ugr.es/keel/keel-dataset/pdfs/2005-Han-LNCS.pdf
6. https://medium.com/vclab/tackling-class-imbalance-with-svm-smote-efa41ec3de5f
7. https://scikit-learn.org/stable/modules/generated/sklearn.model_selection.GridSearchCV.html

[1] Note that the references cited are fictitious for this example.