Study of Practical ML Techniques for Improving Performance of the Off-The-Shelf ML Algorithms.

Jumabek Alikhanov
HumbleBee.AI
Namangan
jumabek@humblee.ai

Azizjon Meliboev
HumbleBee.AI
Kokand
mabdulaziz@humblebee.ai

Abstract

Advancements in machine learning (ML) are facilitating the availability of ML tools for wider use by non-specialists, enabling the deployment of off-the-shelf models with default settings. However, the performance of off-the-shelf models may not be optimal for custom tasks. This study demonstrates simple yet effective techniques for improving ML model performance for custom tasks. The authors utilize exploratory data analysis and data visualization to explain example data to readers, aiding in the formulation and solution of the ML problem. The study then guides users through data preparation and performance assessment metrics for the learning algorithm. Finally, sensitivity analyses are conducted on the classification algorithm type, hyperparameters, amount of data, and presence of feature selection. The authors provide a user-friendly jupyter notebook for hands-on experience, accessible at https://github.com/Jumabek/practical-steps-to-improve-ml-performance.

0 Introduction

In the era of big data, industries are increasingly turning to machine learning (ML) to solve business problems, with the aim of improving efficiency, reducing costs, and optimizing outcomes. However, the popularity and widespread use of ML can lead to its use as a black box, where the decision-making process of the algorithm is not fully understood, and settings that impact performance are not optimized. This can result in sub-optimal solutions where the potential power of ML is not fully exploited. The objective of this paper is to provide a practical introduction to newcomers in the field of ML, to help them optimize their model performance. The authors provide readers with knowledge about the impact of simple ML practices on improving model performance, including detailed exploratory data analysis, experimental settings, and ML procedures. They demonstrate how the performance of classifiers can vary significantly on the same data, and explain the importance of performing a comprehensive analysis. Simple techniques such as hyperparameter tuning and feature selection are shown to improve classifier performance significantly, and the generalization of findings and intuitive expectations are examined with multiple classifiers and settings. The authors illustrate these insights using a widely-used breast cancer classification problem and dataset.

Breast cancer classification is a well-researched area, where numerous studies have investigated the effect of different ML-based classification methods [**?**, **?**, **?**, **?**, **?**, **?**, **?**, **?**]. The authors note that one of the next steps in this research area is to focus more on exploratory data analysis, which provides insight to readers about the type of data. They also highlight the importance of hyperparameter tuning, where most research works use default parameters provided by the tool, but they demonstrate that hyperparameters can have a significant effect even in small datasets. Additionally, the authors explain

and demonstrate the importance of feature selection, where notable improvements in performance are reported.

0.1 Scope

The focus of this work can be delineated into three main areas:

Data Type: Our study concentrates on tabular data that includes numerous features describing the target variable, i.e., the presence of breast cancer.

Dataset: To investigate and demonstrate the vast search spaces during the experiment while ensuring the sufficiency of the dataset to derive meaningful insights, we have chosen a relatively small dataset. Therefore, we selected the publicly available Breast Cancer Coimbra dataset [**?**].

Learning Algorithms: While deep learning-based methods are prevalent in the field of AI, they typically require a sufficiently large dataset to prevent overfitting. As our focus is on a relatively small dataset, we restrict ourselves to using only machine learning-based methods as classifiers.

0.2 Contributions

Our contributions to this research can be summarized as follows:

- An extensive exploratory data analysis that serves as a tutorial for beginners in the field.
- Examination of the impact of feature selection and hyperparameter tuning on multiple off-the-shelf classifiers.
- Demonstration of the effect of the amount of data on various learning models.
- Full source code release in a Jupyter notebook, allowing readers to gain hands-on experience while studying the paper.

0.3 Organization

The remainder of the paper is organized as follows. In Section 2, we review related work on the breast cancer classification task. Section 3 provides an in-depth exploration of the data, including its statistical properties and the relationship between independent variables. In Section 4, we describe our experimental settings and experiments. Section 5 reports the results of exploring the impact of the type of selected machine learning algorithm and their sensitivity to the presence of the feature selection approach. Finally, in Section 6, we conclude the paper. For a list of abbreviations used, please refer to Table **??**.

1 Related Work

The present study aims to explore the effectiveness of feature selection techniques in improving the performance of machine learning-based models for breast cancer diagnosis. To provide context, we will highlight previous studies that are closely related.

Khourdifi et al. [**?**] proposed an overview of the evolution of large data in the health system and applied four machine learning classification algorithms - Random Forest, Naive Bayes, Support Vector Machines (SVM), and K-Nearest Neighbors (KNN) - to a breast cancer dataset, namely the Wisconsin Hospitals Madison Breast Cancer [**?**] (2013). The authors used an effective method to predict breast cancer based on patients' clinical records. According to the performance of models, SVM achieved the highest accuracy score, which is 97.9

Yue and Wang [**?**] provided an overview of machine learning algorithms that include Artificial Neural Networks, SVM, Decision Trees, and K-Nearest Neighbors. Their primary data was drawn from the Wisconsin breast cancer database (WBCD), which is the benchmark database for comparing the results through different algorithms. Machine learning algorithms that have been used on the WBCD database in diagnosis and prognosis showed different levels of accuracy that ranged between 94.36

Alghunaim et al. addressed the problem of breast cancer prediction in the big data context [**?**]. They considered two varieties of data gene expression (GE) and DNA methylation (DM) and chose Apache Spark as a platform. To create models that help in predicting breast cancer, they selected three different classification algorithms - support vector machine, decision tree, and random forest. They conducted a comprehensive comparative study using three scenarios with the GE, DM, and GE and DM combined. These authors used two datasets from The Cancer Genome Atlas [**?**]. The results showed that the SVM classifier in the Spark environment outperforms other classifiers by an accuracy score of 99.68

Asri et al. [**?**] used the SVM algorithm and evaluated their approach on the Wisconsin Breast Cancer dataset. For comparative study, they trained with other algorithms such as KNN, Naives Bayes, and the decision tree variant of CART. The accuracy of prediction for each algorithm is compared, and the results are analyzed. Bayrak et al. [**?**] also used the same dataset and applied SVM and ANN algorithms for the prediction of the classification of breast cancer. SVM showed the best performance in the accuracy of 96.9

Rehman et al. [**?**] proposed the implementation of machine learning models using Logistic Regression, SVM, and KNN on the dataset taken from the UCI repository. Experimental results showed that SVM is the best for predictive analysis with an accuracy score of 92.7

2 Data Description and Exploratory Data Analysis

In this section, we introduce the dataset of choice for this research, its properties, and insights.

