Machine Learning on Breast Cancer Wisconsin (Diagnostic)

Predict whether the cancer is benign or malignant

Yuqin Shen

Electrical and Computer Engineering

Duke University

Table of Contents

1. Data Set

Breast Cancer Wisconsin (Diagnostic) -- Predict whether the cancer is benign or malignant [1] , [2]

2. Rationale and objectives of the study

2.1 Goal

The goal of this project is to predict whether the breast cancer is benign or malignant.

2.2 Public health concern for women worldwide

Breast cancer is the most common cancer among women and one of the major causes of death among women worldwide. One in eight women in the United States will be diagnosed with breast cancer in her lifetime. Worldwide, it is estimated that with more than one million cases and nearly 600,000 deaths occurring annually [3] . On average, every 2 minutes a woman is diagnosed with breast cancer and 1 woman will die of breast cancer every 13 minutes [4] [3] . Public health data indicated that the global burden of breast cancer in women, measured by incidence, mortality, and economic costs, is substantial and on the increase.

2.3 Early detection helps recovery

When detected in its early stages, there is a 30% chance that the cancer can be treated effectively [5] . Death rates from breast cancer have been declining since about 1990, in part due to better screening and early detection, increased awareness, and continually improving treatment options. A woman can get a better chance of complete regaining from the cancer if diagnosed at prior stage, thus firing the need of development of efficient diagnosis techniques like Information and Communication Technologies (ICT) [6] .

2.4 Detection dilemma

However, late detection of advanced-stage tumors makes the treatment more difficult. Currently, the most used techniques to detect breast cancer in early stages are: mammography (63% to 97% correctness), FNA (Fine Needle Aspiration) with visual interpretation (65% to 98% correctness) and surgical biopsy (approximately 100% correctness) [7] . Therefore, mammography and FNA with visual interpretation correctness varies widely, and the surgical biopsy, although reliable, is invasive and costly [8] .

What is more, normally in traditional cases, double-reading (same mammograms are read by two radiologists individually) has been encouraged in decreasing the percentage of overlooked cancers and it is at present the supreme technique incorporated in most of the screening programs instead of the fact that it earns in surplus workload and costs.

Therefore, a platform for the computer-aided detection/diagnosis systems is established for backing up a single radiologist reading mammograms providing sustenance to her/his decisions.

3. Previous research work

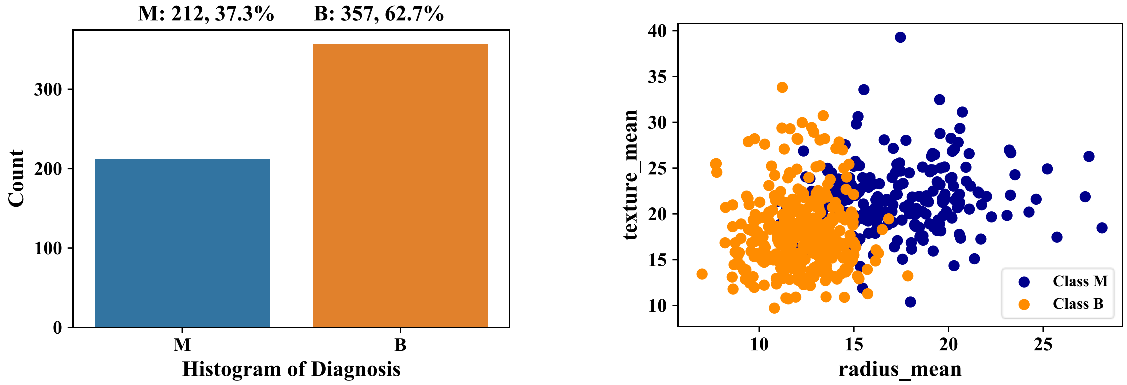
Some papers were published during the last 30 years trying to achieve the best performance for the computational interpretation of FNA samples.

- Omar,et al [9] applied three machine-learning algorithms (Support Vector Machine, K-nearest neighbors, and Decision tree) to do classifier training. The outcomes of their study have revealed that quadradic support vector machine grants the largest accuracy of (98.1%) with lowest false discovery rates. Their experiments were carried out using Matlab.
- Vikas, et al [10] applied three popular data mining algorithms (Naïve Bayes, RBF Network, J48) to develop the prediction models. They used 10-fold cross-validation methods to measure the unbiased estimate of the three prediction models. The results indicated that the Naïve Bayes performed best which received 97.36% accuracy as a predictor on the holdout sample, RBF Network came out to be the second with 96.77% accuracy, J48 came out third with 93.41% accuracy.
- Ioannis, et al [11] used probabilistic and generalized regression neural classifiers to deal with the breast cancer diagnosis and prognosis using the Wisconsin Diagnostic Breast Cancer (WDBC) as well as the Wisconsin Prognostic Breast Cancer (WPBC) data sets. The accuracy of the neural classifiers reaches nearly 98% for the diagnosis and 92% for the prognosis problem.
- Borges, et al. [12] used two well-known machine learning techniques to test their training models: Bayesian Networks and J48. The best accuracy in their paper was achieved by the Bayesian Networks algorithm, 97.80% of accuracy in its best configuration, while J48 had 96.5% of accuracy.

More previous research work could be searched in Google Scholar or Springer, archived paper could be downloaded via Duke Library.

4. Data Description

Wisconsin Diagnostic Breast Cancer (WDBC) dataset were utilized in this project. The dataset is publically available [1] , [2] , and was created by Dr. William H. Wolberg, physician at the University of Wisconsin Hospital at Madison, Wisconsin, USA. The dataset consists of **569** instances (357 benign – 212 malignant), where each one represents FNA test measurements for one diagnosis case. For this dataset each instance has **32** attributes, where the first two attributes correspond to a unique identification number and the diagnosis (benign / malignant). The rest 30 features are computations for ten real-valued features, along with their mean, standard error and the mean of the three largest values for each cell nucleus respectively. The attribute information of the features is summarized in Table 1, and diagnosis information is show in Fig.1, while more details of feature information examples could be found in Fig. 2.



(a) (b)

Figure 1: (a) Histogram of Diagnosis Distribution, (b) Scatter plot of radius_mean and texture_mean

Here to get a general view of the dataset, Fig. 1(a) shows Diagnosis distribution of dataset, and Fig.1(b) shows first two features plot classified by diagnosis value (M / B).

Table 1. General numeric description of first six features

	radius_mean	texture_mean	perimeter_mean	area_mean	smoothness_mean	compactness_mean
count	569	569	569	569	569	569
mean	14.13	19.29	91.97	654.89	0.10	0.10
std	3.52	4.301	24.30	351.91	0.01	0.05
min	6.98	9.71	43.79	143.5	0.05	0.02
25%	11.7	16.17	75.17	420.3	0.09	0.06
50%	13.37	18.84	86.24	551.1	0.10	0.09
75%	15.78	21.8	104.1	782.7	0.11	0.13
max	28.11	39.28	188.5	2501	0.16	0.35

These are all real-valued features computed for each cell nucleus, from the graph and data visualization, we could observe that diagnosis value (M / B) are closed related to the shape or value of cell nucleus. Thus, we could apply machine learning algorithm to train an optimized model for the data set, then test our trained model within the dataset.

Table 2. Summary of Data Set

Class Distribution	Benign: 357, 62.7%		
	Malignant: 212, 37.3%		
Number of Instances	569		
Features	**Attribute**		**Type**
	1	radius (mean of distances from center to points on the perimeter)	Numeric
	2	texture (standard deviation of gray-scale values)	Numeric
	3	perimeter	Numeric
	4	area	Numeric
	5	smoothness (local variation in radius lengths)	Numeric
	6	compactness (perimeter^2 / area – 1.0)	Numeric
	7	concavity (severity of concave portions of the contour)	Numeric
	8	concave points (number of concave portions of the contour)	Numeric
	9	symmetry	Numeric
	10	fractal dimension (“coastline approximation” – 1)	Numeric
	11	ID number	Numeric
	12	Diagnosis	Nominal
Missing Values	None		

5. Anticipated Technical Approach/Methods

Original dataset before being publically available, there are missing points. However, this dataset has been revised, so it could be considered ‘noise-free’ and has none missing feature values.
We typically follow four steps to do data analysis for a data set.

- Step 1: Data Pre-processing
- Step 2: Data Visualization
- Step 3: Model Training and Classification
- Step 4: Performance Evaluation

5.1 Data Pre-processing

First step of the dataset analysis would be pre-processing the data. A lot of factors affect the success of Machine Learning (ML) on a given task. The representation and quality of the instance data is first and foremost. If there is much irrelevant and redundant information present or noisy and unreliable data, then knowledge discovery during the training phase is more difficult [13] [14] [5] .

Though it was claimed there were no missing values, we need to be meticulous when we are dealing with data. The pre-processing will focus on managing the missing attributes, the unbalanced data (outlier values) and the number of attributes used to train the classifier.

5.1.1 Cleaning Outliers

Outlier is a value that lies in a data series on its extremes, which is either very small or large and thus can affect the overall observation made from the data series.

Fig. 2 simply shows a Box Plot Diagram to identify outliers.

