XGBoost models for cancer diagnoses in Wisconsin breast cancer data set

Christina Kallendorf

Udacity Machine Learning capstone project

Introduction

Motivation/Domain background

Machine Learning opens up a vast spectrum of medical applications. Some examples are medical image processing for diagnosis, classification of medical findings as well as disease prognosis and risk assessment [1]. Sidey-Gibbons claim that machine learning can be highly beneficial to the areas of diagnosis and outcome prediction in medical applications [2]. In particular, Machine Learning can be supportive to diagnose and classify specific sub-types of cancer from biopsy probes. In this context, Machine Learning can lead to an "effective and accurate decision making", as the authors of [3] point out. This may help to diagnose malignant tumors early and better define a suitable treatment. As pointed out by [13], this may help to "increase survival rates from 56% to more than 86%". The project proposed here takes up one classification example of the area of cancer research and focuses on the identification of malignant breast tumors based on cytology features from biopsies. Note that feature extraction is not subject of this project and features are provided with the data. Determination of feature values by processing images of the biopsy samples is not subject of this project.

Problem statement

Based on a medical patient's biopsy, it needs to be determined whether the investigated tissue contains malignant, i. e. cancer cells. In order to react to the growth of cancer cells quickly and appropriately, it is necessary to define such cells at an early state. Usually, biopsy results are classified by laboratory staff, based on their investigation of cell properties under a microscope. Therefore, a diagnosis is subject to individual judgement and human error [2].

Solution statement

A Machine Learning model can be able to identify breast cancer reliably for a given sample. It is a supervised binary classification that is itself based on a defined set of cytology features from biopsies, such as shape and size. As a result, cytology features obtained from biopsies enable a Machine Learning model to provide an effective, accurate and fast diagnosis. This has advantages over a subjective diagnosis generated by laboratory staff, see also [2], and may help physicians in decision-making, refer to [13].

Dataset

This project employs a data set of breast tissue biopsies that was collected by University of Wisconsin Hospitals, Madison, by Dr. William H. Wolberg [4, 5, 6, 7] from tumor patients during the years of 1989 till 1991. This data set comprises eleven cytology characteristics of breast fine-needle aspirates. Specifically, the cytology features presented in this data where computed from images of biopsies, using

methods developed in [4-6]. It comprises 699 data samples with nine cytology features, Ids and the classification of benign tissue, by value 0, and malignant, i. e. cancer, tissue by value 1. The original data is published at https://archive.ics.uci.edu/ml/datasets/Breast+Cancer+Wisconsin+%28Original%29 , please referto [8]. The data set employed here was obtained from https://www.kaggle.com/johnyquest/wisconsin-breast-cancer-cytology-features as a formatted .csv file .

XGBoost Algorithm for Classifying Cytology features

The data set used here has previously been employed for machine learning classification studies, in particular using SVM as well as Neural Networks. For further details, refer, for instance, to references [8-13]. In recent years, extreme gradient boosting (XGBoost) algorithms have become popular, giving "state-of-the-art results on many standard classification benchmarks" [14], while succeeding most popular algorithms in speed and scaling well to high data volumes, as the authors of [14] point out. Motivated by the fact, that XGBoost models have been used by many price-winners in machine learning competitions, such as Kaggle or KDDCup in 2015 [14], I intend to create an XGBoost model for classifying the Wisconsin Breast Cancer Dataset introduced above. This project is guided by one central question: How does an XGBoost model compare to models presented in literature previously? In particular, does it outperform an SVG model, which achieves sensitivity of 97%, specificity of 0.94% and an accuracy = 0.96?

What again is XGBoost? Let us shortly recapitulate:

XGBoost is a learning algorithm that applies boosting to both linear and tree learners. Specifically, boosting implies that an optimal predictor is constructed incrementally from a sum of weaker predictors, known as weak learners, using a greedy algorithm. Each new predictor is obtained by adding the previously predictors to the current predictor and optimizing the weights of this linear combination, which are basically obtained from partial derivatives, i. e. the gradient of the loss function [16]. Extreme gradient boosters can be seen as a new generation focusing on high-performance computing, by providing scalability and speed increase and supporting distributed computing, also refer to [14].