2.1 Data Description

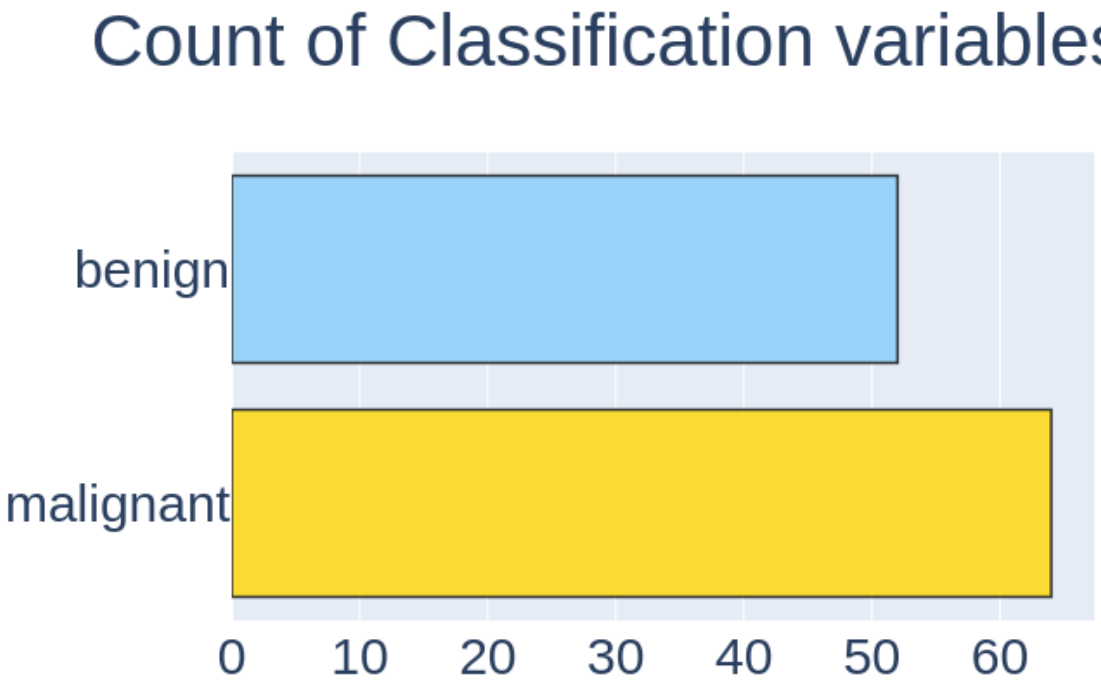
The Breast Cancer Coimbra dataset [**?**] was used for this research. It was created by the Faculty of Medicine of the University of Coimbra and the University Hospital Centre of Coimbra, and gathered 9 independent variables of health information from 64 breast cancer patients and 52 healthy people. The dataset features include Age (years), BMI (g/m2), Glucose (mg/dL), Insulin (µU/mL), HOMA (homeostatic model assessment), Leptin (ng/mL), Adiponectin (µg/mL), Resistant (ng/mL), and MCP-1 (pg/dL). The target variable is encoded as 1 for healthy controls and 2 for patients with breast cancer. The dataset is stored in a CSV file with 10 columns, where the first 9 columns are features and the last column indicates whether breast cancer is malignant. The dataset contains a total of 116 rows, which consist of both patient and healthy control groups. The count and distribution of the target variable can be seen in Figure 1 a and b. A detailed description of the features can be found in Table 1.

Table 1: Description of Features.

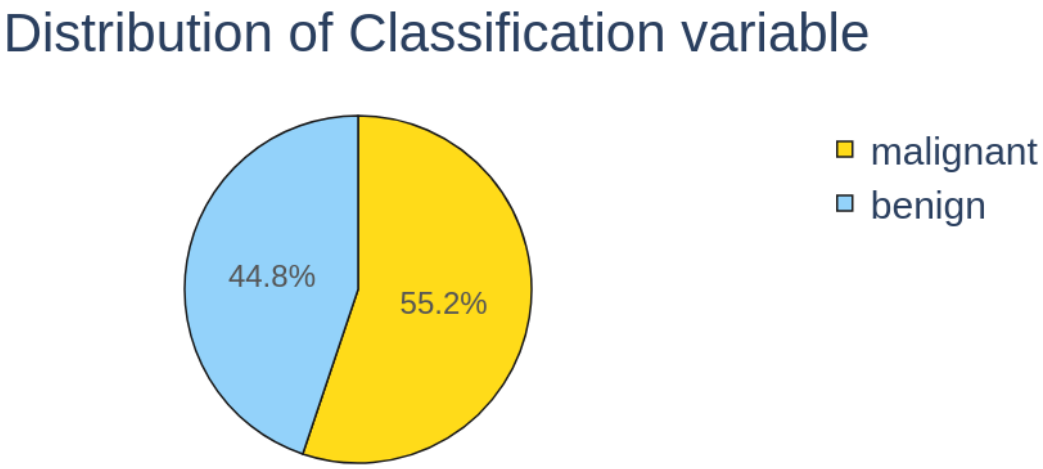
	Age	BMI	Glucose	Insulin	HOMA	Leptin	Adiponectin	Resistin	MCP.1
mean	57.3	27.6	97.8	10.0	2.7	26.6	10.2	14.7	534.6
std	16.1	5.0	22.5	10.1	3.6	19.2	6.8	12.4	345.9
min	24.0	18.4	60.0	2.4	0.5	4.3	1.7	3.2	45.8
25% percentile	45.0	23.0	85.8	4.4	0.9	12.3	5.5	6.9	270.0
50% percentile	56.0	27.7	92.0	5.9	1.4	20.3	8.4	10.8	471.3
75% percentile	71.0	31.2	102.0	11.2	2.9	37.4	11.8	17.8	700.1
max	89.0	38.6	201.0	58.5	25.1	90.3	38.0	82.1	1698.4

2.2 Correlation analysis

Correlation describes the data moving together. For instance, we can see in Fig 2 Insulin and HOMA are correlated with a correlation coefficient of 0.93 which means almost every time the insulin level changes so does the HOMA. Correlation matrices can be analyzed for interdependence analysis and relationships among variables [**?**]. A general correlation matrix has parameters, each one representing the level of correlation between pairs of variables. The correlation matrices contain positive, negative, and neutral correlation coefficients as shown in Fig 2. Highly correlated features are Insulin and HOMA with a 0.93 coefficient. Independent variables that correlate negatively the most are Adiponectin and BMI with a correlation coefficient of -0.30. The most uncorrelated features are Age and BMI whose coefficient is 0.008. It is noteworthy to mention that correlation does not indicate causation [**?**].



(a)



(b)

