Machine Learning in Healthcare: A Comparison of Random Forests and k-NN Algorithm Performance in Breast Cancer Classification

Ryan Nazareth and Hannes Draxl

Motivation and Hypothesis

- **Motivation:** To predict whether breast cancer cells can be identified as malignant or benign based on measurements of cell features from digitized images of breast tissue masses [8]. This poses a binary classification problem. Here we choose to compare two classification algorithms for this purpose: k-Nearest Neighbours (k-NN) [5] and Random Forests (RF) [3].
- **Hypothesis:** Whilst we expect both algorithms to produce good results, we hypothesize that RF will perform better based on the ability to model complex patterns whilst not overfitting too heavily. The use of Principal Component Analysis (PCA) is also widely used in the literature to reduce dimensionality and we expect it to improve our model performance for this dataset.

Pros and Cons of k-NN and Random Forest models

Random Forest

- An ensemble of deep and independent trees built on bootstrap samples of the original data.
- Instead of providing each feature at a split node, only a maximum amount of randomly sampled features are given [3]
- This has a decorrelating effect on the individual trees in the ensemble.

Pros

- Achieves high predictive performance on numerous tasks.
- Speeds up training process through parallelization.
- The ensemble of trees leads to a real probability measure.

Cons

- If the majority of random features are noisy, RF are likely to perform poorly [5].
- Large amount of trees can be computational expensive.

K-NN

- Memorizes training data instead of learning a discriminative function.
- Uses a distance metric to find k-samples in the training data closest to a validation/test sample.
- A majority vote assigns the class label.

Pros

- Relatively simple to understand.
- The memory approach of the k-NN-Classifier allows for an immediate adaption to additional and new training data.

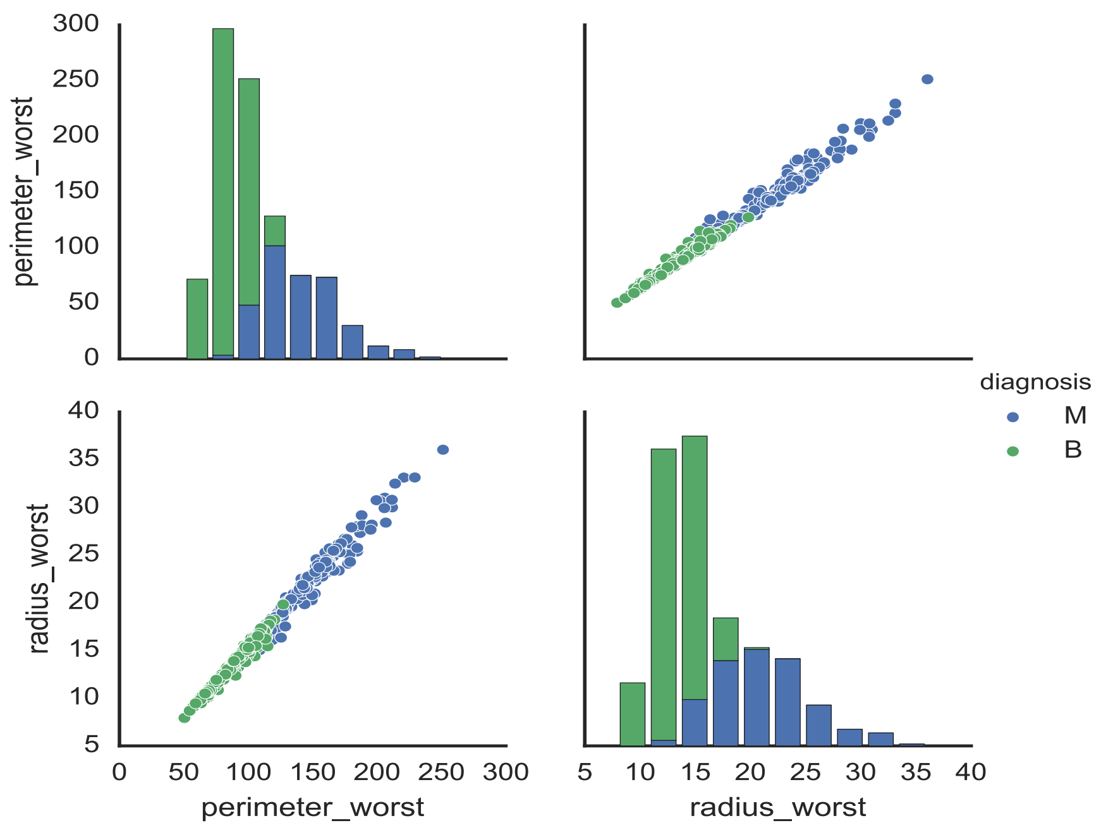
Cons

- Prone to the Curse of Dimensionality as samples become very distant to each other in higher dimensions.
- Memory intensive as training samples have to be permanently stored [1].
- Requires tuning optimal k and distance metrics.