XGBoost is available as open source algorithm, for instance, in Python via the XGBoost package, as well as an built-in managed algorithm on commercial machine learning platforms such as SageMaker.

Data preparation and exploration

Properties of input data

In preparation for training and testing, the Wisconsin Breast Cancer data set (WBDC) is imported as `DataFrame`.Duplicates as well as rows with null values are removed from the data records, resulting in a set of 675 samples, out of which 236, i. e. 35%, are labeled as malignant and 439, i. e. 65% are labeled as benign. The column with IDs of original aspirates is removed, as they do not contain relevant information to classifying any data.

The samples comprise the following cytology characteristics of the fine needle aspirates with values ranging from 1 to 10:

- Clump thickness,

- Uniformity of cell size,
- Uniformity of cell shape,
- Marginal adhesion,
- Single epithelial cell size,
- Bare nuclei,
- Bland chromatin,
- Normal nucleoli,
- Mitoses.

Data exploration

In order to give a first overview of the distribution of the respective data samples, the method `plot_feature_distribution` systematically creates scatter plots of the given data with respect to two indicated features. It is applied to all features pairwise. Some of the results are presented in Fig. 2; purple spots indicate samples of benign probes, yellow marks malignant cases. These plots indicate that in benign cases, most of the feature values concentrate in the left lower quadrant, while malignant data points scatter throughout the outer area. However, many outliers illustrate that neither of the features is a clear marker for malignant or benign cases.

Furthermore, the feature's correlation matrix (i. e. two components' variances normalized by the product of standard variances) is determined and illustrated by Fig. 2 below.