Figure 1: Count and distribution of target variable

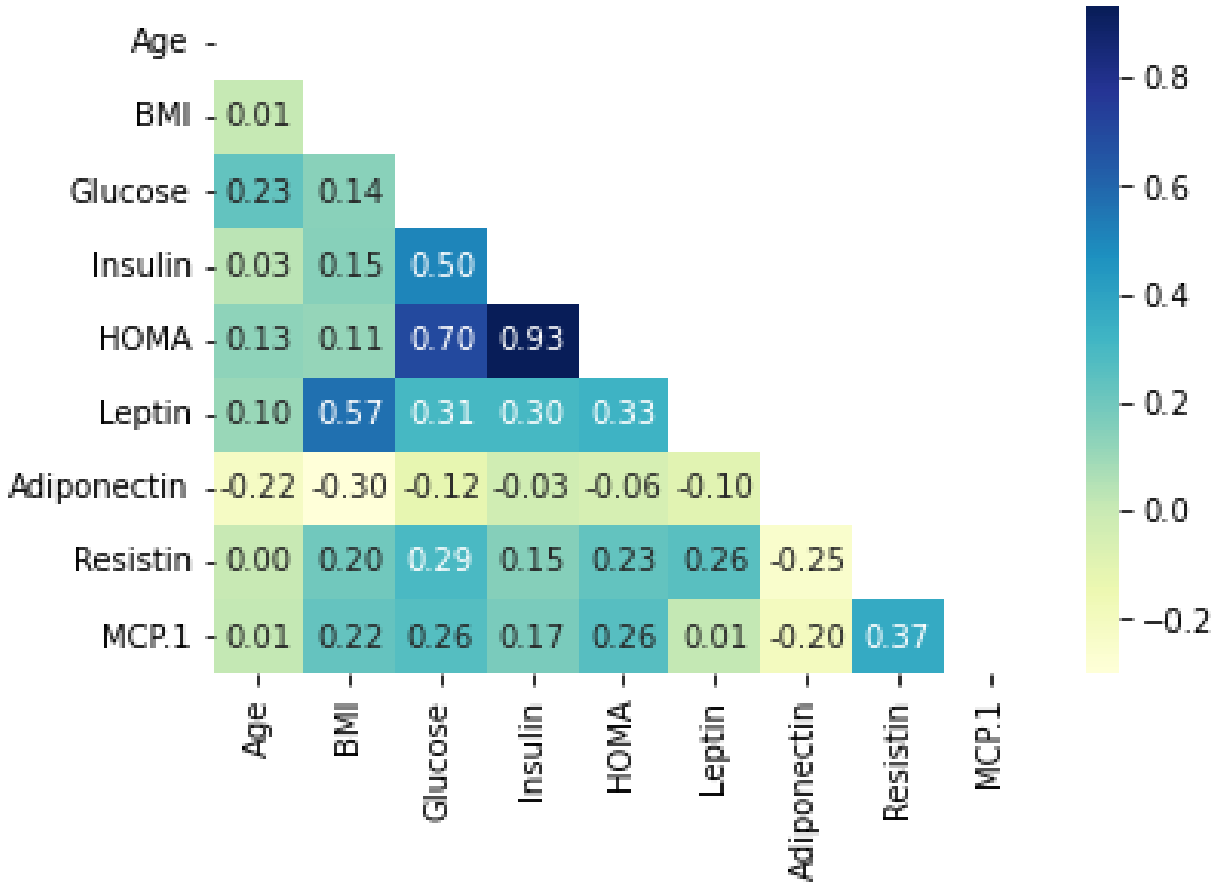


Figure 2: Correlation matrix.

To gain insights into how correlated features define the outcome variable, we plotted scatterplots for the two most highly correlated feature pairs: HOMA vs. Insulin and Glucose vs. HOMA. These are shown in Fig 3a and 3b.

The correlation plot of HOMA vs. Insulin indicates that when insulin and HOMA are proportional, it is less likely for breast cancer to be malignant. On the other hand, the plot for HOMA and Glucose shows that high levels of both HOMA and Glucose are associated with malignant breast cancer.

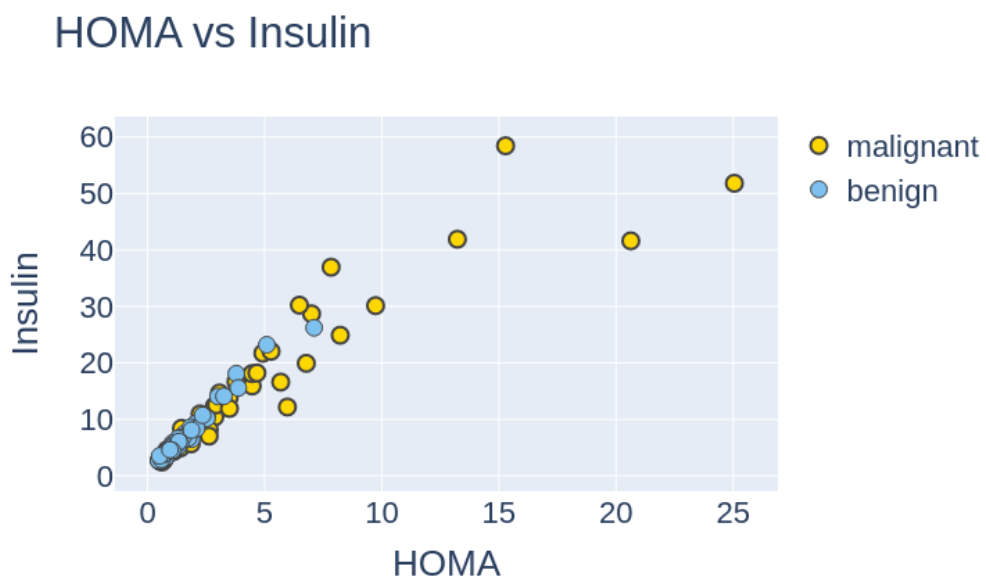
Interestingly, the worst uncorrelated features, Age and BMI, are represented in Fig 3c. Despite being independent of each other, they do not seem to provide much information for distinguishing between malignant and benign cancer.

Overall, these insights suggest that HOMA and Glucose may be important factors in determining the malignancy of breast cancer, while Age and BMI may not be as useful in this regard.

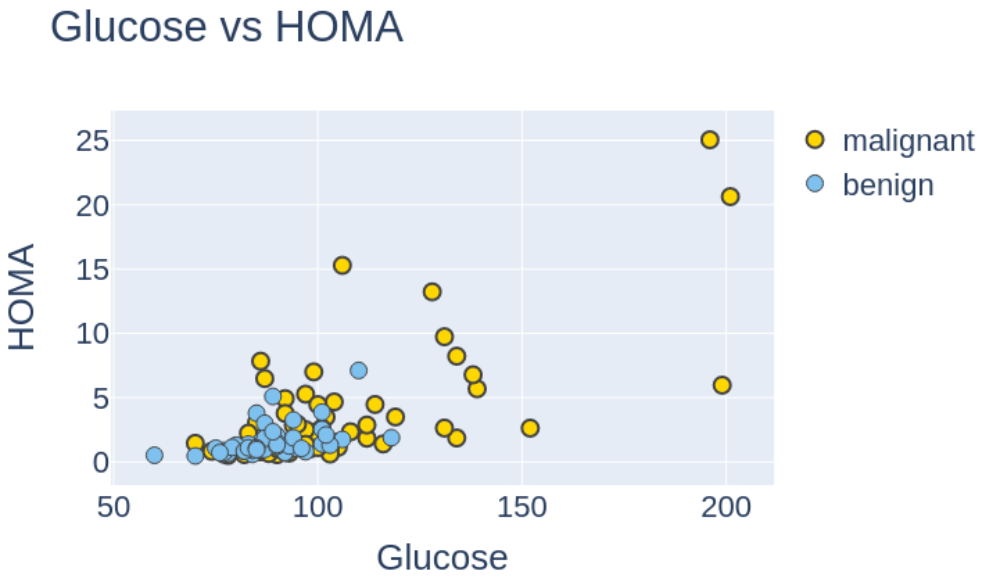
2.3 Distribution of individual features over positive and negative samples

In the previous subsection, we saw that independent variables such as HOMA, Insulin, and Glucose are helpful for identifying whether breast cancer is malignant or benign. In the following, we will inspect their individual distribution.

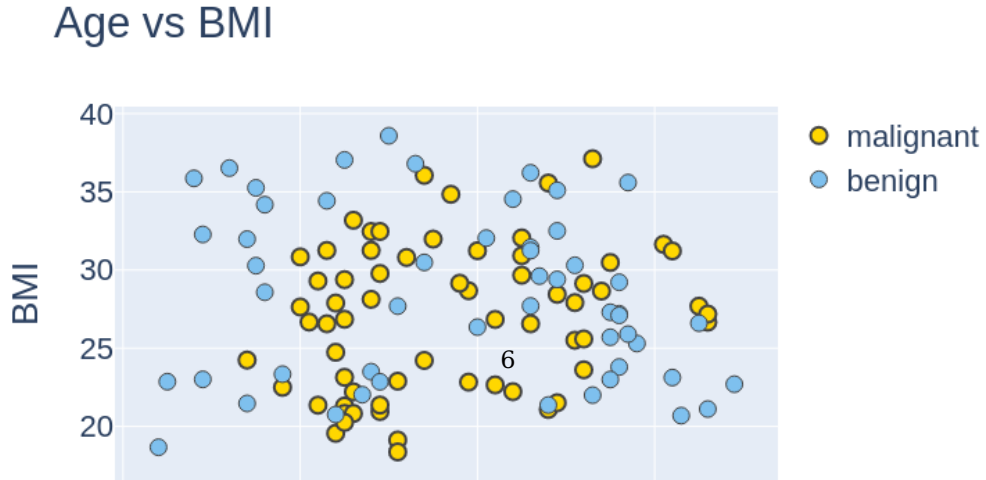
Overall we can observe from Fig. 4 a,b, and c that higher levels of HOMA, Insulin and Glucose are associated with a higher probability of malignant breast cancer.