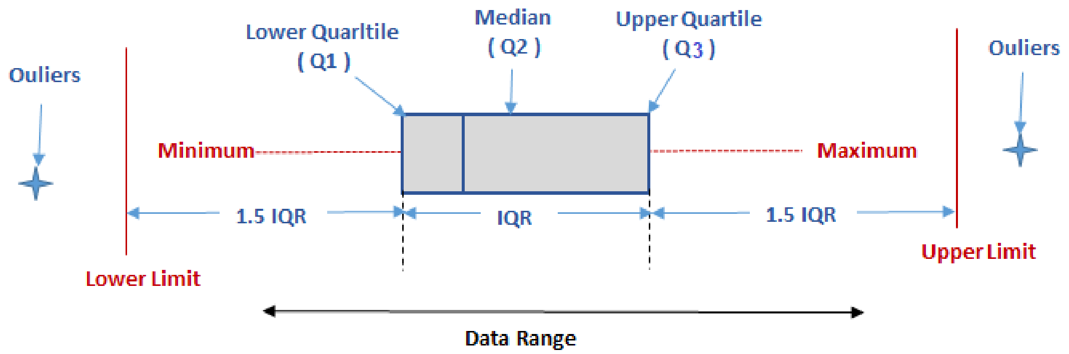


Figure 2: Outlier Definition

Box plot diagram is also termed as Whisker’s plot. It is a graphical method typically depicted by quartiles and inter quartiles that helps in defining the upper outer limit and lower outer limit beyond which any data lying will be considered as outliers.

The box plot uses the median (Q_2), the lower (Q_1) and upper(Q_3) quartiles (defined as the 25th and 75th percentiles). The Interquartile range (IQR) is the spread of the middle 50% of the data values (Q_3-Q_1).

$$IQR = Q_3 - Q_1 \quad (1)$$

$$Lower\ Limit = Q_1 - 1.5*IQR \quad (2)$$

$$\mathit{Upper\ Limit} = Q_3 + 1.5 * \mathit{IQR} \qquad (3)$$

$$\mathit{Lower\ Outer\ Limit} = Q_1 - 3 * \mathit{IQR} \qquad (4)$$

$$\mathit{Upper\ Outer\ Limit} = Q_3 + 3 * \mathit{IQR}$$

Here we would remove rows which contain value outside of 3 standard deviations.

5.1.2 Data Normalization

Normalization is a "**scaling down**" transformation of features. This works well when the difference between the maximum and minimum values of a feature is large, e.g. 0 and 1000. When normalization is performed the value magnitudes and scaled to appreciably low values. For some classification algorithms like (neural network and KNN), data normalization plays an essential part. The three most common methods for data normalization are [13] :

1) **Min-Max scaling**

It is also known as min-max normalization, which rescales data to have values in [0, 1] or [-1, 1]. Selecting the target range depends on the nature of the data. One possible formula to achieve this is:

$$x_{new} = \frac{x - x_{min}}{x_{max} - x_{min}} \qquad (5)$$

Where x is an original value, x_{new} is the normalized value.

2) Mean normalization

$$x_{new} = \frac{x - \mu}{x_{max} - x_{min}} \qquad (6)$$

Where x is an original value, x_{new} is the normalized value, μ is the mean (average).

3) Z-score normalization

The result of Z-score normalization is that the features will be rescaled so that they'll have the properties of a standard normal distribution with $\mu = 0$ and $\sigma = 1$. Standard scores (also called z scores) of the instances are computed as follows:

$$x_{new} = \frac{x - \mu}{\sigma} \qquad (7)$$

Where x is an original value, x_{new} is the normalized value, μ is the mean (average) and σ is the standard deviation from the mean.

Standardizing the features so that they are centered around 0 with a standard deviation of 1 is not only important if we are comparing measurements that have different units, but it is also a general requirement for many machine learning algorithms.

In this project, Mean Normalization is used to scale the data set.

5.2 Data Visualization

Second step would be data visualization. As we have 30 features, it would be complex and time-costing to train such multi-features model. We could use PCA and t-SNE for Data Visualization.

5.2.1 Principal Components Analysis (PCA)

PCA is a technique for reducing the number of dimensions in a dataset whilst retaining most information. It is a method of spectral clustering. Spectral clustering is related to kernel principal components, a non-linear version of linear principal components. Standard linear principal components (PCA) are obtained from the eigenvectors of the covariance matrix and give directions in which the data have maximal variance [14] . It is using the correlation between some dimensions and tries to provide a minimum number of variables that keeps the maximum amount of variation or information about how the original data is distributed.

One of the most important applications of PCA is for speeding up machine learning algorithms. It does not do this using guesswork but using hard mathematics and it uses something known as the eigenvalues and eigenvectors of the data-matrix. These eigenvectors of the covariance matrix have the property that they point along the major directions of variation in the data. These are the directions of maximum variation in a dataset.

5.2.2 T-Distributed Stochastic Neighboring Entities (t-SNE)

t-Distributed Stochastic Neighbor Embedding (t-SNE) is another technique for dimensionality reduction and is particularly well suited for the visualization of high-dimensional datasets. Contrary to PCA it is not a mathematical technique but a probablistic one. t-Distributed stochastic neighbor embedding (t-SNE) minimizes the divergence between two distributions: a distribution that measures pairwise similarities of the input objects and a distribution that measures pairwise similarities of the corresponding low-dimensional points in the embedding [15] .

5.3 Model Training and Classification

Third step would be using different data mining algorithms to develop prediction models. We could apply the algorithm we learn in class like KNN, or other applicable algorithms like (Naïve Bayes, RBF Network, J48, Support Vector Machine, and Decision tree) to solve the diagnosis problem. As the dataset is publically available and the same, previous work is replicable. The main reason for the project is to learn, thus I think reproduce previous work should be meaningful.

5.3.1 K-Nearest Neighbors (KNN) classification

The K-Nearest Neighbors (KNN) algorithm is a type of supervised machine learning algorithms [16] ,[17] , and it is one of the oldest and simplest methods of pattern classification [18] . KNN classifies instances based on their similarity, as it is extremely easy to implement in its most basic form, it is one of the most popular algorithms for pattern recognition. It is a type of Lazy learning algorithms where the function is only approximated locally and all computation is deferred until classification. An object is classified by a majority of its neighbors. KNN is a non-parametric learning algorithm, which means that it doesn’t assume anything about the underlying data.

Typically, most KNN classifiers use simple Euclidean distances to measure the dissimilarities between examples represented as vector inputs [18] .

Let $\{(\overrightarrow{x_i}, y_i)\}_{i=1}^{n}$ denote a training set of n labeled instances with inputs $\overrightarrow{x_i} \in \mathrm{R}^d$ and discrete (but not necessarily binary) class labels y_i. We use the binary matrix $y_{ij} \in \{0,1\}$ to indicate whether or not the y_i and y_j match. The squared distances are computed by a linear transformation $\mathrm{L}: \mathrm{R}^d \to \mathrm{R}^d$:

$$\mathrm{D}(\overrightarrow{x_i}, \overrightarrow{x_j}) = \|\mathrm{L}(\overrightarrow{x_i} - \overrightarrow{x_j})\|^2 \tag{8}$$

Computed by eq. (8), we could specify k "target" neighbors for each input $\overrightarrow{x_i}$. The target neighbors can simply be identified as k nearest neighbors, determined by Euclidean distance, that share the same label y_i. KNN decision statistic is computed as:

$$\lambda(x) = \frac{1}{k}\sum 1(\text{or } M) \text{ for } k \text{ nearest points} \tag{9}$$

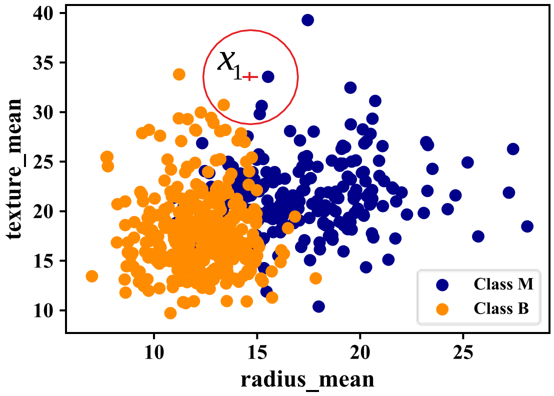


Figure 3: KNN decision statistic example

From Fig. 3, assume the threshold as β, then computed by eq. (9), we have $\lambda(x_1)$ = 3/4, when $\lambda(x_1) \geq \beta$, we classified input $\overrightarrow{x_1}$ as Class M, and when $\lambda(x_1) < \beta$, we classified its label as Class B. The majority decision boundary is defined when we set $\beta = 0.5$.

5.3.2 Logistic Regression

In supervised machine learning, assume we have some data denoted as *X* which is labeled as y. The goal is to learn a mapping f(*X*) -> *y*.

If *y* is a continuous variable, this type of procedure is known as regression. However, if *y* takes on discrete values, such as 0 or 1, or one of *k* values, this procedure is known as classification. In a special case where there are only 2 states that *y* can take, this is known as binary classification.