Initial Statistical Analysis of Dataset

- Breast Cancer Wisconsin (Diagnostic) Data Set [2][9]
- 569 samples and 31 columns (ID column was dropped)
- The 30 features can be classified into 10 physical cell dimension measurements with additional correlated attributes.
- Binary target variable consisting of 'benign' and 'malignant' tumor classes. Slight class imbalance with 357 benign vs. 212 malignant cancer samples.
- The table below depicts the mean and standard deviation of the five features, which are highest correlated (point biserial) with the target variable.
- Observations: The mean and the standard deviation for the features are much larger for malignant tumor classes. The scatterplot on the right can be interpreted as high correlation between features indicating redundant information. Most features are positively skewed. A pattern in terms of the class distribution with regard to the feature value can be observed: the higher the feature value, the more likely it seems that the cancer is malignant.

Features	Mean B	Mean M	Std B	Std M
concave points_worst	0.07	0.18	0.03	0.04
perimeter_worst	87.00	141.37	13.52	29.45
concave points_mean	0.02	0.08	0.01	0.03
radius_worst	13.37	21.13	1.98	4.28
perimeter_mean	78.07	115.36	11.80	21.85



Evaluation Methodology

- The dataset was split into a training/test set (70:30 ratio). The features were scaled to mean=0 and std=1. Stratified 10-fold cross validation (CV) was used to guarantee a roughly equal sized class distribution between the individual folds which is especially important to account for the slight class imbalance in the data set (benign: 63%, malignant: 37%) .
- CV was then used with the Bayesian optimization [7] for Hyperparameter Tuning (HP). This internally maintains a Gaussian process model of the objective function, and trains the model using objective function evaluations [6]. Instead of a brute force gridsearch, bayesian optimisation uses an acquisition function to evaluate which HP combinations to run next [6].
- The performances of both k-NN and RF with and without the use of PCA dimensionality reduction were explored.
- The Accuracy score was used for performance measurement. Also, a confusion matrix for calculating the F1 score, commonly used for evaluating algorithm performance on imbalanced datasets was computed.

Choice of Optimisation Parameters and Results

Without PCA

- For the k-NN model we chose to optimise over a range of distance measures (Euclidean, Minkowski, Chebychev etc.) and tune k between 1 to 100 neighbours. Euclidean distance and 6 neighbours were the optimal parameters, which achieved an average CV accuracy of 96.7%. This model produced a F1 score of 0.959 (precision/recall: 1/0.921) on the test set.
- For the RF Model we chose to tune max features [1:10], the splitting criterion [Gini, Deviance] and the numbers of trees [10:200]. Here, the best HPs are 10 for the maximum number of random features per split node with Gini as the splitting criterion and 157 trees in the ensemble. These HP settings lead to an average CV accuracy performance of 95.9% and a F1 score of 0.953 (precision/recall: 0.938/0.968) on the test set. All these metrics for the test set can be easily recalculated from the confusion matrix (Figure 1). A classification error plot (Figure 2) displays how the CV, Out of Bag (OOB) [5] and test error progress through different numbers of trees within the RF. The first 20 trees seem to be crucial overall as the loss is heavily reduced during this stage.

With PCA

- The evaluation has been reconstructed with PCA. The explained variance percentage plot (Figure 3) for PCA analysis shows that the first 10 components account for approximately 95% of the variance in the data. We chose to proceed with these components for tuning our models. The tuned models with PCA produced an average CV accuracy of approximately 96.7% and 96.2% for k-NN and RF respectively.
- On the test data, compared to the cases above where PCA was not used, both k-NN and RF with PCA had a slightly worse F1 score of 0.95 and 0.935 respectively.