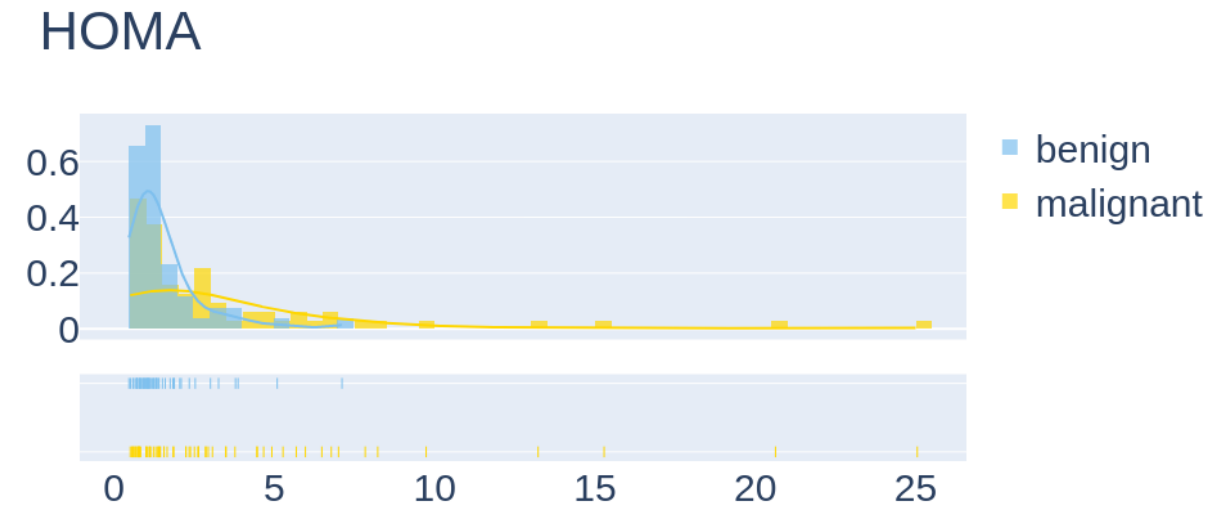

HOMA vs Insulin
Insulin
HOMA
malignant
benign
0
10
20
30
40
50
60
0
5
10
15
20
25

(a)

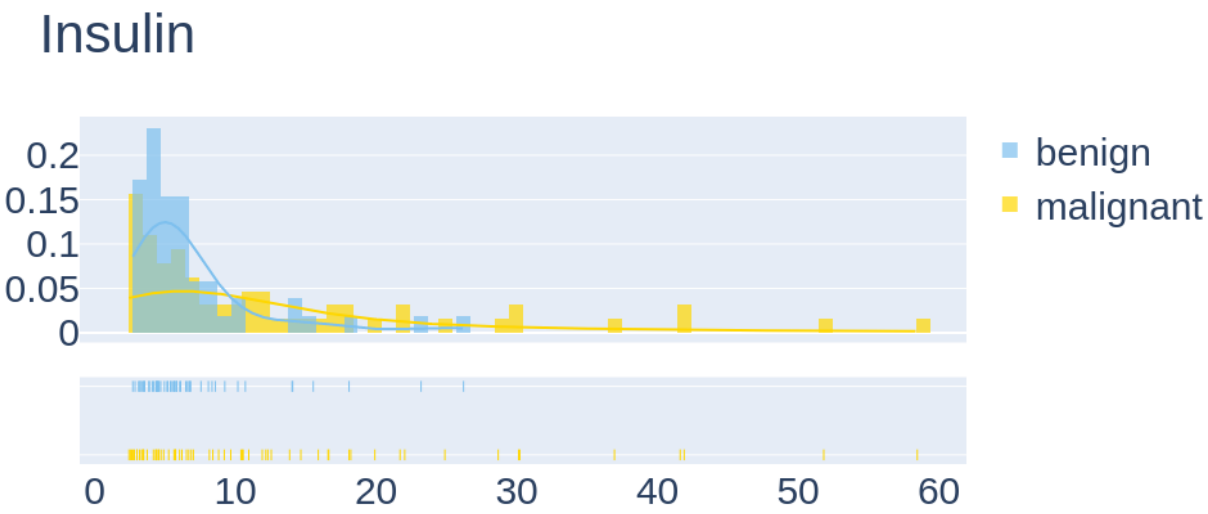

Glucose vs HOMA
HOMA
Glucose
malignant
benign
0
5
10
15
20
25
50
100
150
200

(b)

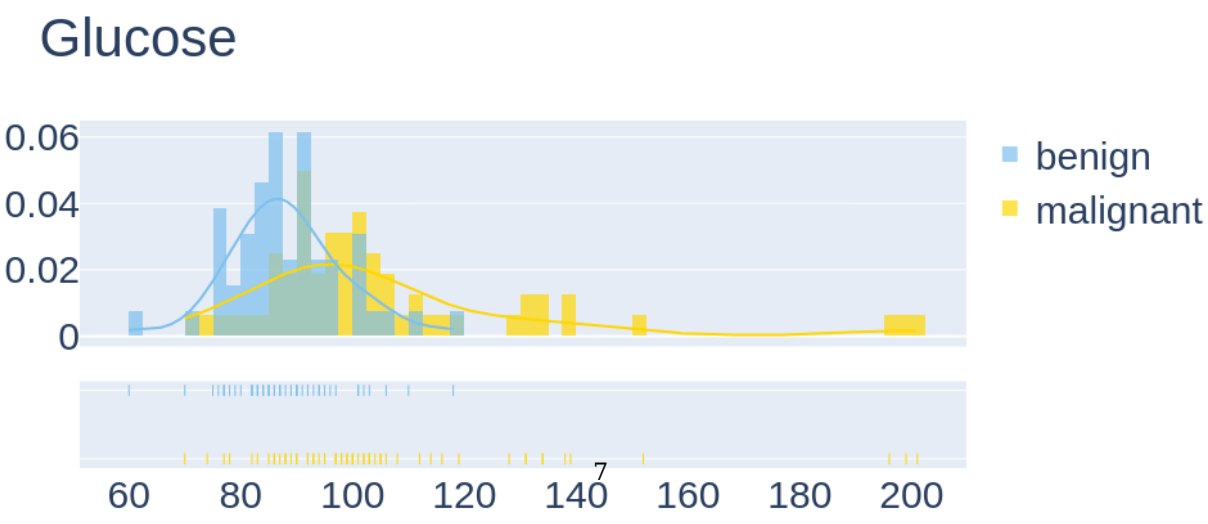

Age vs BMI
BMI
malignant
benign
20
25
30
35
40


HOMA
0.6
0.4
0.2
0
0
5
10
15
20
25
benign
malignant

(a)


Insulin
0.2
0.15
0.1
0.05
0
0
10
20
30
40
50
60
benign
malignant

(b)


Glucose
0.06
0.04
0.02
0
60
80
100
120
140
160
180
200
benign
malignant

(c)

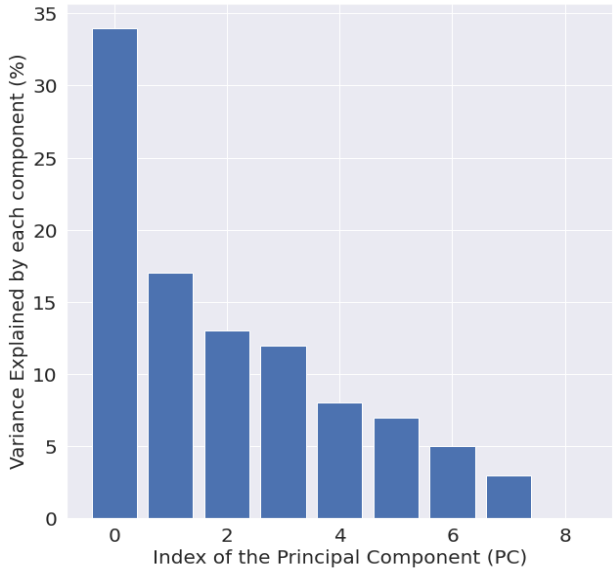


Figure 5: Percentage of Variance in data that is explained by each Principal Component(PC)s.