Logistic Regression is generally used for classification purposes, and it is one of the most popular binary classification algorithms and is related to linear regression. Unlike Linear Regression, the dependent variable can take a limited number of values only i.e, the dependent variable is categorical. It belongs to a class of models known as generalized linear models or GLMs [20] .

The logistic function derives its name:

$$\sigma(x) = \frac{e^x}{1+e^x} = \frac{1}{1+e^{-x}} \tag{10}$$

Logistic Regression provides a functional form $f(X)$ to expression *P*(*y*|*X*). The model is usually formulated mathematically by relating the probability of some event, *E*, occurring, conditional on a vector, $\vec{x}$, through the functional form of a logistic cdf. Thus, we have [19] :

$$p(\vec{x}) \equiv \Pr\{E \mid \vec{x}\} = \frac{1}{1+\exp\{-\alpha - \beta'\vec{x}\}} \tag{11}$$

where (α, β) are unknown parameters that are estimated from the data. This model may be used for classifying an object into one of two populations by letting E denote the event that object belongs to the first population and letting $\vec{x}$ denote a profile vector of attributes of the object to be classified [20] .

For normal discrimination or classification problem which is formulated by assuming that the two populations are multivariate normal with equal convariance matrices, $\sum$, and that the costs of misclassification are equal. If $\overrightarrow{u_1}$ and $\overrightarrow{u_2}$ denote the mean vectors of the two populations, a likelihood ratio test readily yields the classification procedure to classify the object into the first population if

$$(\overrightarrow{u_1} - \overrightarrow{u_2})'\sum{}^{-1}\vec{x} + \frac{1}{2}(\overrightarrow{u_1} + \overrightarrow{u_2})'\sum{}^{-1}(\overrightarrow{u_2} - \overrightarrow{u_1})\ldots\log(q_2 / q_1) \tag{12}$$

Where (q_1, q_2) denote the prior classification probabilities. The parameters $(\overrightarrow{u_1}, \overrightarrow{u_2}, \Sigma)$ are estimated from the data, while (q_1, q_2) are assessed from the context [21] .

5.3.3 Bayes Classification

Bayes classifier is a probabilistic classifier based on applying Bayes' theorem with strong (naive) independence assumptions between the features. This conditional independence assumption rarely holds true in real world application, hence the characterization as Naïve yet the algorithm tends to perform well and learn rapidly in various supervised classification problems [22] [23] .

Bayes' Rule

$$P(\omega_j / \vec{x}) = \frac{p(\vec{x} / \omega_j)P(\omega_j)}{p(\vec{x})} = \frac{likelihood * prior}{evidence} \tag{13}$$

Where ω denote the state of nature (class label), $P(\omega)$ denote prior probability of class, and $p(\vec{x})$ denote probability density function (evidence), and it is the prior probability of predictor. We decide ω_1 if $P(\omega_1 / \vec{x}) >$ $P(\omega_2 / \vec{x})$, otherwise decide ω_2.

The probability of error is defined as:

$$P(error / \vec{x}) = \begin{cases} P(\omega_1 / \vec{x}) \text{ if we decide } \omega_2 \\ P(\omega_2 / \vec{x}) \text{ if we decide } \omega_2 \end{cases} \tag{14}$$

The Bayes rule is optimum, that is it minimizes the average probability error.

Expected Loss

Suppose we observe $\vec{x}$ and take action α_i, the expected risk with taking action α_i is defined as [23] :

$$R(\alpha_i / \vec{x}) = \sum_{j=1}^{c} \lambda(\alpha_i / \omega_j) P(\omega_j / \vec{x}) \tag{15}$$

Where *c* denote a set of categories ω_1, ω_2, …, ω_c, *l* denote a fininte set of actions α_1, α_2, …, α_l. The overall risk is defined as:

$$R = \int R(\alpha(\vec{x}) / \vec{x}) p(\vec{x}) d\vec{x} \tag{16}$$

Where $\alpha(\vec{x})$ is a general decision rule that determines which action α_1, α_2, … α_l to take for every $\vec{x}$.

The optimum decision rule is the Bayes rule.

Represent a classifier is through discriminant functions:

$$g_i(\vec{x}), \text{ i = 1, 2, …, c} \tag{17}$$

A feature vector *x* is assigned to class ω_i if:

$$g_i(\vec{x}) > g_j(\vec{x}), \text{ for all } j \neq i \tag{18}$$

Assuming the zero-one loss function:

$$g_i(\vec{x}) = P(\omega_i / \vec{x}) = p(\vec{x} / \omega_i) P(\omega_i) \tag{19}$$

$$\text{In}\, g_i(\vec{x}) = \text{In}\, p(\vec{x} / \omega_i) + \text{In}\, P(\omega_i)$$

As an example of two categories, we will decide ω_1 if $g(\vec{x})$ > 0; otherwise decide ω_2.

Multivariate Normal Density

A normal distribution over two or more variables (*d* variables/dimensions), the formal definition is [21]:

$$p(\vec{x}) = \frac{1}{(2\pi)^{d/2} |\sum|^{1/2}} \exp[-\frac{1}{2}(\vec{x} - \vec{u})^t \sum{}^{-1} (\vec{x} - u)] \tag{20}$$

$$u = \int_{-\infty}^{\infty} \vec{x} p(\vec{x}) d\vec{x} \tag{21}$$

$$\sum = \int (\vec{x} - \vec{u})(\vec{x} - \vec{u})^t p(\vec{x}) d\vec{x} \tag{22}$$

In (natural log) of eq. (19), it gives a general form for the discriminant functions:

$$g_i(\vec{x}) = -\frac{1}{2}(\vec{x} - \overrightarrow{u_i})^t \sum{}_i^{-1} (\vec{x} - \vec{u_i}) - \frac{d}{2} \text{In}\, 2\pi - \frac{1}{2} |\Sigma_i| + \text{In}\, P(\omega_i) \tag{23}$$

For arbitrary Σ_i, the decision boundaries are hyperquadrics: can be hyperplanes, hyperplane pairs, hyperspheres, hyperellipsoids, hyperparabaloids.

5.3.4 SVM (Support Vector Machine)

Support Vector Machine (SVM) is introduced by Vapnik et al. [24] . It is a very powerful method that has been applied in a wide variety of applications. The basic concept in SVM is the hyper plane classifier, or linear separability. Two basic ideas are applied to achieve linear separability, SVM

[3] : margin maximization and kernels that is, mapping input space to a higher-dimensional space (or feature space).

Here we will discuss Linear Support Vector Machine. Given a set of training data x_1, x_2, …, x_m, belonging to two different classes *y*. These training set is projected to the high dimensional space by the transformation ϕ. Then each hyperplane is the feature space H: $\omega \square \phi(x)+b$ separating the two classes must satisfy the following conditions [25] :

$$\omega \square \phi(x_i)+b \geqslant 1 \text{ if } y_i = 1 \tag{24}$$

$$\omega \square \phi(x_i)+b \leqslant -1 \text{ if } y_i = -1 \tag{25}$$

The optimal separating hyperplane is the one maximizing the margin M given by the equation:

$$M = \min_{x_i|y_i=1}[\frac{\omega \cdot \phi(x_i)+b}{|\omega|}] - \max_{x_i|y_i=-1}[\frac{\omega \cdot \phi(x_i)+b}{|\omega|}] \tag{26}$$

To maximize the margin M, one need to minimize

$$\Phi = \frac{\omega^2}{2} \tag{27}$$

Using the Lagrange multipliers and the Kuhn-Tucker theorem. The problem can be translated to the following dual problem [24] :

Minimize:

$$\sum_{i=1}^{m} \alpha_i - \frac{1}{2}\sum_{i,j=1}^{m} \alpha_i \alpha_j y_i y_j \phi(x_i)\cdot\phi(x_j) \tag{28}$$

Subject to the constraints:

$$\alpha_i \geqslant 0 \tag{29}$$

$$\sum_{i=1}^{m} \alpha_i y_i = 0 \tag{30}$$

where α_i are the Lagrangian multipliers with one Lagrange multiplier for each training data point. The support vectors are the point x_i for which α_i is strictly positive.

SVM has shown good performance in pattern classification problems, where labelled data for each class are available in the development phase.

5.3.5 Cross Validation

1) Train/Test Splits

The most common approach in machine learning to estimating how well a model works is to use the Training and Test set procedures [27] .

Usually we would follow the rules to operate train_test_split:

- Randomly select a percentage (usually ~80%) of the dataset for training
- Evaluate the model on the remaining (~20%) of the dataset

It is easy to implement but may be biased in some way if the training set and the testing set could not represent the typical dataset features. What is more, it can overestimate error as the model only uses some of the data to fit the model, and test on the other.

2) Cross-Validation

Cross-Validation is the most common technique that addresses some of the downsides of the training/test splits. Instead of splitting the data set once, cross-validation splits the data many times and averages the error over the splits.

