# Machine Learning to Predict Breast Cancer Survival and Identify Gene Expression Clusters

Zhuoni Jie
Dept. of Computer Science and Technology
University of Cambridge
zj245@cam.ac.uk

*Word Count: 2497*

## I. INTRODUCTION

Breast cancer is a genetically and clinically heterogeneous cancer, and is the most common cancer among women worldwide [1]. There have been several widely accepted methods to classify breast cancers into subgroups, and analyze the presence or absence of immunohistochemical (IHC) markers like ER, PR and HER2 [2]. Clinical guidelines such as Nottingham Prognostic Index (NPI) based on clinicopathological features and prognostic panels based on gene markers have proven instrumental in guiding clinical decisions [3][4]. An accurate breast cancer prognosis is important. With the development of methods to facilitate the analysis of big data, traditional prognostic tools may be refined implementing machine learning methods. Decision tree, support vector machine, and artificial neural network have been used in cancer research for the past two decades [5][6][7]. Ensemble methods have also been successful in cancer classification [8]. For feature preprocessing, dimensionality reduction methods, such as principal components analysis on known gene signatures have been found to improve the prediction of breast cancer survival [9].

Since cancer progression is likely to relate with a small fraction of genes, research has extensively studied constructing models to identify genes associated with cancer progression (e.g., [10], [11]), and one of the most commonly used approaches is correlation analysis [12]. However, correlation analysis can only find genes with a linear dependency with survival time, and can only analyze one gene at a time.

In this project, I systematically examine performance across four supervised and two unsupervised machine learning models for breast cancer diagnosis. This project's goal is to predict a discrete breast cancer survival status combining clinicopathological features and genomic features with one feature selection technique, and discover clusters in gene expression across different tumors.

## II. DATASET AND FEATURES

### A. Data

This project uses the Molecular Taxonomy of Breast Cancer International Consortium (METABRIC) dataset accessed from cBioPortal [13] [14], which has detailed clinical information of patients for a median of 10-year follow-up period along with genomic data (copy number aberrations and gene expression) from fresh frozen tumor tissue. METABRIC contains a merged sample of 2509 patients, including 2506 breast cancer cases and 3 breast sarcoma cases, and contains the expression levels of 25160 genes and copy number data of 30566 genes from tumor samples.

In the current analysis, I focus on patients with both clinical and genomic data, and excluded samples with either incomplete clinical or genomic data. Genomic data in the form of gene expression levels were examined. After excluding patients without complete genomic measurement results and/or without gene expression measurement data, 1519 patients were included in the following analysis.

*1) Clinical Information:* Seventeen clinical parameters are contained for each patient, including lymph nodes examined positive, Nottingham prognostic index (NPI), cellularity of tumor content, cohort, testing for estrogen receptor (ER), human epidermal growth factor receptor (HER2) status, inferred menopausal status, integrative cluster, age at diagnosis, Pam50 + Claudin-low subtype, three-gene classifier subtype, primary tumor laterality, tumor other histologic subtype, type of breast surgery, chemotherapy, radio therapy, and hormone therapy status were included in the analysis.

*2) Genomic Data:* 18483 mutated genes are included, with PIK3CA, TP53, MUC16, AHNAK2, and SYNE1 among the most frequent ones. However, somatic mutation profiles of breast cancer patients exhibit a very sparse data form, and even clinically identical patients may share no more than a single mutation. From a machine learning perspective, having a limited number of patients (a far less number of patients than the number of effected genes in a cohort) introduces a dimensionality challenge. Therefore, I do feature extraction and feature selection as described in the following sections.

*3) Survival Status:* There are three indicators of survival: time from breast cancer diagnosis to last follow-up (overall survival in months), overal survival status, and status of the patient (alive or dead) at last follow-up time (patient's vital status). Survival data is highly skewed and right-censored, since patients may be alive at the end of the study or lost to follow-up.