Table 2: Top 4 features that contributed for each principal component

Components	**Top 4 Projected Features**
Component0	HOMA, Insulin, Glucose, Leptin
Component1	BMI, Adiponectin, Insulin, HOMA
Component2	Leptin, MCP.1, BMI, Resistin
Component3	Age, MCP.1, Resistin, Adiponectin
Component4	Resistin, Adiponectin, Insulin, Leptin
Component5	MCP.1, Adiponectin, Resistin, Age
Component6	Glucose, Insulin, Age, Resistin
Component7	Leptin, BMI, MCP.1, Resistin
Component8	HOMA, Insulin, Glucose, BMI

2.4 PCA representations

Principal Component Analysis (PCA) is a common technique for the reduction of high data dimension. PCA represents a multivariate data table as a smaller set of variables (summary indices) in order to observe insights such as trends, clusters, and outliers. In other words, PCA is helpful because it can project high-dimensional data into a lower dimension. For applying PCA, data normalization plays an essential role because for that reason we first normalized the data. We implemented that the number of PCA components are same as the number of original features which is nine. As shown in Fig. 5 a variation captured by each Principal Component decreases. Interestingly only seven PCAs are enough to capture 100% variation of nine-dimensional features.

In Fig. 6, we can see that with only two principal components we can already get a representation that is rich enough to distinguish malignant and benign samples in 3 dimensional plot.

In Table 2, We can see the top 4 features that contribute for the given principal components. Common features that affect the variance in the data are HOMA, Insulin, Glucose, and Leptin.

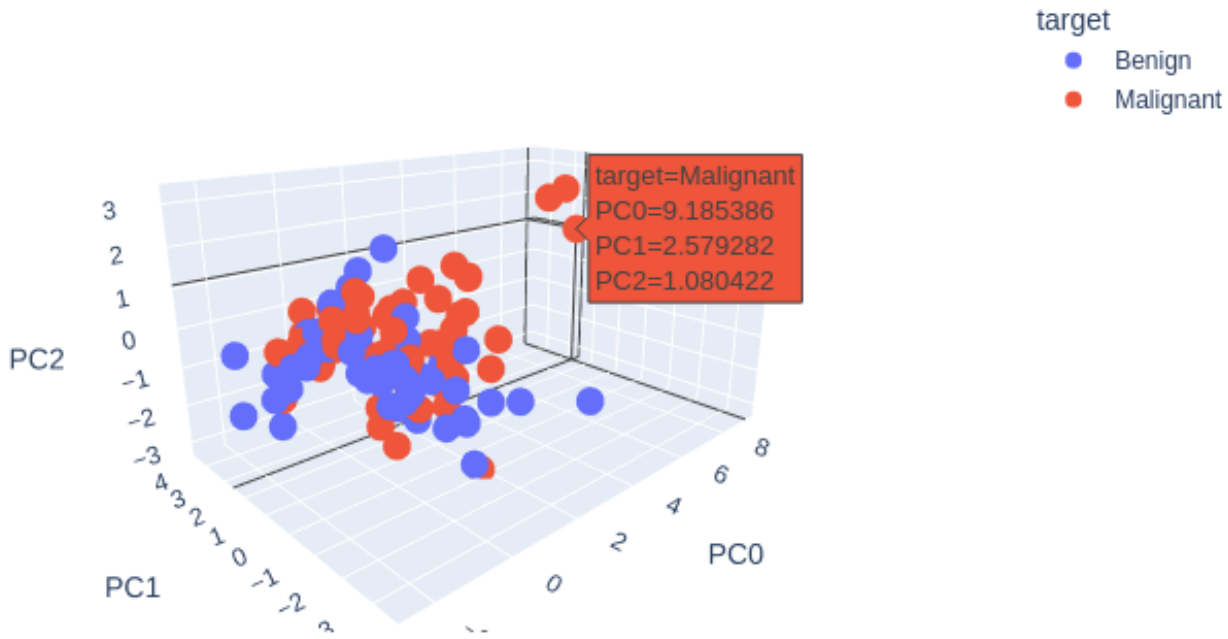


Figure 6: Data description with three most prominent principal component (PC)s in 3 dimensions.

3 Experimental Setting

In this section, we first describe our experimental dataset, machine learning methods and their hyper-parameter settings, type of feature selection technique used. Then we move on defining performance metrics.

3.1 Dataset Preparation

In this section, we provide more detailed information on the data preprocessing steps that we performed on the Coimbra Breast Cancer dataset. Our goal was to prepare the dataset for machine learning analysis by addressing potential issues related to data quality, normalization, and splitting the data into training and test sets.

Firstly, we checked the quality of the data by looking for any missing or erroneous values. Fortunately, we found that the dataset was complete and had no missing or erroneous values.

Next, we standardized the data using a standard normalization technique known as zero-centering. This technique is used to ensure that each feature has a similar scale and therefore, has an equal weight in the analysis. We used the following formula for zero-centering the data:

$$z = \frac{x - \mu}{s} \tag{1}$$

Here, x represents the original value of the feature, μ represents the mean of the feature in the training samples, and s represents the standard deviation of the feature in the training samples.

After standardization, we split the data into training and test sets. We randomly sampled 20

To further optimize our model, we performed 5-fold cross-validation during hyperparameter tuning. During cross-validation, the training set was divided into 5 subsets of equal size, and each subset was

Table 3: Train and test distribution.

	Train	**Test**
Benign	46	6
Malignant	46	18

used once as the validation set while the remaining subsets were used for training the model. This process was repeated 5 times with a different subset used as the validation set each time.

The final step in our data preprocessing was to encode the target variable (i.e., breast cancer diagnosis) as a binary variable. We mapped the Malignant diagnosis to 1 and the Benign diagnosis to 0, which allowed us to use binary classification techniques to train our model.

In summary, our data preprocessing steps involved checking the quality of the dataset, standardizing the data, splitting the dataset into training and test sets, and encoding the target variable. These steps were necessary to ensure that our data was ready for machine learning analysis and to optimize the performance of our machine learning model.

3.2 Machine Learning Classifiers and hyper parameter settings

Machine learning classifiers are algorithms that automatically categorize or sort data into one or more "classes" and this task is a type of supervised learning problem.

In general, machine learning classifiers have two types of parameters: those that are learned from the training data such as the weights in logistic regression, and the parameters of a learning algorithm that are optimized separately. The latter are the tuning hyper-parameters of a model. For example, the strength of a regularization parameter in logistic regression or the parameter that defines the maximum depth of a decision tree. We applied one of popular hyper-parameter optimization technique called Grid search, which can further help to improve the performance of a model by finding the optimal combination of hyper-parameter values. The Grid search approach is quite simple that can search for optimal parameters from a from a list of potential values for different hyperparameters and the grid search algorithm evaluates the model performance for each combination to obtain the classifier with optimal combination of hyper-parameter values.

3.3 Cross Validation

Cross validation is when part of the data is reserved (holdout) for evaluating the model. For small datasets, it is common to use K-Fold Cross Validation where data is split into train/test set K times and results are aggregated. Evaluating the model on each part of data allows to take advantage of all the data for evaluation. Additionally, repeating the split K-times ensures to mitigate the bias of particular split.