The most common form of cross-validation is known as **k-fold** validation. In k-fold cross-validation, the data is split into k evenly-size groups, where k is usually 5 or 10. The general idea is:

- Shuffle the dataset randomly
- Split the dataset into k groups
- For each unique group:
    - Take the group as a hold our or test data set
    - Take the remaining groups as a training data set
    - Fit a model on the training set and evaluate it on the test set
    - Retain the evaluation score and discard the model
- Average the performance of the model using all groups model evaluation scores

The Cross-Validated error is estimated by [26] :

$$CV_{(k)} = \frac{1}{k}\sum_{i=1}^{k} MSE_i \tag{31}$$

K-fold is simple, and it generally results in a less biased or less optimistic estimate of the model skill than other methods.

5.3.6 Feature Selection

Feature selection is the process of removing redundant or irrelevant features from the original data set. This reduces the dimensionality of the data and enables learning algorithms to operate faster and more effectively [28] . Generally, features could be characterized as three types [29] :

- Relevant Features: These are features have influence on the output and their role can not be assumed by the rest.
- Irrelevant Features: These are features which have no influence on the output, and whose values are generated at random for each instance.
- Redundant Features: A redundancy exists whenever a feature can take the role of another, thus they are duplicated of other features and will cost of operation time.

Feature Selection algorithms in general have two components:

(1) A selection algorithm that generates proposed subsets of features and attempts to find an optimal subset.

(2) An evaluation algorithm or machine learning model that determines how ‘good’ the proposed feature subset is. This provides feedback to measure the selection algorithm.

Many variable selection algorithms include variable ranking as a principal or auxiliary selection mechanism because of its simplicity, scalability, and good empirical success. Several approaches to the variable selection problem using information theoretic criteria have been proposed [30] . Many rely on empirical estimates of the mutual information between each variable and the target:

$$I(i)=\int_{x_i}\int_{y} p(\vec{x_i},y)\log\frac{p(\vec{x_i},y)}{p(\vec{x_i})p(y)}d\vec{x}dy \tag{32}$$

Where $p(\vec{x_i})$ and $p(y)$ are the probability densities of $\vec{x_i}$ and *y*, and $p(\vec{x_i},y)$ is the joint density. The criterion $I(i)$ is a measure of dependency between the density of variable $\vec{x_i}$ and the density of the target *y*. The case of discrete or nominal variables is probable the easiest case to find out $p(\vec{x_i})$, $p(y)$, $p(\vec{x_i},y)$ because the integral becomes a sum:

$$I(i)=\sum_{x_i}\sum_{y}P(X=\vec{x_i},Y=y)\log\frac{P(X=\vec{x_i},Y=y)}{P(X=\vec{x_i})P(Y=y)} \tag{33}$$

The probabilities are then estimated from frequency counts.

In Python, we could use headmap to find out the features which has great influence of the output.

5.4 Model Performance Evaluation

Fourth step would be model performance measuring. We have many influencing factors to define how good a training model is. Normally we often use precision, recall, F1 Score, ROC Curve, AUC (Area under Curve), etc. to evaluate machine learning algorithms. To perform evaluation, we could apply classification accuracy, sensitivity, specificity, positive and negative predictor values and confusion matrix to evaluate different model performance.

5.4.1 Confusion Matrix

To understand how well a model is actually performing, we will need to look at the Confusion Matrix, typically a supervised learning one. It is a table with two rows and two columns that reports the number of False Positives, False Negatives, True Positives, and True negatives, shown as Table 3. This allows more detailed analysis than mere proportion of correct classifications (accuracy).

Table 3: Confusion Matrix

		True Condition	
		Positive	Negative
Predicted Condition	Positive	**True Positive**	**False Positive**
	Negative	**False Negative**	**True Negative**

- True Positive (**TP**)
  A true positive occurs when the true value is positive, and the predicted value is also positive.

- True Negative (**TN**)
  A true negative occurs when the true value is negative, and the predicted value is also negative.

- False Positive (**FP**)
A false positive occurs when the true value is negative, but the predicted value is positive.

- False Negative (**FN**)
A false positive occurs when the true value is negative, but the predicted value is positive.

The confusion matrix in itself is not a performance measure as such, but most of performance metrics are based on Confusion Matrix. With those definitions of its inside terms, we can look at different methods for evaluating binary classification models.

5.4.2 Accuracy

Accuracy in classification problems is the number of correct predictions made by the model over all predictions made. It is equivalent to:

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{34}$$

Accuracy is a good measure when the target variable classes in the data are nearly balanced. It should NEVER be used as a measure when the target variable classes are a majority of one class.

In Python, accuracy is referred as score, thus in the performance evaluation, score is used in this report.

5.4.3 ROC & AUC

1) Threshold

For classification models that output a probability, we define a threshold to compute the Confusion Matrix. For example, the model for logistic regression is:

$$P(Y = 1 | X) = \frac{e^{\mathbf{w}^T \mathbf{x}}}{1 + e^{\mathbf{w}^T \mathbf{x}}} \tag{35}$$

By definition, the Confusion Matrix need to have value 1 or 0. P(Y=1|X) is a probability in [0,1], thus we need to choose a threshold to classify the binary value of P(Y=1|X). Assume β as the threshold, so that the predictions in logistic regression (or any other probabilistic classifier) are 1 if $\hat{y}$ is equal or greater than β and 0 otherwise.

By default, 0.5 is the choice for logistic regression classifier.

2) Receiver Operating Characteristic (ROC)

Different values of a threshold β can result in different confusion matrices. The ROC curve is created by plotting the True Positive rate (TPR) against the False Positive rate (FPR) at various threshold settings, where TPR is on y-axis and FPR is on the x-axis.

ROC curves are a nice way to see how any predictive model can distinguish between the true positives and negatives

3) Area Under the Curve (AUC)

To summarize the characteristics of a given Receiver Operating Characteristic plot, we use the Area Under the Receiver Operating Characteristic (AUROC) or simply Area Under the Curve (AUC).

The AUC, as its name suggests, is simply the area under the ROC curve. It tells how much model is capable of distinguish between classes. Higher the AUC, better the model. An excellent model has AUC near to 1 which means good measure of separability. The diagonal line shows the worst possible classifier and has an associated AUC of 0.5.

5.4.4 Precision & Recall

The Precision, or Positive Predictive Value of a classifier, represents, of the examples which the classifier has predicted to be positive, what percentage are actually positive. It is computed by:

$$Precision = \frac{TP}{TP+FP} \tag{36}$$

Precision is only available after a threshold has already been chosen.

Recall or Sensitivity, is also known as the True Positive Rate, is computed by:

$$Recall = \frac{TP}{TP+FN} \tag{37}$$

It represents the proportion of positive instances (in total) that are classified as positive.

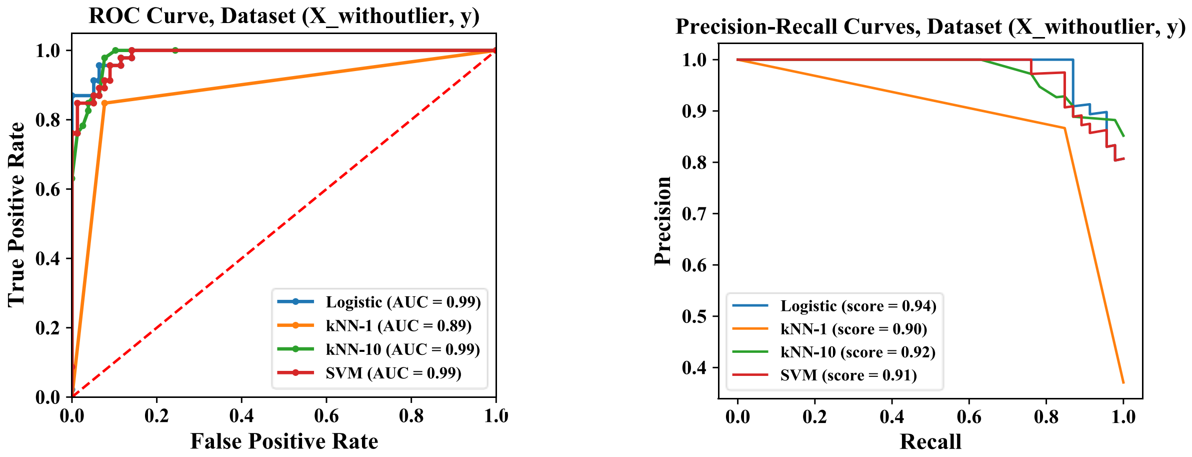
Recall gives us information about a classifier's performance with respect to false negatives (how many did we miss), while precision gives us information about its performance with respect to false positives (how many did we caught). Therefore, it depends on what information we prefer to get when we decide which measure to use.

The analogue to the Receiver Operating Characteristic is known as the Precision-Recall Curve.