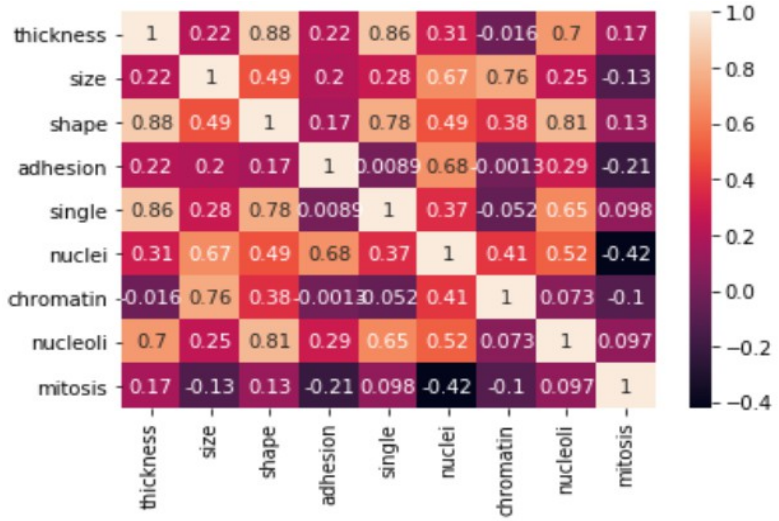

Fig. 1: Heatmap of correlation values of all features

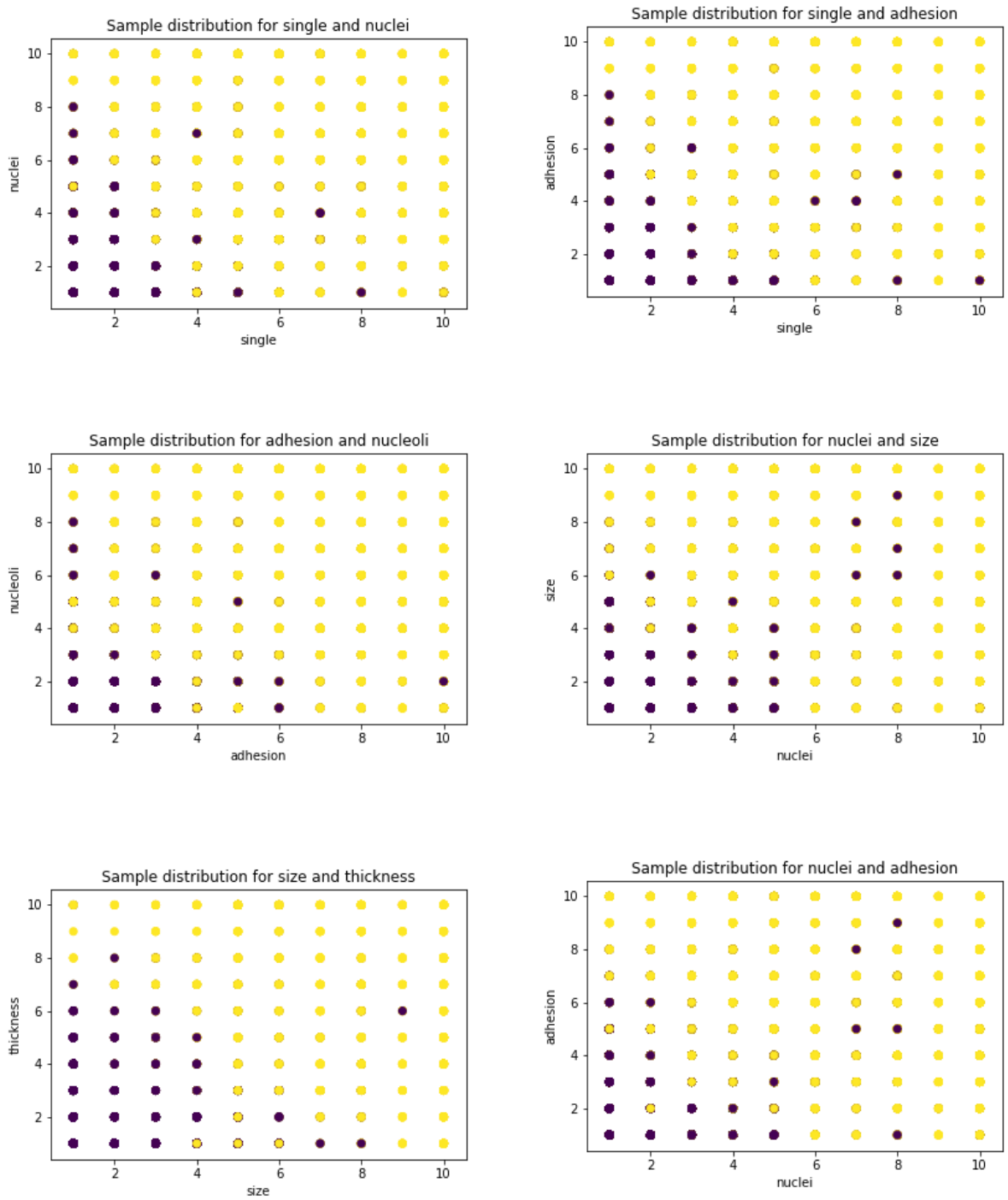

Fig. 2: Scatter plot of some of the feature distributions

The matrix indicates that neither of these features is sufficiently described by any other and therefore, could be simply omitted. However, few features prove to be at least a highly correlated with a correlation greater than 80%, which are, in detail:

- shape and thickness,
- shape and nucleoli,
- single and thickness.

Data preparation

In preparation for training, the data is randomly shuffled, as the provided data set is ordered by creation time of the fine needle aspirates. It is divided into 50% training , 25% test and 25% validation data. Labels, i. e. the classes of the sample, are separated. Next, the function `data_as_txt` is used to prepare data as csv files with labels as first column, followed by all features of interest, and upload these files to S3 (default) bucket.

Evaluation metrics

Definition of evaluation metrics

Using machine learning for diagnostic classification, major objectives should be minimization of false positives and false negatives, while optimizing accuracy of the model. As suggested by [2], results presented here are measured and optimized for the following metrics,

- Sensitivity=true positives/(true positives + false negatives), i. e. the clinical term for recall.
- Specificity=true negatives/(true negatives + false positives),
- Accuracy= (true positives + true negatives) / total predictions.