Another approach, more commonly used on larger datasets is simple train/test set split where model development (training, validating) is done on train set and evaluation is performed on test set. Due to its simplicity we resort to using this approach in our experiments.

3.4 Hyper-Parameter Tuning

Hyper parameter tuning is considered as part of the model development procedure. Hence, during tuning only train set is used. In this paper, grid search is exploited to find the optimal parameters from the set of potential candidates. Grid search is chosen owing to its simplicity which is sufficient for the objective of this paper. For more advanced approaches we refer the readers to recent works on hyper parameter optimization such as in [**?**].

3.5 Feature Selection

For the supervised problem setting there are different types of feature selection methods. As shown in Fig. 7, some feature selection algorithms use data characteristic (filter methods), others may iteratively

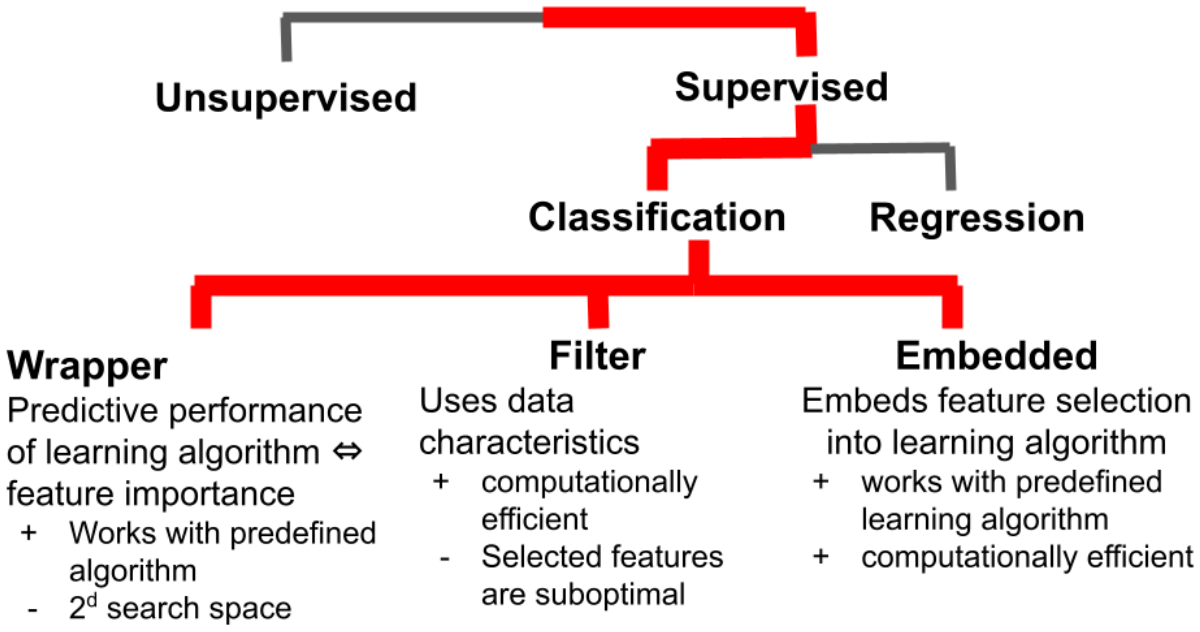


Figure 7: Overview of feature selection approaches for the supervised classification setting.

take advantage of the learning algorithm's predictive performance to assess the quality of selected features so far (wrapper methods), or third type of feature selection methods can make use of the intrinsic structure of a learning algorithm to embed feature selection into the underlying model (embedded methods). Wrapper methods work with predefined algorithm so feature selection can provide optimial results for the specific classification algorithm. On the other hand it has the disadvantage of large search space. This is because algorithm needs to evaluate each subset of features to find the optimal subset. Filter methods are independent of any learning algorithms where feature importance is ranked according to some evaluation criteria to select the top most important features. Hence, they are efficient however, selected set of features are not optimal for the specific classifier. Embedded methods are the trade-off in between where they include interaction with learning algorithm (i.e., classifier) and they are far more efficient than wrapper based methods since no combinatorical evaluation is required for each subset of features. Ridge regression, lasso and elasticnet are the most widely used feature selection methods [**?**].

3.5.1 Recursive Feature Elimination (RFE)

For the purpose of this study we selected Recursive feature elimination (RFE) - an instance of wrapper based feature selection methods. Our justification is that it is a simple method that can be used to demonstrate the power of feature selection to the ML practitioners due to its powerful effect on classification performance.

Applicability of RFE RFE requires classifier to have importance coefficient. Hence, non parametric classifiers such as KNN and GP cannot be used with RFE.

3.6 Performance Metrics

Essential metrics for measuring the performance of a binary classification model can be listed as ROC, Learning curve, Precision-Recall curve and Confusion Matrix.

The Confusion Matrix (CM) is fundamental metric for evaluating the performance of a classification model. It also allows visualization of the algorithm's performance. The True Positive (TP) measure

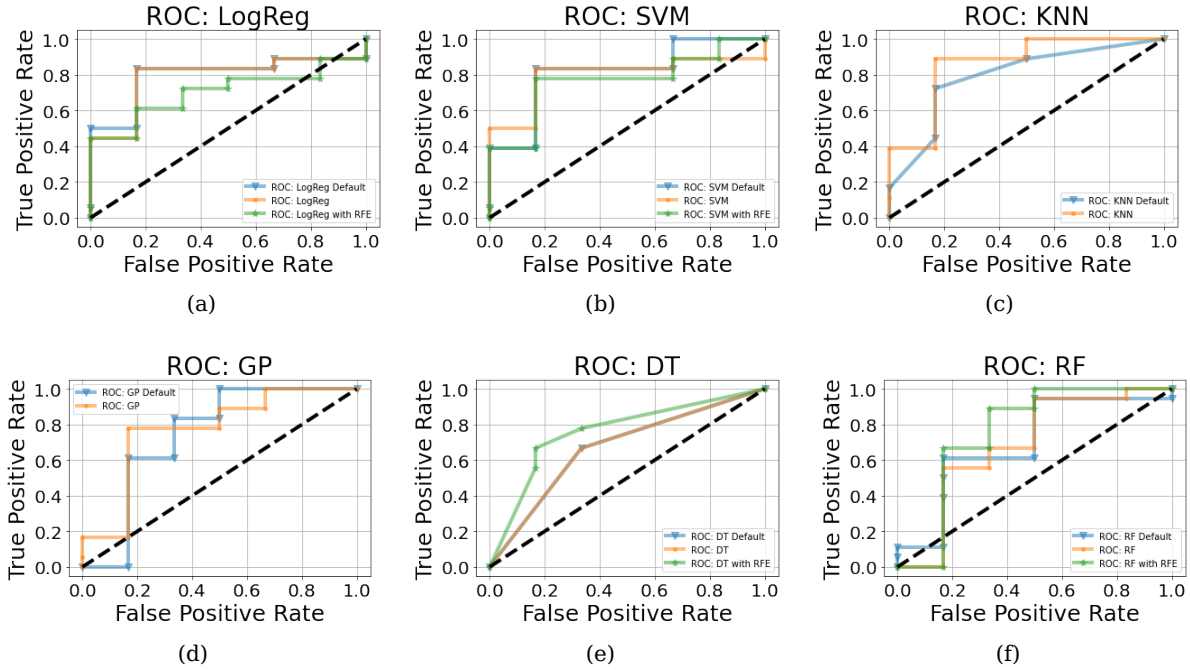


Figure 8: Comparison of Classifiers with ROC Curve.

indicates the number of samples that are positive and classifier predicted the them as positive. Note in the problem of breast cancer detection, positive class refers to Malignant cancer. The False Positive (FP) is the equivalent samples that are not positive but classifier predicted as positive (i.e., malignant). Usually most important part of confusion matrix are the TP and FP. The True Negative (TN) is equivalent to those correctly classified as negative (i.e., benign). The False Negative (FN) is equivalent to those incorrectly identified as negative (i.e., benign) even though they are positive (i.e., malignant). For assessing the performance of the model following evaluation metrics are used

Accuracy is the percentage of the number of malignant classified correctly versus total the of patients shown in Equation 2.

$$\frac{TP + TN}{TP + TN + FP + FN} \tag{2}$$

Recall shows the percentage of the number of patients identified correctly over the total number of malignant, as shown in Equation 3.

$$\frac{TP}{TP + FN} \tag{3}$$

Precision is equivalent to the number of correctly identified patient is divided by the summation of correctly identified and incorrectly identified, as shown in Equation 4.

$$\frac{TP}{TP + FP} \tag{4}$$

Precision Recall Curve Precision-recall curve shows the tradeoff between precision and recall for different thresholds.