6. Result Discussion

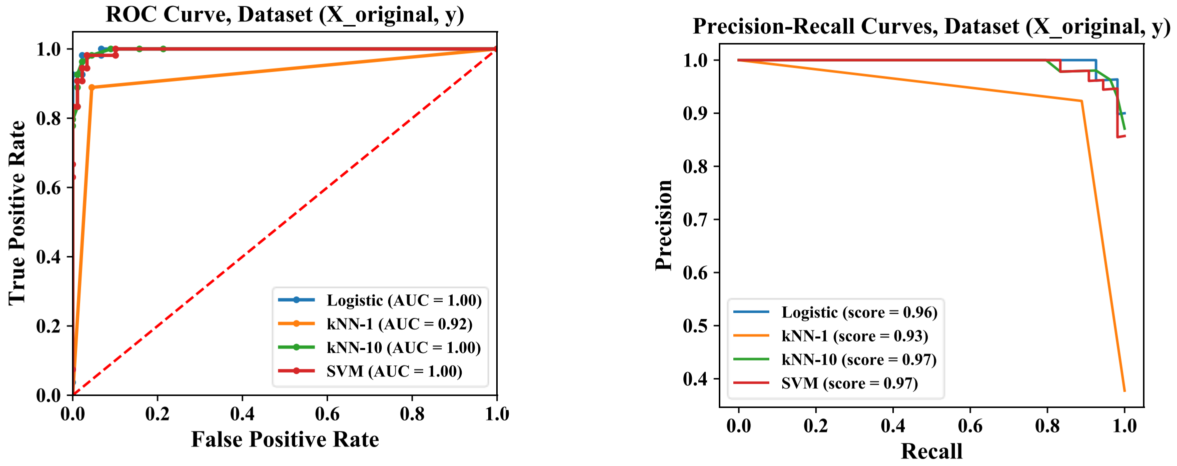
6.1 Cleaning Outlier

Dealing with outliers requires knowledge about the outlier, the dataset and possibly domain knowledge. If the outlier is a data processing or entry error, it can generally be removed, or replaced with say, the mean (without the outlier). If errors have been ruled out, then the outlier might be legitimate value. We could run several models and compare the results to make our decisions.



(a) (b)

Figure 4: (a) ROC Curve of data set without outliers, (b) Precision-Recall Curves of data set without Outliers



(a) (b)

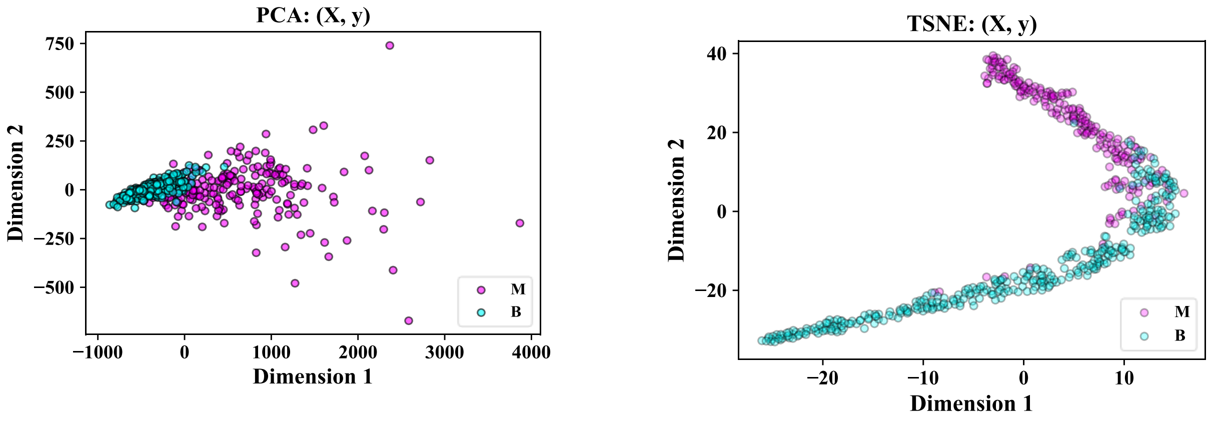
Figure 5: (a) ROC Curve of data set with outliers, (b) Precision-Recall Curves of data set with outliers

From Fig. 4 and Fig. 5, we could see that for most of the four models, the data set included outliers gets better performance on both AUC and score. The reason is that our dataset is presumably noise-free, thus there is no need to clean outliers. Every data is a legitimate value.

Noted: According to the analysis above, we did the following model training and testing without cleaning outliers.

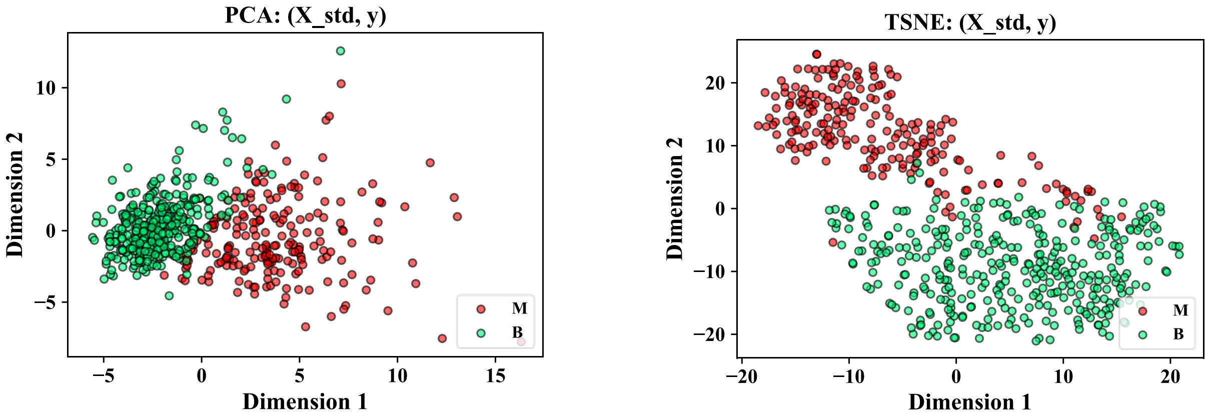
6.2 Data Normalization

Normalization is a technique often applied as part of data preparation for machine learning. The goal of normalization is to change the values of numeric columns in the dataset to a common scale, without distorting differences in the ranges of values. Not every algorithm or data set needs to do data normalization.



(a) (b)

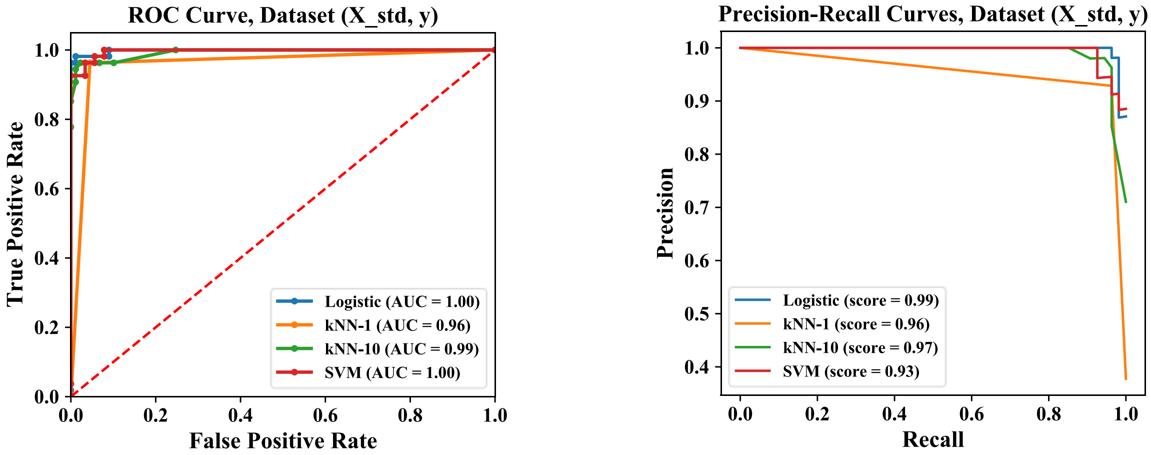
Figure 6: (a) PCA scatter Plot of data set before normalization, (b) TSNE scatter Plot of data set before normalization



(a) (b)

Figure 7: (a) PCA scatter Plot of data set after normalization, (b) TSNE scatter Plot of data set after normalization

Compared Fig. 6 with Fig. 7, we could see that after data normalization, the data points become more distinguishable as they are distributed sparsely and more balanced. It becomes more obvious that there is distinguishable boundary between the data class, and these data are clustered.



(a) (b)

Figure 8: (a) ROC Curve of data set after normalization, (b) Precision-Recall of data set after normalization

Besides, compared Fig. 5 and Fig. 8, we could see that the overall performance of AUC under ROC curve and scores with Precision-Recall curves after data normalization is tentatively better than before, except for SVM whose score performance has been reduced a bit.