The method `eval_metrics` computes these metrics, based on the test labels and the predicted labels of the test data. Furthermore, the results as well as the hyperparameters of the estimators are documented in a *metrics.csv* file, in order to be accessible later on.

Model setup & Implementation

XGBRegressor

`XGBRegressor`, or `XGBClassifier`, from `xgboost` library are used for a first evaluation of suitable booster, which is not supported by hyperparameter optimization. Specifically, this includes

- type of booster,
- tree-method, if a tree-booster is applied,
- as well as the evaluation metric

used for building the model. Due to the data set's small size, computation can be completed fast. Note that the objective should be set to binary hinge, which is suitable for 1-0, or in other words, true-false classifications. With respect to recall and specificity, results in the below table illustrate that the tree-boosters(gbtree and dart), applied with an exact greedy algorithm succeeds the approximate as well as the faster histogram optimized approximate greedy algorithms, as well as the linear booster,

independent of the applied evaluation metrics. With respect to accuracy, a linear booster is performs slightly better.

Booster	**gbtree**		**dart**		**gblinear**
tree_method	Auto/exact	Approx/hist	auto/exact	approx/hist	N/a
eval_metric	mae, rmse, error, mlogloss	mae, rmse, error, mlogloss	mae, rmse, error, mlogloss	mae, rmse, error, mlogloss'	mae, rmse, error, mlogloss
Recall	0.97222222	0.94444444	0.97222222	0.94444444	0.95833333
Specificity	0.9787234	0.95918367	0.978723	0.95918367	0.96938776
Accuracy	0.95857988	0.95857988	0.95857988	0.95857988	0.9704142

Sagemaker XGBooster

`Sagemaker XGBoost` is used as a framework with a customized training script as entry point. The original estimator is instantiated with booster `gbtree`, using the `exact` tree method and `rmse` as evaluation metrics. The customized training script, available at `source/train.py`, provided as entry point is a simple preparation for additional tasks which could arise in future, for example, regarding data processing.
One disadvantage of using a gradient boosting algorithm is the variety of hyperparameters which impact the quality of the model. For a systematic checkup of combinations of different hyperparameters, sagemaker hyperparameter tuning is employed for 20 combinations on 4 machines in parallel. The following parameters are selected by Bayesian optimization:
- numerical rounds, out of an integer range from 40 to 200,
- maximal depth, out of an integer range from six to twelve,
- eta and gamma, out of continuous parameter ranges in an interval from 0 to 0.4, and 0 to 1, respectively.

The model with resulting optimal hyperparameters from hyperparameter training job is finally deployed and used for evaluating our test data. In detail, these parameters are `'eta': '0.2572110893586683', 'gamma': '0.5662623155210383', 'max_depth': '10'` and `'num_round': '135'`.

Implementation challenges

Implementing XGBoost with a customized training script as entry point was particularly challenging. I have added a function `prediction_labels` which reads in data per line from a defined test data file, calls the predict function and appends the results to one array. Specifically, it was not clear how to define the line split for input data for prediction in txt/csv format and matrix format was rejected for security reasons.

Results & Conclusion

In comparison to results given by [2] for GLM, SVM and a neural network approach (ANN), with

Accuracy: 0.9763313609467456
Specificity: 0.9894736842105263
and Recall: 0.9861111111111112

the presented XGBoost model exceeds all of these algorithms with respect to each of the evaluation metrics, please view the result table presented below.

In this context, I would like to highlight , that

1. yes, this answers the original question positively – XG boost outperforms the SVM model presented by [2] and
2. Hyperparameter tuning has significantly improved the model as compared to the initial investigation.

	GLM (ref. [2])	SVM (ref. [2])	ANN (ref. [2])	XGBoost (gbtree)	XGBoost (gbtree) with PCA reduction to 7 components
Recall	0.99	0.97	0.99	0.99	0.96
Specificity	0.87	0.94	0.86	0.99	0.96
Accuracy	0.95	0.96	0.94	0.98	0.97

Finally, two critical aspects need to be considered:

First of all, train and validation data set comprise a rather small number of samples, considering in particular the number of variants which are possible for all 9 features and the high number of outliers. This may imply that results are 'incidentally' good just for a small range of test data used here, but would not hold for a larger set of data. However, at least when comparing to previous investigations of the same data set, one may argue that these results had the same limitation on data.