ROC Curve The Receiver operating characteristic (ROC) curve is created by plotting the True positive rate (TPR) against the False positive rate (FPR) at various threshold settings.

Learning Curve The learning curve is useful to plot for improving the performance of our models included in our analysis. In our work, learning curve shows curves of training score and cross-validated

Table 4: Sensitivity analysis: hyper parameter tuning and feature selection.

classifier	scenario	accuracy	precision	recall	f1score
LogReg	default	0.708	0.923	0.667	0.774
	tuned	0.708	0.923	0.667	0.774
	selection	0.708	0.867	0.722	0.788
SVM	default	0.792	0.933	0.778	0.848
	tuned	0.750	0.929	0.722	0.813
	selected	0.750	0.929	0.722	0.813
KNN	default	0.750	0.929	0.722	0.813
	tuned	0.792	0.933	0.778	0.848
GP	default	0.750	0.875	0.778	0.824
	tuned	0.708	0.824	0.778	0.800
DT	default	0.667	0.857	0.667	0.750
	tuned	0.667	0.857	0.667	0.750
	selected	0.708	0.923	0.667	0.774
RF	default	0.625	0.800	0.667	0.727
	tuned	0.667	0.857	0.667	0.750
	selected	0.833	0.889	0.889	0.889

score. The cross-validation is for evaluating predictive models by partitioning the into a training set and a test set.

4 Experimental Results

4.1 Sensitivity Analysis

In this section sensitivity analysis for multiple classification algorithms in different settings. High level overview of our experiments can be seen in Table 4. In the table classifier scenario column describes three separate scenerios for each classifier. ‘default‘ corresponds to model with default hyperparamters, ‘tuned‘ corresponds to model with tuned hyper parameters. Hyper parameters tuned with grid search. Finally ‘selected‘ scenerio corresponds to model with tuned hyper parameters and RFE feature selection.

Table 4 shows very common problem in ML where techniques such as hyper parameter tuning and or feature selection not always improve performance. For instance, hyper parameter tuning is actually degreding performance for SVM, GP and RF. We speculate this is effected by the small data size where training set tuned hyper parameters are not the optimal ones for test set. Another interesting observation is RFE improved performance for LogReg, DT and RF while it insignificant effect on SVM. Useful insight from the results is that SVM with default parameters are well suited for our task. This could be perhaps explained by decision boundary of SVM where only specific examples are needed. In other words since SVM needs data points only in the decison boundary, it is well suited for small scale datasets.

Another, aspect of comparision is ROC Curve which provides evaluation of classifier on multiple thresholds. As can be seen from Fig. 8, RFE based feature selection is improving perfromance only for DT and RF. RFE negatively affects LogReg and SVM.

Learning curve of a classifier provides useful insight into how classifier learning improves as it sees more data. As can be seen from Fig. 9 and Fig. 10 such curves are affected whether hyper parameter of the model is tuned and/or whether feature selection is employed.

4.2 Performance of each classifiers

In this section we describe the confusion matrix as another aspect of classifier performance. For each classifier, their best setting from Table 4 is chosen. F1-score is used as our primary metric. Description of confusion matrix is explained in Experimental Setting section. Default threshold of 0.5 for binary classification setting is used for computing CM. It is noteworthy to mention that CM is depicted only

for test set which is not seen during model development. Main takeaway message from this section is that, feature selection and hyper parameter tuning does not always improve performance. For some classifiers it improved but for some default setting resulted in best empirical performance.

Logistic Regression: Best performance is achieved where RFE feature selection is applied and hyperparameters are tuned. Corresponding confusion matrix can be seen in Fig. 11a where 13 out of 18 malignant cases are detected correctly. Following hyper-parameters are selected.

```
'C': 100,
'penalty': 'l2'
```

SVM: Interestingly, for SVM classifier, best performing case turned out to be without feature selection setting on off-the-shelf (default) hyperparameters. Corresponding confusion matrix can be seen in Fig. 11b.

KNN: Non-parameteric KNN classifier achieved same performance as SVM where best performing case was found after tuning its hyper parameters. Where best parameters found by grid search are shown below. Corresponding confusion matrix can be seen in Fig. 11c.

```
'algorithm': 'auto',
'leaf_size': 10,
'n_neighbors': 6,
'weights': 'distance'
```

Gaussian Process: For GP, off-the-shelf hyper-parameters resulted in best performance. Corresponding confusion matrix can be seen in 11d.

Decision Tree: For DT, best performing case was found with hyper parameter tuning and RFE feature selection. Corresponding confusion matrix can be seen in Fig. 11e. Selected hyper-parameters are shown below.

```
'criterion': 'gini',
'max_depth': 6,
'min_samples_leaf': 1,
'min_samples_split': 6
```

Random Forest: For RF, best performing case was found with hyper parameter tuning and feature selection. Fig. 11f shows highest TPR for RF classifier where 16 out of 18 malignant cases are identified correctly. Selected hyper-parameters can be found below.

```
'criterion': 'gini',
'max_depth': 6,
'max_features': 'auto',
'n_estimators': 500
```

5 Conclusion

In this paper we demonstrated practical steps for improving ML model's classification performance. Results showed how simple feature selection approach can improve the performance of ML classifier significantly. Importance of hyper-parameter tuning is also shown. Generalization of findings for multiple off-the-shelf algorithms is also provided. Conclusion can be made that hyper-parameter tuning and feature selection can be applied with simple steps to obtain notable performance boost. Accompanying hands on source code which will be made public once paper is published is written in user friendly fashion. Jupyter notebook helps practitioners of the field achieve the same in their custom ML task.