Therefore, we could see that there are **two benefits** for data normalization:

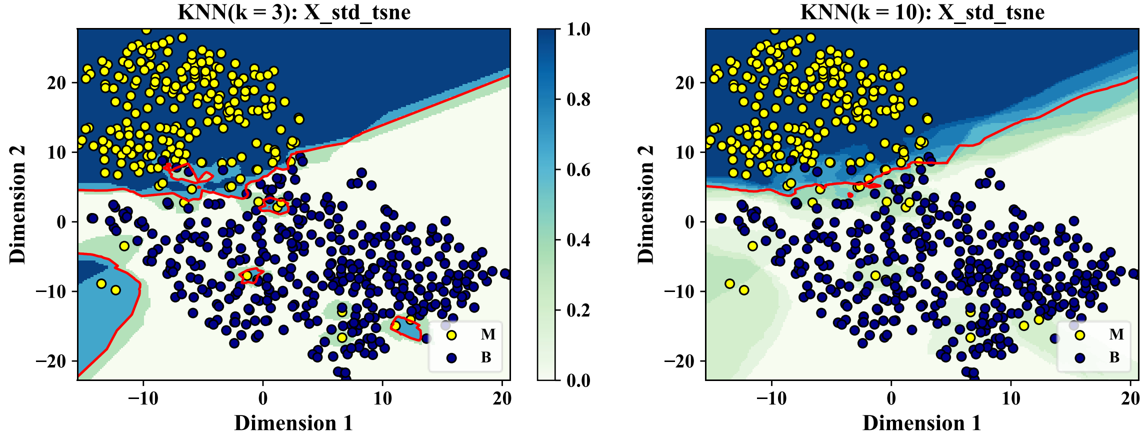
- Data visualization is improved, which means classes becomes distinct
- Models performance becomes better, which means higher AUC, score, etc.

Actually, these two benefits are bound up with each other. If we get good classifications, that usually means the classes are clustered and distinguishable.

Noted: According to the analysis above, we did the following model training and testing with data set which has been normalized.

6.3 Decision Statistic Surface

In general, a pattern classifier partitions the feature space into volumes called decision regions. Decision boundaries or decision surfaces are the surfaces that separated the decision regions. It is a good way of classification visualization to explore the data and compare different classifiers.

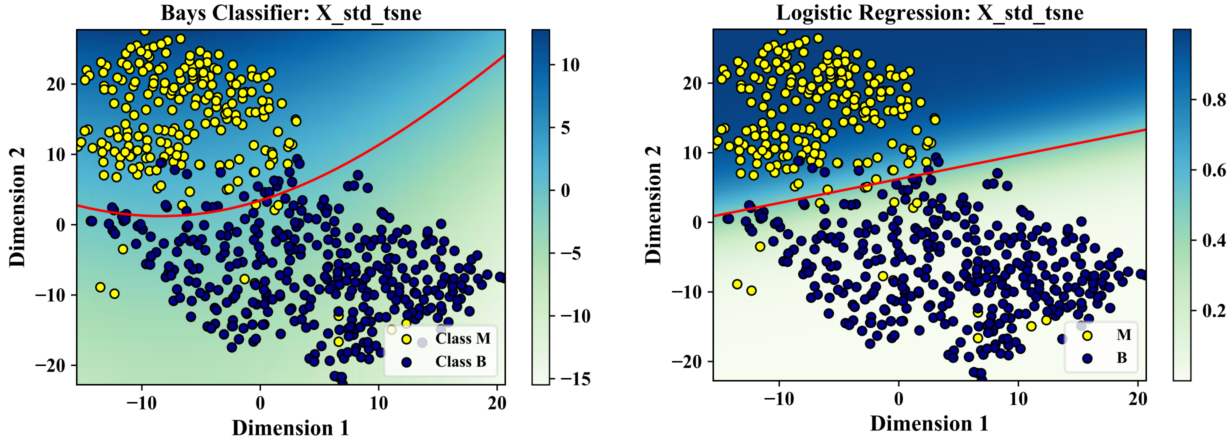


(a) (b)

Figure 9: (a) Decision statistic surfaces for KNN (k = 1), (b) Decision statistic surfaces for KNN (k = 10)

Fig. 9 shows us the KNN decision boundaries given different k values. As k increase, more and more neighbors are taken into consideration to infer the class of a test point. When k is small, there is overfitting problems because the classifier has no flexibility to model the boundary between the class. However, if k becomes really large, the test point might include more training points from the opposite class than from the same class, which would result in underfitting problems.

Fig. 9 also shows that the majority decision boundary when k =10 is smoother than that when k = 3.



(a) (b)

Figure 10: (a) Decision statistic surfaces for Bayes Classifier, (b) Decision statistic surfaces for Logistic Regression

Fig. 10 (a) shows the decision statistic surface and majority decision boundary of bayes classifier. As we make no simplifying assumptions regarding the covariance structure (i.e., the features may be dependent, and the covariance matrices are unique), the discriminant function eq. (23) are quadratic function.

When we make no assumptions, both classes are using their own mean vector and their own covariance matrices, which make the most correct prediction of data distribution. The decision boundaries are hyperquadrics.

Fig. 10 (b) shows the decision statistic surface and majority decision boundary of Logistic Regression. According to the Logistic Regression discriminant function or classification probability function, we could see that Logistic Discriminant doesn't take the outliers into much account, which makes it a diagonal linear boundary.

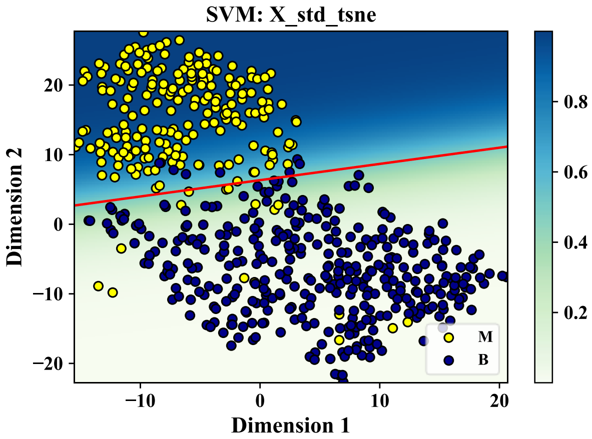


Figure 11: Decision statistic surfaces for SVM

Fig. 11 shows the decision statistic surface and majority decision boundary of SVM. SVM projects the input data into a kernel space, then it builds a linear model in this kernel space. A classification SVM model attempts to separate the target classes with the widest possible margin. We could see from the graph, that a linear boundary seperate the Malicious and Benign class.

From the above decision statistic surfaces graphs, we could find that it is helpful to learn the differences between different classifiers. If you have problems understand underlying mechanisms, plotting the decision statistic surfaces would help you gain more knowledge.

6.4 Cross Validation

The goal of cross-validation is to estimate the expected level of fit of a model to a data set that is independent of the data that were used to train the model. It can be used to estimate any quantitative measure of fit that is appropriate for the data and model.

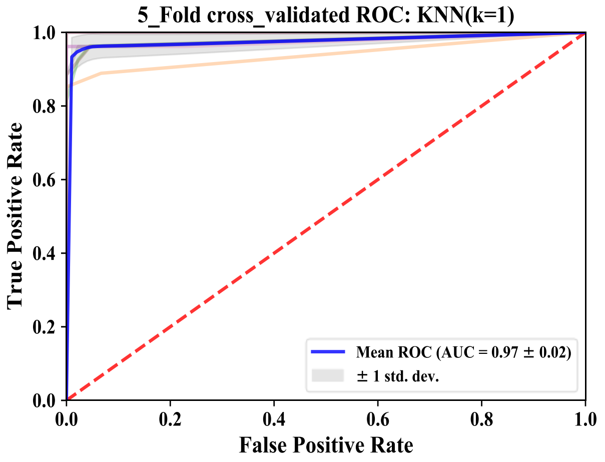


Fig. 12 shows the ROC performance of 5-Fold cross validation under KNN algorithm when k is set to 1. We estimate our model with the mean of every iteration ROC which was shown as blue line. For each group, the ROC is show as semitransparent lines.

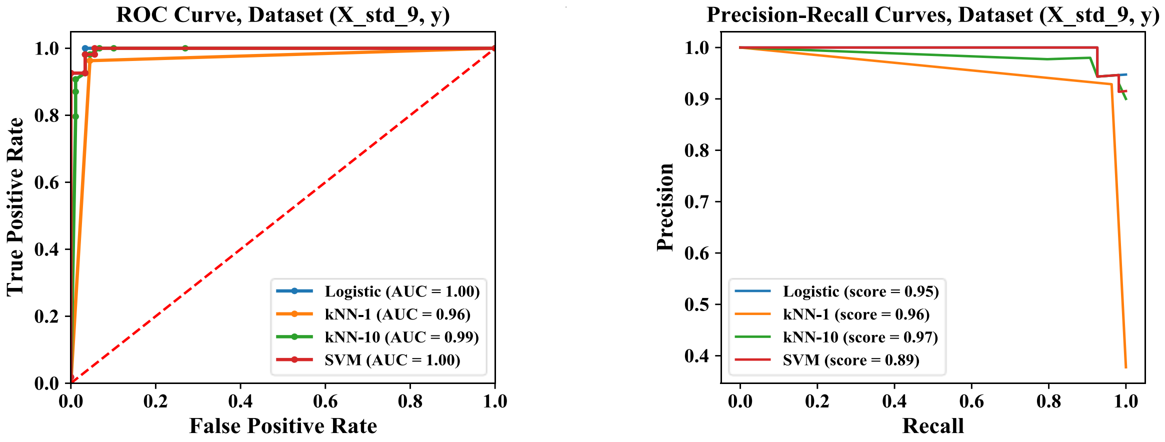
Figure 12: ROC Curve of 5-Fold on KNN (k=1)

We could see that there are three ROCs which are lower than the mean ROC. That means AUC under these groups are smaller. This is because we can guarantee that train-and-split reach a balanced dataset which typically represent out dataset for a single time.