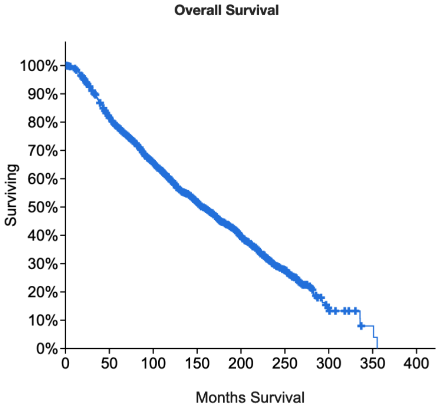


Fig. 1. Overal survival rate of the METABRIC samples versus time

### B. Feature Selection

Principle Component Analysis (PCA) is a classical technique that rotationally transforms features of a dataset into a lower dimensional set of uncorrelated features called principal components (PCs). PCA is commonly used to reduce the dimension of a dataset, since too many dimensions is wasteful for learning algorithms and might lead to overfitting problems. As PCA tries to find orthogonal projects of the dataset, it makes the strong assumption that some of the variables in our dataset is linearly correlated. We combined both clinicopathological features and genomic features to generate potential predictors for our models, and implemented PCA to reduce feature dimension. By setting proper PCs, I obtained 92.65% of explained variance, and reduced the feature dimensionality from original 18501 to 1100.

Other possible feature selection methods include the Recursive Feature Elimination, which starts with the initial set of features and recursively remove one feature that is the least important until the desired number of features is finally reached, and the Correlation Heat Map.

## III. METHODS

I built four supervised machine learning models that predict the patients status (dead or alive) based on the selected features, and also used two unsupervised clustering methods to detect patterns in gene expressions. For survival prediction, patient status was used as the target variable and all other features as the input features.

When building the models, I randomly set starting values for the model parameters, and then did hypertuning to find the optimal values for the model to improve accuracy. I used *GridSearchCV* in *Scikit-Learn* library [15] to do hyperparameter tuning, which works by training the model multiple times on a range of parameters that are specified.

### A. k-Nearest Neighbors

The k-Nearest Neighbors algorithm (kNN) [16] is a supervised non-parametric machine learning model used for classification and regression. kNN works by taking a data point and looking at the $k$ closest labeled data points, and then this data point is assigned the label of the majority of the $k$ closest points. Distances such as Euclidean distance and Manhattan distance are used in kNN. The algorithm uses 'feature similarity' to predict values of any new data points, which means the new point is assigned a value based on how closely it resembles the points in the training set. In kNN classification the output is a class membership, while in kNN regression is the property value for the object. The best choice of $k$ depends upon the data. Generally, larger values of $k$ reduces effect of the noise on the classification, but make boundaries between classes less distinct.

### B. Linear Support Vector Machine

Support Vector Machines (SVMs) [17] are supervised learning models that utilize hyperplanes and support vectors to analyze data for classification and regression. Given labeled training data, the SVM algorithm outputs an optimal hyperplane which optimally categorizes new examples.

SVM is known for its kernel trick to handle nonlinear input spaces. The main function of the kernel is to transform the given dataset input data into the required form. There are various types of functions such as linear, polynomial, and radial basis function (RBF). Polynomial and RBF are useful for non-linear hyperplane. Polynomial and RBF kernels compute the separation line in the higher dimension. In some of the applications, it is suggested to use a more complex kernel to separate the classes that are curved or nonlinear. Here I use a linear SVM model. SVM has other two parameters: a regularization parameter and a Gamma. In *Scikit-learn*, $C$ is the penalty parameter, which represents misclassification or error term. The misclassification or error term tells the SVM optimization how much error is bearable. A smaller value of $C$ creates a small-margin hyperplane and a larger value creates a larger-margin hyperplane. A lower value of $Gamma$ will loosely fit the training dataset, whereas a higher value will exactly fit the training dataset, which causes over-fitting.

SVM's remarkably robust performance with respect to sparse and noisy data making is making it systematically used in a variety of applications such as face detection [18], intrusion detection [19], classification of genes [20], and handwriting recognition [21].

### C. Random Forest

Random Forests (RF) or random decision forests [22] are an ensemble learning method for classification, regression and other tasks that operates by constructing a multitude of decision trees at training time and outputting the class that is the mode of the classes (classification) or mean prediction (regression) of the individual trees. In each decision tree, the input enters at the root of the tree and traverses down the tree according to the split decision at each node. Along the way, data gets bucketed into smaller and smaller sets.