-0cm

References

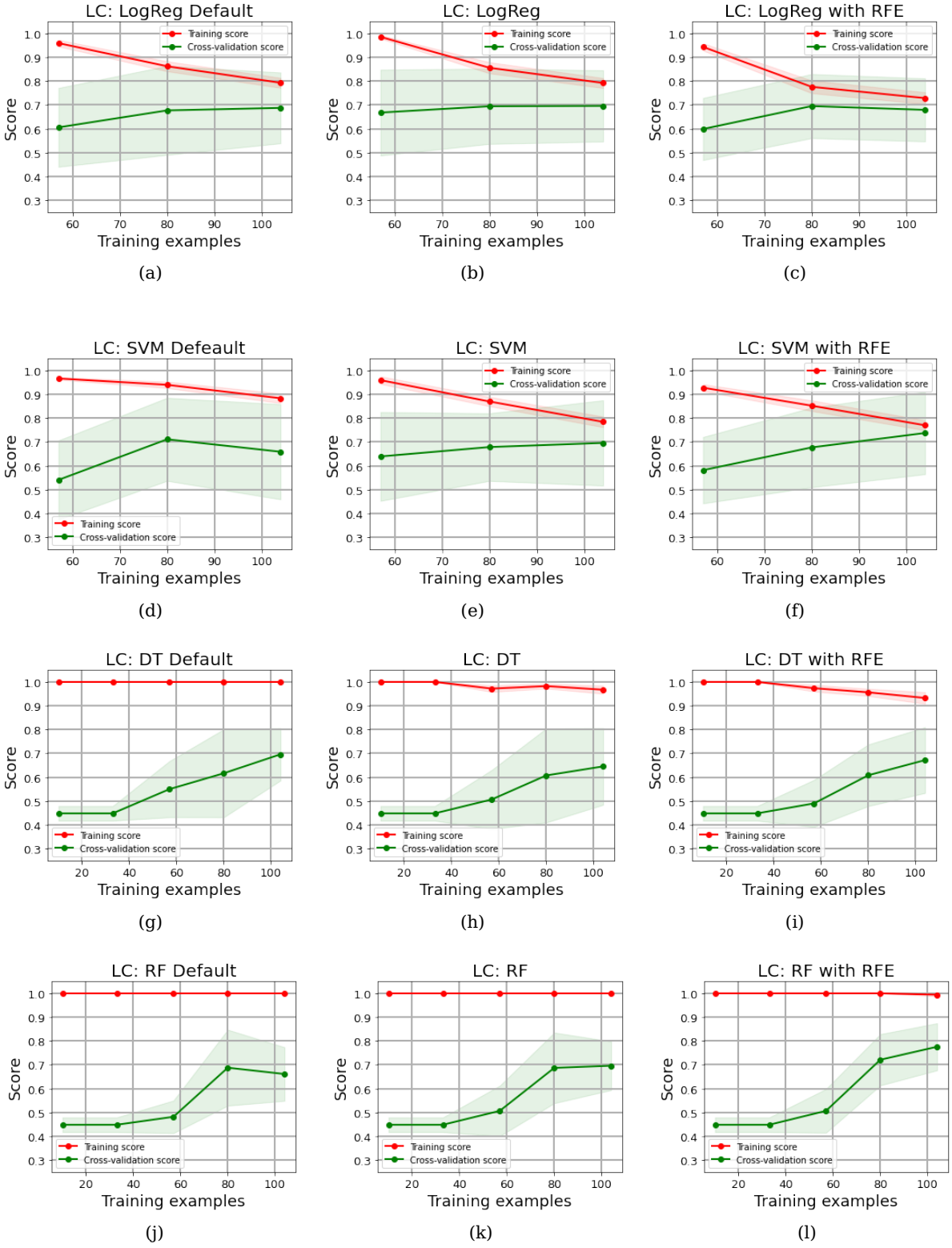


Figure 9: Sensitivity of learning curve on hyper parameters and feature selection. Each row corresponds to separate classifier: LogReg, SVM, DT, RF.

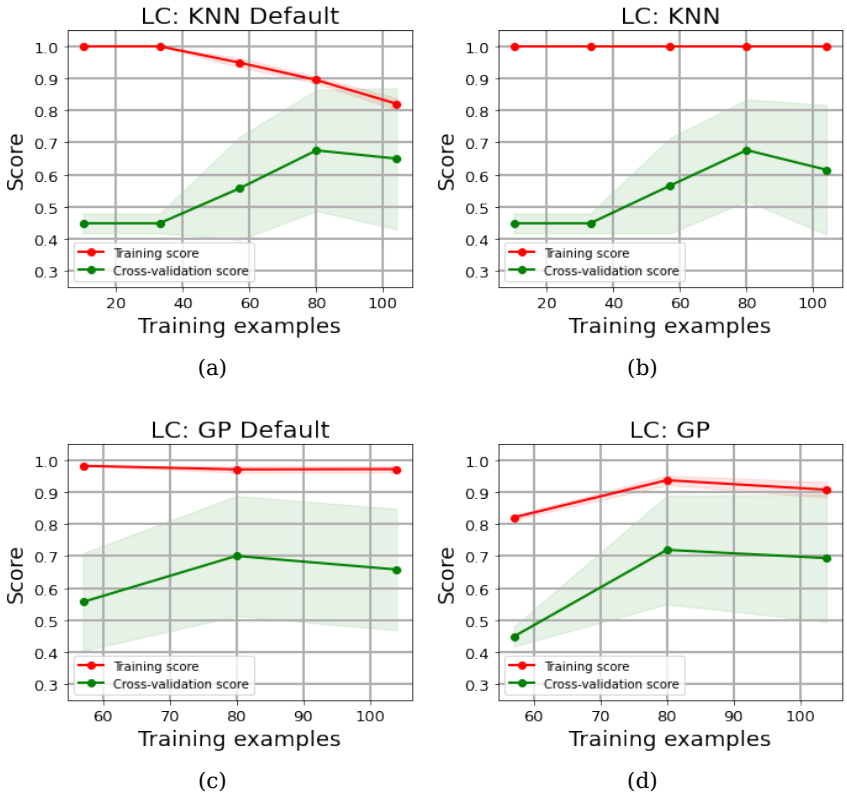


Figure 10: Sensitivity of learning curve on hyper parameters and feature selection.

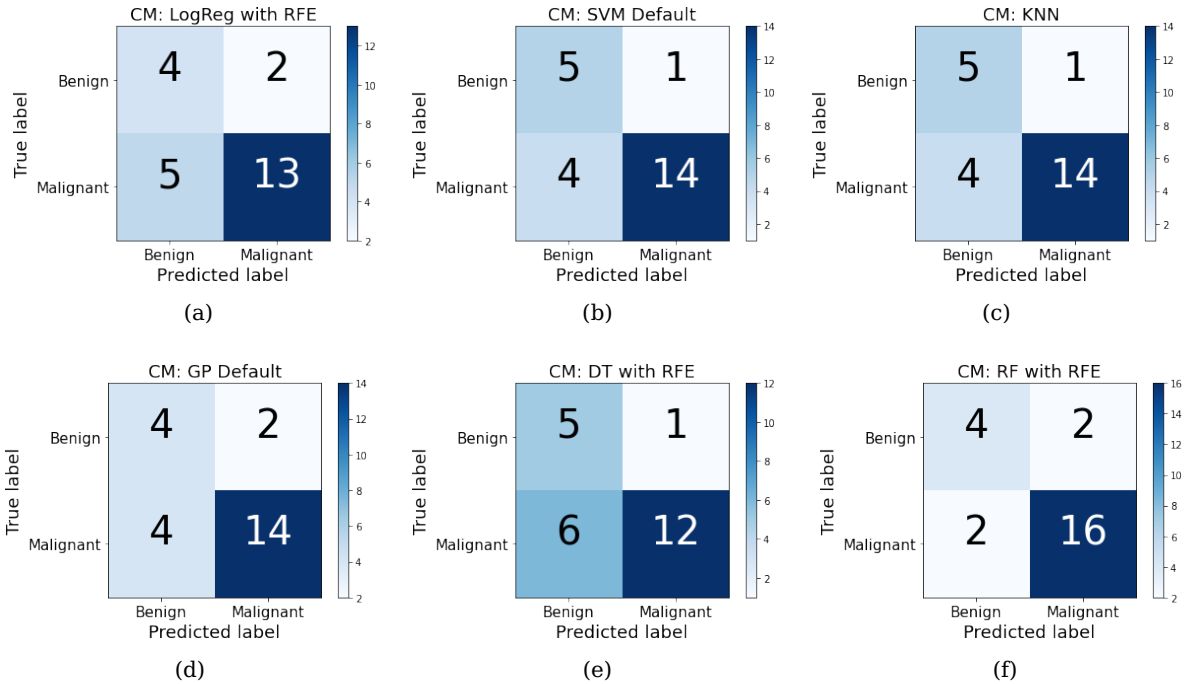


Figure 11: Confusion matrix: LogReg, SVM, KNN, GP, DT, RF.