According to the above analysis, we could see that cross-validation helps us receive a better ROC and higher AUC. Therefore, the model would perform better when we have a new data in terms of accuracy of its predictions.

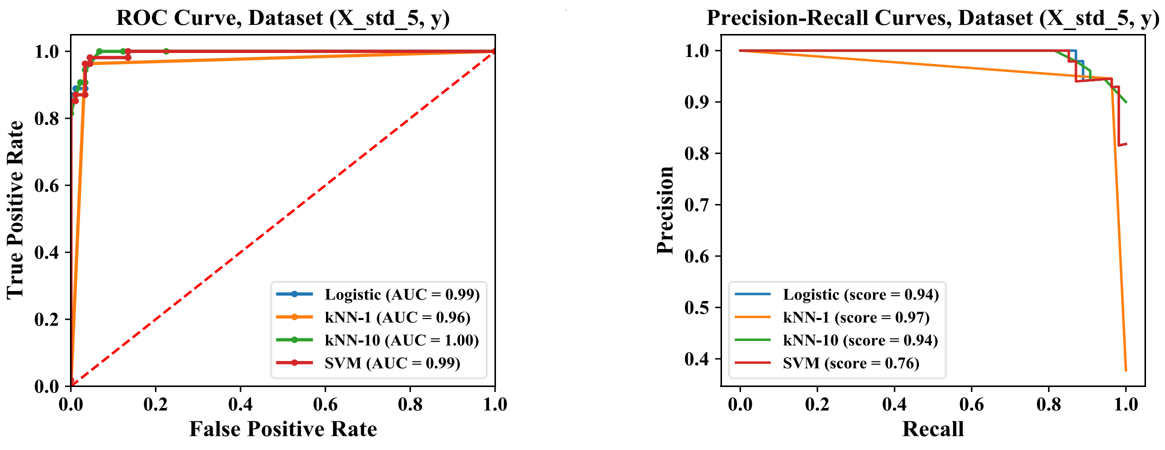
6.5 Feature Selection

Feature selection provides the classifiers to be fast, cost-effective, and more accurate, which makes it becomes a hot research area on machine learning and data mining.



(a) (b)

Figure 13: (a) ROC Curve of data set with 9 features, (b) Precision-Recall of data set with 9 features



(a) (b)

Figure 14: (a) ROC Curve of data set with 5 features, (b) Precision-Recall of data set with 5 features

Compared with both ROC and Precision-Recall for data set with full (30) features (Fig. 5), 9 features (Fig. 13), 5 features (Fig.14), we could see that the overall performance of AUC and Accuracy are tentatively better when we include more features.

We noticed that for KNN (k=1) full features did the worst performance, that is because with full features and k = 1, the model is highly possible to result in overfitting problems. And for SVM, the performance decreased obviously with feature decreasing. Own to its nature, SVM are sensitive to feature selection.

Even though there are some difference for AUC and Accuracy performance for different feature sets, we observed that the difference is tentatively minor if we select the features wisely. Here we could use 9-feature for our model, which has similar results and will reduce our computing work and simplify our model.

9 Features: radius_mean, perimeter_mean, area_mean, concavity_mean, concave points mean, radius_worst, perimeter_worst, area_worst, concave points_worst.

5 Features: concave points_mean, perimeter_worst, concave points_worst.

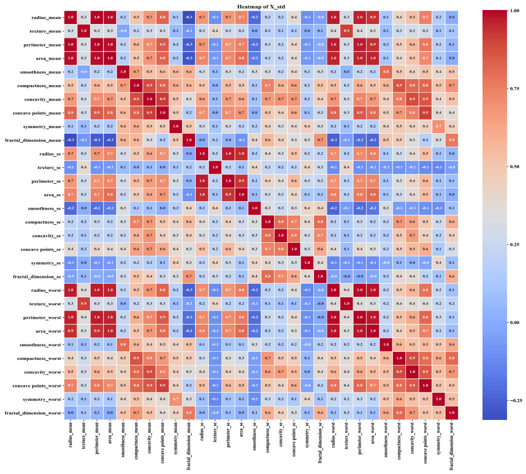


Fig. 15 shows the heatmap for the dataset we use for feature selection. We could see that the features selected by feature selection algorithms are aligned with the heatmap, that is the more red color, which means the feature are of vital importance to the output.

Besides, the 5 features are also part of 9 features, which is correct because the selection algorithm order the features in ascending order according to their influences.

Figure 15: Heatmap of Full Features Dataset (full image)

7. Conclusions

From the discussion part, we learn that there are a lot of factors influencing the performance of machine learning model. Raw data characteristics, data preprocessing, training algorithms and measurements for performance are all crucial to evaluate a model. What is more, there are different requirements or preferences for different applications. When we design or evaluate a model, we need some domain knowledge or consult professional expects to get the needs of model application. It is the people who will use the model for prediction to decide how ‘good’ the model is.

In our case, we are working on clinical data sets, which means it is relative better if we get more recall because we should avoid missing positive instances for patients. Details about the data analysis are explained in Discussion. To summarize the pipeline for my data analysis, I extracted the numeric values for trend viewing.

Data Preprocessing

In practice, the data we collected usually include some abnormal values due to instrument measurement accuracy, carelessness of recorders, or unavoidable factors. Thus I first tried to exclude outliers for the dataset. Here Fig. 16(a) shows that for our dataset the performance of both AUC and accuracy tend to be better when we include outlier. The reason is because the raw data set we are using are pretty clean, which means there are no null values, and abnormal values are exclude already.

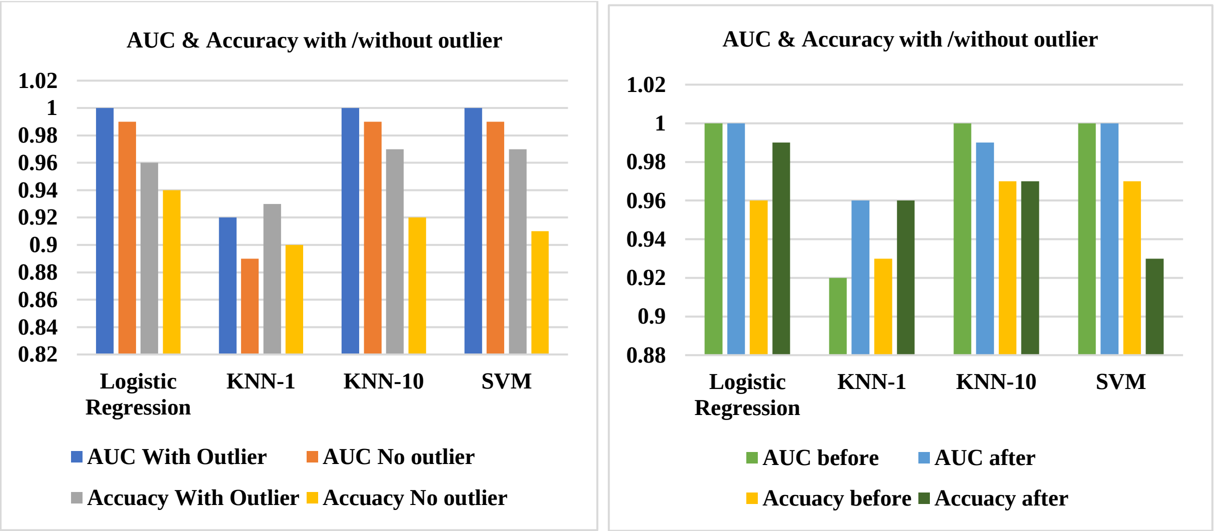


Figure 16: (a) AUC & Accuracy with or without outliers, (b) AUC & Accuracy before or after normalization

However, if the data set range is too big for features, or there are large number of outliers, we should consider cleaning outlier before we start modelling.

Data normalization is another integral preprocessing step, which could help us get better performance. From Fig. 16(b) we could see that the overall performance of AUC and accuracy are better after data normalization, except that KNN-1 did worse on AUC. This is because the overfitting problem for k =1. Previously Fig. 5 and Fig. 8 also shows that after data normalization, precision-recall curve tends to be better.

Data normalization is a necessary step especially when the data features has wide range or large min and max values. Because it helps us scale down the data set to 0~1.

What is more, referring to Fig. 6 and Fig. 7, data normalization also helps better visualize clusters for the dataset using PCA and t-SNE. That is because after data scaling, the data is distributed in a relatively small scope (-1, 1) or (0,1).