Random decision forests correct for decision trees' habit of overfitting to their training set. In a random forest, the hyperparameters include the number of decision trees in the forest and the number of features considered by each tree when splitting a node.

### D. Gradient Boosting

Gradient Boosting (GB) [23] is a machine learning technique for regression and classification problems, which trains many models in a gradual, additive and sequential manner, and produces a prediction model in the form of an ensemble of weak prediction models, typically decision trees. The major difference between Gradient Boosting Algorithm and other boosting algorithms (e.g., AdaBoost [24]) is how the algorithms identify the shortcomings of weak learners (e.g., decision trees). While the AdaBoost model identifies the shortcomings by using high weight data points, gradient boosting performs the same by using gradients in the loss function. By allowing optimization of an arbitrary differentiable loss function, Gradient Boosting generalizes the boosting algorithms.

GB has there hyperparameters. Parameter $J$ is the number of terminal nodes in trees. Parameter $M$ is the number of gradient boosting iterations. Increasing $M$ reduces the error on training set, but setting it too high may lead to overfitting. And parameter $\nu$ is the learning rate, which modifies the update rule in regularization by shrinkage. Empirically, using small learning rates has been found to yield dramatic improvements in model's generalization ability over GB models without shrinking.

### E. k-Means Clustering

K-Means Clustering [25] is a method of vector quantization which aims to partition $n$ observations into $k$ clusters in which each observation belongs to the cluster with the nearest mean. This method produces exactly $k$ different clusters of greatest possible distinction. The best number of clusters k leading to the greatest separation must be computed from the data, and the problem is computationally difficult (NP-hard). However, efficient heuristic algorithms converge quickly to a local optimum. The objective of K-Means clustering is to minimize total intra-cluster variance or the squared error function, and the parameter is $k$, the number of clusters.

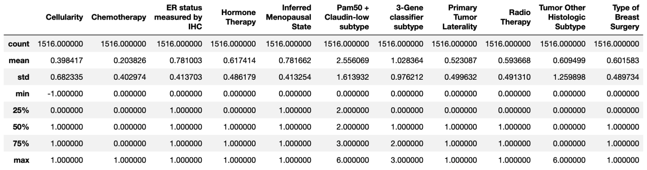
### F. Hierarchical Clustering

Hierarchical Clustering [26] is a method of cluster analysis which seeks to build a hierarchy of clusters. There are two types of strategies for hierarchical clustering, divisive and agglomerative. Agglomerative strategy is a bottom-up approach. Each observation starts in its own cluster, and pairs of clusters are merged as one moves up the hierarchy. Divisive strategy is a top-down approach. All observations start in one cluster, and splits are performed recursively as one moves down the hierarchy. Here I implement the agglomerative strategy.

## IV. Evaluation and Results

I trained a series of machine learning methods to predict survival status with 5-fold cross-validation of the training set upon 5 random training/validation splits using kNN, SVM, Random Forest, and Gradient Boosting. For each split, 80% of the analytic cohort were randomly selected as our training dataset. Model performance was examined in the remaining 20% validation dataset, by plotting receiver-operating characteristic (ROC) curve and estimating the accuracy and speed.

The analysis cohort consisted of 1212 breast cancer patients with average age of 61 years and average NPI of 4. ER status, and HER2 status were positive in 78.10% and 17.02% of patients, respectively. Part of the statistical clinical data summary is shown in Fig. 2.