Second, it is well-known that tree-learning methods are sensitive to over-fitting. A higher volume of data records would be required in order to investigate on this.

Outlook:

Finally, I would like to discuss on correlation of features. The initial data exploration indicated that there is a correlation of more than 80% between the features shape with both thickness and nucleoli, as well as single and thickness. Does this imply that data records can be reliably classified using less features?

This is interesting to investigate, as a high correlation of features can have a negative impact on performance of the algorithms employed. Against this background, I utilize a PCA Analysis for dimensionality reduction of component space. Considering the number of correlated features, it can be assumed the material possibly reduced by two, i. e. PCA is applied with seven components. After that, the XGBoost model developed above is employed for determining the evaluation metrics. This still leads to surprisingly high results, with an accuracy of 97%, specificity of 97% and recall of 96%. An additional improvement may be expected from further hyperparameter tuning.

References:

[1] de Bruijne, Marleen (2016). Machine learning approaches in medical image analysis: From detection to diagnosis. *Medical Image Analysis*, 33. doi: 10.1016/j.media.2016.06.032

[2] Sidey-Gibbons, J., Sidey-Gibbons, C. (2019). Machine learning in medicine: a practical introduction. *BMC Med Res Methodol*, 19(64). doi: 10.1186/s12874-019-0681-4

[3] Konstantina Kourou, Themis P. Exarchos, Konstantinos P. Exarchos, Michalis V. Karamouzis, Dimitrios I. Fotiadis (2015). Machine learning applications in cancer prognosis and prediction. *Computational and Structural Biotechnology Journal*, Volume 13, 2015, pp. 8-17. doi: 10.1016/j.csbj.2014.11.005.

[4] William H. Wolberg and O.L. Mangasarian (1990). Multisurface method of pattern separation for medical diagnosis applied to breast cytology. *Proceedings of the National Academy of Sciences, U.S.A*., 87, 9193-9196.

[5] O. L. Mangasarian and W. H. Wolberg (1990). Cancer diagnosis via linear programming. *SIAM News*, 23 (5), 1-18.

[6] O. L. Mangasarian, R. Setiono, and W.H. Wolberg: "Pattern recognition via linear programming: Theory and application to medical diagnosis", in *Large-scale numerical optimization*, Thomas F. Coleman and Yuying Li, editors, SIAM Publications, Philadelphia 1990, pp 22-30.

[7] K. P. Bennett & O. L. Mangasarian (1992). Robust linear programming discrimination of two linearly inseparable sets. *Optimization Methods and Software,* 1, 23-34.

[8] Dua, D. and Graff, C. (2019). UCI Machine Learning Repository [http://archive.ics.uci.edu/ml]. Irvine, CA: University of California, School of Information and Computer Science.

[9] Gavin Brown. Diversity in Neural Network Ensembles. The University of Birmingham. 2004.

[11] Hussein A. Abbass. An evolutionary artificial neural networks approach for breast cancer diagnosis. Artificial Intelligence in Medicine, 25. 2002.

[12] Agarap, Abien Fred M. (2019). On Breast Cancer Detection: An Application of Machine Learning Algorithms on the Wisconsin Diagnostic Dataset. arXiv:1711.07831v4 [cs.LG] 7 Feb 2019

[13]Montazeri M, Montazeri M, Montazeri M, Beigzadeh A. (2016). Machine learning models in breast cancer survival prediction. *Technol Health Care,* 24(1), 31-42. doi:10.3233/THC-151071

[14] T. Chen , C. Guestrin (2016) . XGBoost: A Scalable Tree Boosting System .*arXiv*. 1603.02754v3

[16] David Forsyth (2019). Applied Machine Learning. *Springer Int. Publishing*,Ed. 1. Doi: 10.1007/978-3-030-18114-7. Isbn 978-3-030-18114-7 .