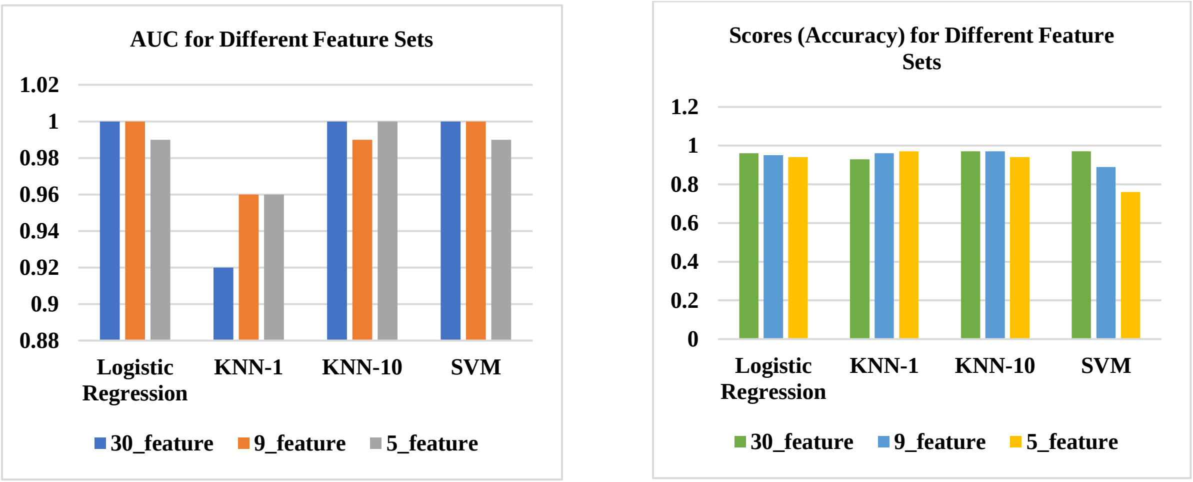
Model Training and Classification

In this project, we performed KNN, Logistic Regression, Bayes Classifier, SVM, Cross Validation, Feature Selection on our data set. Because the dataset we used is clearly clustered and looks ordinary distributed, not one class circled by another.

According to previously decision statistic surface and majority decision boundary plots, we notice that both Logistic Regression and SVM have similar performance, they are both linear boundaries, while KNN displays irregular boundaries and Bayes Classier boundary is hyperquadric.

Logistic Regression, SVM, KNN-10 did better performance than KNN-1, that is because KNN-1 has overfitting problems.

Previously we also notice that cross validation helps us get better performance on both ROC and Precision-Recall. It is usually set to 5 or 10 folds for machine learning. Because this will help avoid imbalanced features distribution, and after several folds, the error is reduced. Too many folds would make the model complex and cost operation time.



(a) (b)

Figure 17: (a) AUC for Different Feature Sets, (b) Accuracy for Different Feature Sets

Finally, we did some research on feature selections. Fig. 17 shows the bar chart for both AUC and Accuracy under different features, more details could be referred in Discussion part. Our data set has full 30 features, when we reduce to 9, and 5 features, we noticed that we get acceptable good performance. Because under feature selection algorithms, we selected the primary features which influence the output in ascending order.

Here 9 features tend to be perfect for our dataset. How many features should be used is decided by the nature of our data and the need for models. Our data set could be divided into three fields and each field contains ten features, what is more other two fields are generated from the first field data. Therefore, we get 9_feature data set which is enough to produce good results.

Model Performance Evaluation

Aside of ROC curve and AUC, I also plot the Precision-Recall curve and accuracy, that is because in hospital, recall is also very important. We should try to get high recall then try to get high precision.

Summary

Overall, the performance of the models or classifiers are very high, one of the reasons is that the data is typical used for machine learning, therefore it has already preprocessed by several researchers. The data is neat, and mid-size, and the binary classes are relatively balanced than other clinical data sets (M: 37.3%, B: 62.7%), as normally malicious class or positive is really small in hospitals. Hence, we could see that the data structure plays a really important role for machine learning model.

What potential research or what I am really interested in is what if we have some really noisy data, or greatly imbalanced? Are there any differences for the data process if we get some raw data in practice? How should we implement the model in real world and make it provide feedback? As far as I know, the hospital information log system is like a disaster. There are a lot of information, several versions of systems and different hospital might use different systems.

These are all interesting and challenging areas.

8. Reference

[1] https://www.kaggle.com/uciml/breast-cancer-wisconsin-data/discussion/62297

[2] http://archive.ics.uci.edu/ml/datasets/Breast+Cancer+Wisconsin+%28Diagnostic%29

[3] Salama, G.I., Abdelhalim, M. and Zeid, M.A.E., 2012. Breast cancer diagnosis on three different datasets using multi-classifiers. Breast Cancer (WDBC), 32(569), p.2.

[4] Coughlin, Steven S., and Donatus U. Ekwueme. Breast cancer as a global health concern. Cancer epidemiology 33.5 (2009): 315-318.

[5] Borges, L.R., 1989. Analysis of the Wisconsin Breast Cancer Dataset and Machine Learning for Breast Cancer Detection. Group, 1(369).

[6] Bhattacherjee, Aindrila, et al. Classification approach for breast cancer detection using back propagation neural network: a study. Biomedical image analysis and mining techniques for improved health outcomes. IGI Global, 2016. 210-221.

[7] Shaikh, Tawseef Ayoub, and Rashid Ali. Applying Machine Learning Algorithms for Early Diagnosis and Prediction of Breast Cancer Risk. *Proceedings of 2nd International Conference on Communication, Computing and Networking*. Springer, Singapore, 2019.

[8] Chaurasia, Vikas, Saurabh Pal, and B. B. Tiwari. Prediction of benign and malignant breast cancer using data mining techniques. Journal of Algorithms & Computational Technology 12.2 (2018): 119-126.

[9] Sri, M. Navya, et al. A Comparative Analysis of Breast Cancer Data Set Using Different Classification Methods. Smart Intelligent Computing and Applications. Springer, Singapore, 2019. 175-181.

[10] Obaid, Omar Ibrahim, et al. Evaluating the Performance of Machine Learning Techniques in the Classification of Wisconsin Breast Cancer. International Journal of Engineering & Technology 7.4.36 (2018): 160-166.

[11] Henriksen, Emilie L., et al. The efficacy of using computer-aided detection (CAD) for detection of breast cancer in mammography screening: a systematic review. Acta Radiologica 60.1 (2019): 13-18.

[12] Anagnostopoulos, Ioannis, et al. The Wisconsin Breast Cancer Problem: Diagnosis and DFS time prognosis using probabilistic and generalised regression neural classifiers. *Oncology Reports, Special Issue Computational Analysis and Decision Support Systems in Oncology* (2005).

[13] Kotsiantis, S.B., Kanellopoulos, D. and Pintelas, P.E., 2006. Data preprocessing for supervised leaning. International Journal of Computer Science, 1(2), pp.111-117.

[14] Friedman, J., Hastie, T. and Tibshirani, R., 2001. *The elements of statistical learning* (Vol. 1, No. 10). New York: Springer series in statistics.

[15] Maaten, L.V.D. and Hinton, G., 2008. Visualizing data using t-SNE. Journal of machine learning research, 9(Nov), pp.2579-2605.

[16] Salama, G.I., Abdelhalim, M. and Zeid, M.A.E., 2012. Breast cancer diagnosis on three different datasets using multi-classifiers. Breast Cancer (WDBC), 32(569), p.2.

[17] Weinberger, K.Q. and Saul, L.K., 2009. Distance metric learning for large margin nearest neighbor classification. Journal of Machine Learning Research, 10(Feb), pp.207-244.

[18] Cover, T.M. and Hart, P.E., 1967. Nearest neighbor pattern classification. IEEE transactions on information theory, 13(1), pp.21-27.

[19] Press, S.J. and Wilson, S., 1978. Choosing between logistic regression and discriminant analysis. Journal of the American Statistical Association, 73(364), pp.699-705.

[20] Nerlove, M. and Press, S.J., 1973. Univariate and multivariate log-linear and logistic models (Vol. 1306). Santa Monica: Rand.

[21] Press, S.J., 1972. Multivariate stable distributions. Journal of Multivariate Analysis, 2(4), pp.444-462.

[22] Dimitoglou, G., Adams, J.A. and Jim, C.M., 2012. Comparison of the C4. 5 and a Naïve Bayes classifier for the prediction of lung cancer survivability. arXiv preprint arXiv:1206.1121.

[23] Friedman, N., Geiger, D. and Goldszmidt, M., 1997. Bayesian network classifiers. Machine learning, 29(2-3), pp.131-163.

[24] Vapnik, V., 2013. *The nature of statistical learning theory*. Springer science & business media.

[25] Kharroubi, J., Petrovska-Delacrétaz, D. and Chollet, G., 2001. Combining GMM's with suport vector machines for text-independent speaker verification. In Seventh European Conference on Speech Communication and Technology.

[26] Krogh, A. and Vedelsby, J., 1995. Neural network ensembles, cross validation, and active learning. In Advances in neural information processing systems (pp. 231-238).

[27] Cudeck, R. and Browne, M.W., 1983. Cross-validation of covariance structures. Multivariate Behavioral Research, 18(2), pp.147-167.

[28] Karabulut, E.M., Özel, S.A. and Ibrikci, T., 2012. A comparative study on the effect of feature selection on classification accuracy. Procedia Technology, 1, pp.323-327.

[29] Bekkerman, R., El-Yaniv, R., Tishby, N. and Winter, Y., 2003. Distributional word clusters vs. words for text categorization. Journal of Machine Learning Research, 3(Mar), pp.1183-1208.

[30] Guyon, I. and Elisseeff, A., 2003. An introduction to variable and feature selection. Journal of machine learning research, 3(Mar), pp.1157-1182.