| | Cellularity | Chemotherapy | ER status measured by IHC | Hormone Therapy | Inferred Menopausal State | Pam50 + Claudin-low subtype | 3-Gene classifier subtype | Primary Tumor Laterality | Radio Therapy | Tumor Other Histologic Subtype | Type of Breast Surgery |
|---|---|---|---|---|---|---|---|---|---|---|---|
| **count** | 1516.000000 | 1516.000000 | 1516.000000 | 1516.000000 | 1516.000000 | 1516.000000 | 1516.000000 | 1516.000000 | 1516.000000 | 1516.000000 | 1516.000000 |
| **mean** | 0.398417 | 0.203826 | 0.781003 | 0.617414 | 0.781662 | 2.556069 | 1.028364 | 0.523087 | 0.593668 | 0.609499 | 0.601583 |
| **std** | 0.682335 | 0.402974 | 0.413703 | 0.486179 | 0.413254 | 1.613932 | 0.976212 | 0.499632 | 0.491310 | 1.259898 | 0.489734 |
| **min** | -1.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 |
| **25%** | 0.000000 | 0.000000 | 1.000000 | 0.000000 | 1.000000 | 2.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 |
| **50%** | 1.000000 | 0.000000 | 1.000000 | 1.000000 | 1.000000 | 2.000000 | 1.000000 | 1.000000 | 1.000000 | 0.000000 | 1.000000 |
| **75%** | 1.000000 | 0.000000 | 1.000000 | 1.000000 | 1.000000 | 3.000000 | 2.000000 | 1.000000 | 1.000000 | 0.000000 | 1.000000 |
| **max** | 1.000000 | 1.000000 | 1.000000 | 1.000000 | 1.000000 | 6.000000 | 3.000000 | 1.000000 | 1.000000 | 6.000000 | 1.000000 |

Fig. 2. Part of the statistical summary of input clinical data

| Algorithm | Accuracy % | F1 % | Speed |
|---|---|---|---|
| k-Nearest Neighbor | 59.63 | 57.02 | 0.58 |
| Linear Support Vector Machines | 57.18 | 56.14 | 1.98 |
| Random Forest | 57.06 | 43.88 | 15.96 |
| Gradient Boosting | 61.20 | 61.00 | 7.92 |

Fig. 3. Model accuracy, F1, and execution speed

The model performance of supervised classification is as shown above in Fig. 3. From the evaluation accuracy we can see none of the models performed better than 61.20% for 5-fold CV, with accuracy values ranging from 57.18% to 61.20%, and F1 score from 43.88% to 61.00%. The models were badly overfitting the data and were not representing the relationships between the features accurately, probably due to inappropriate processing with genetic information. In particular, the Random Forest resulted in the worst accuracy and F1 score, which is an ensemble learning method using multiple weak prediction models (decision trees) to form a single model in a stage-wise fashion. It is the most overfitted model. Among the four models, Gradient Boosting got the much better results for both accuracy and F1 scores.

As for model run time, each run time is averaged by five duplicate executions. The kNN model is the fastest, 27 times faster than the worst Random Forest model. In addition, it is interesting to note that kNN is more accurate and faster than linear SVM and RF, even though kNN is a more simple model. Gradient Boosting model probably offers a trade-off between accuracy and speed here in our experiment.

To reduce overfitting, especially for overfitting of the Random Forest model, there are three possible ways: feature selection, tuning the number of trees in each forest, and tuning the max number of features in each tree. More Gridsearch can be implemented to search for better hyperparameters for RF model. During the model training, I also observed improvement of performance after tuning the model parameters.

The ROC curve is also a useful tool for evaluating models [27]. ROC curves feature true positive rate on the Y axis, and false positive rate on the X axis. This means that the top left corner of the plot is the ideal point. The curves of

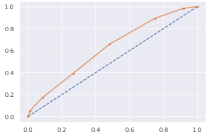


Fig. 4. ROC for kNN

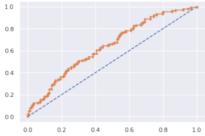


Fig. 5. ROC for SVM

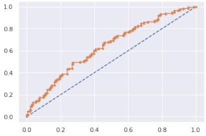


Fig. 6. ROC for GB

different models can be compared directly in general, and further calculating the area under the curve (AUC) can be used as a summary of the model skill. The shape of the curve contains a lot of information, including what we might care about most for a problem, the expected false positive rate, and the false negative rate. The ROC curve for three models are presented here in Fig. 4, Fig. 5, and Fig. 6. We can see three models have similar performance regarding the ROC curve's position. Since it is ideal to maximize the true positive rate while minimizing the false positive rate, we also take a look at the steepness of each curve, and again they have similar patterns.

Here I implemented k-Means and Hierarchical Clustering to do gene expression clustering analysis. For gene clustering, pairwise similarity metrics among genes are calculated on the basis of expression ratio measurements across all tumours. I plotted the average distances of observations from the cluster centroid to use the Elbow Method to identify number of clusters to choose.

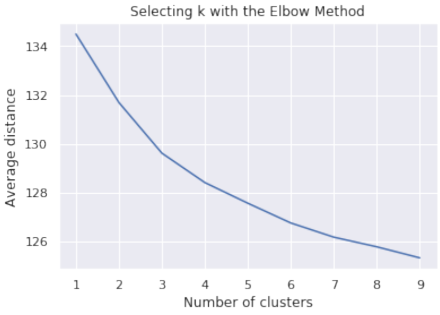


Fig. 7. Selecting the number of clusters with the Elbow Method

Looking at the bend in the Elbow plot (Fig. 7) that shows where the average distance value might be leveling off such that adding more clusters does not decrease the average distance as much, I choose 4 as the number of clusters. From Fig. 8 we can see the genetic expression results are clustered into four groups, and we can further do tumour clustering, and identify more relations between genetic expression and tumour formulation. Fig. 9 & 10 present the clustering result using hierarchical clustering algorithm, resulting in four clusters.

## V. CONCLUSION AND FUTURE WORK

In this analysis of machine learning methods in breast cancer survival prediction, we found that no model got significantly good prediction results, probably because of the construction way of the features, making models overfitted. Among these models, Gradient Boosting model outperformed others in accuracy, and kNN outperformed in speed. The Random Forest model is the worst performed and most overfitted. Using unsupervised learning, we got some intuitive gene expression subgroups, which are worth analyzing when taking into more biological and pathological knowledge contexts.

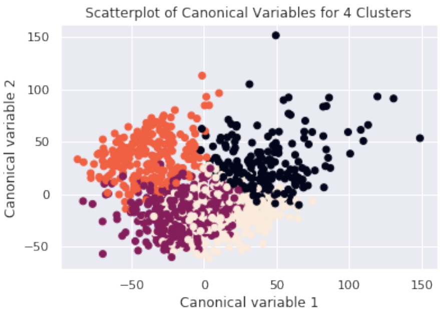


Fig. 8. Clustering result of the k-means clustering algorithm, resulting in four clusters

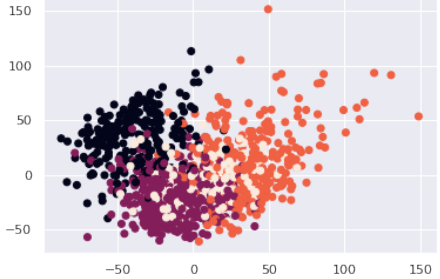


Fig. 9. Clustering result of the hierarchical clustering algorithm, resulting in three clusters

Future work can be extended by refining the feature selection methods, especially discussing better ways to fusion the clinical and gene expression data. In addition, instead of solely predicting the survival status, we can predict survival time of the patient. We can also do more detailed analysis regarding the influencial variables and their underlying relationships, for example, calculating each variable's importance by taking the difference between whole model accuracy and model accuracy after permuting each predictor variable. We can also do subgroup analysis (e.g., high-risk and low-risk groups).

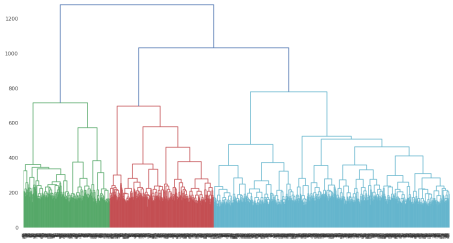


Fig. 10. An indexed hierarchy of the hierarchical cluster, and the merging levels correspond to the measure of dissimilarity between the cluster groups

# References

[1] Ferlay, Jacques, et al. ”Cancer incidence and mortality worldwide: sources, methods and major patterns in GLOBOCAN 2012.” International journal of cancer 136.5 (2015): E359-E386.

[2] Elston, Christopher W., and Ian O. Ellis. ”Pathological prognostic factors in breast cancer. I. The value of histological grade in breast cancer: experience from a large study with longterm followup.” Histopathology 19.5 (1991): 403-410.

[3] Galea, Marcus H., et al. ”The Nottingham Prognostic Index in primary breast cancer.” Breast cancer research and treatment 22.3 (1992): 207-219.

[4] Sparano, Joseph A., et al. ”Prospective validation of a 21-gene expression assay in breast cancer.” New England Journal of Medicine 373.21 (2015): 2005-2014.

[5] Maclin, Philip S., et al. ”Using neural networks to diagnose cancer.” Journal of medical systems 15.1 (1991): 11-19.

[6] Simes, R. John. ”Treatment selection for cancer patients: application of statistical decision theory to the treatment of advanced ovarian cancer.” Journal of chronic diseases 38.2 (1985): 171-186.

[7] Kourou, Konstantina, et al. ”Machine learning applications in cancer prognosis and prediction.” Computational and structural biotechnology journal 13 (2015): 8-17.

[8] Tan, Aik Choon, and David Gilbert. ”Ensemble machine learning on gene expression data for cancer classification.” (2003).

[9] Zhao, Xi, et al. ”Combining gene signatures improves prediction of breast cancer survival.” PLoS One 6.3 (2011): e17845.

[10] Van’t Veer, Laura J., et al. ”Gene expression profiling predicts clinical outcome of breast cancer.” nature 415.6871 (2002): 530.

[11] Sun, Yijun, et al. ”Improved breast cancer prognosis through the combination of clinical and genetic markers.” Bioinformatics 23.1 (2006): 30-37.

[12] Slamon, Dennis J., et al. ”Human breast cancer: correlation of relapse and survival with amplification of the HER-2/neu oncogene.” science 235.4785 (1987): 177-182.

[13] Curtis, Christina, et al. ”The genomic and transcriptomic architecture of 2,000 breast tumours reveals novel subgroups.” Nature 486.7403 (2012): 346.

[14] Pereira, Bernard, et al. ”The somatic mutation profiles of 2,433 breast cancers refine their genomic and transcriptomic landscapes.” Nature communications 7 (2016): 11479.

[15] Pedregosa, Fabian, et al. ”Scikit-learn: Machine learning in Python.” Journal of machine learning research 12.Oct (2011): 2825-2830.

[16] Keller, James M., Michael R. Gray, and James A. Givens. ”A fuzzy k-nearest neighbor algorithm.” IEEE transactions on systems, man, and cybernetics 4 (1985): 580-585.

[17] Suykens, Johan AK, and Joos Vandewalle. ”Least squares support vector machine classifiers.” Neural processing letters 9.3 (1999): 293-300.

[18] Osuna, Edgar, Robert Freund, and Federico Girosit. ”Training support vector machines: an application to face detection.” Computer vision and pattern recognition, 1997. Proceedings., 1997 IEEE computer society conference on. IEEE, 1997.

[19] Mukkamala, Srinivas, Guadalupe Janoski, and Andrew Sung. ”Intrusion detection using neural networks and support vector machines.” Neural Networks, 2002. IJCNN’02. Proceedings of the 2002 International Joint Conference on. Vol. 2. IEEE, 2002.

[20] Guyon, Isabelle, et al. ”Gene selection for cancer classification using support vector machines.” Machine learning 46.1-3 (2002): 389-422.

[21] Dong, Jian-xiong, Adam Krzyak, and Ching Y. Suen. ”An improved handwritten Chinese character recognition system using support vector machine.” Pattern Recognition Letters 26.12 (2005): 1849-1856.

[22] Liaw, Andy, and Matthew Wiener. ”Classification and regression by randomForest.” R news 2.3 (2002): 18-22.

[23] Friedman, Jerome H. ”Greedy function approximation: a gradient boosting machine.” Annals of statistics (2001): 1189-1232.

[24] Viola, Paul, and Michael Jones. ”Rapid object detection using a boosted cascade of simple features.” Computer Vision and Pattern Recognition, 2001. CVPR 2001. Proceedings of the 2001 IEEE Computer Society Conference on. Vol. 1. IEEE, 2001.

[25] Hartigan, John A., and Manchek A. Wong. ”Algorithm AS 136: A k-means clustering algorithm.” Journal of the Royal Statistical Society. Series C (Applied Statistics) 28.1 (1979): 100-108.

[26] Corpet, Florence. ”Multiple sequence alignment with hierarchical clustering.” Nucleic acids research 16.22 (1988): 10881-10890.

[27] Bradley, Andrew P. ”The use of the area under the ROC curve in the evaluation of machine learning algorithms.” Pattern recognition 30.7 (1997): 1145-1159.