Performance on the test data	Confusion Matrix : k-NN (RF)	
	Predicted Malignant (Positive)	Predicted Benign (Negative)
True Malignant (Positive)	58 (61)	5 (2)
True Benign (Negative)	0 (4)	108 (104)

Figure 1: Confusion matrix on test set (without PCA)

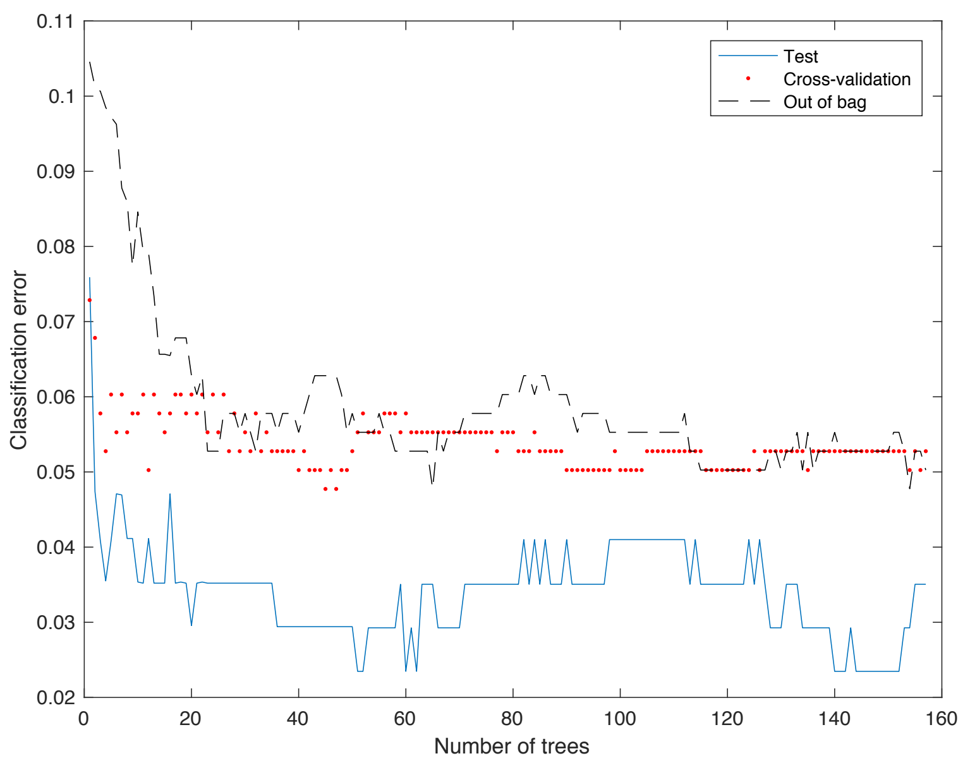


Figure 2: RF CV/OOB/Test loss

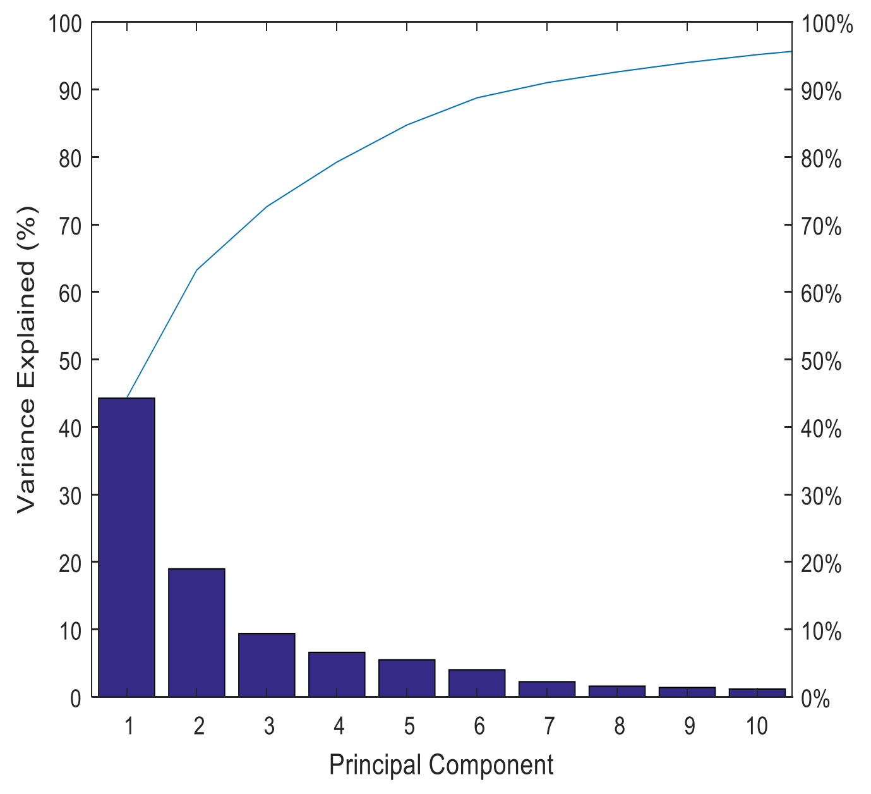


Figure 3: PCA explained variance ratio

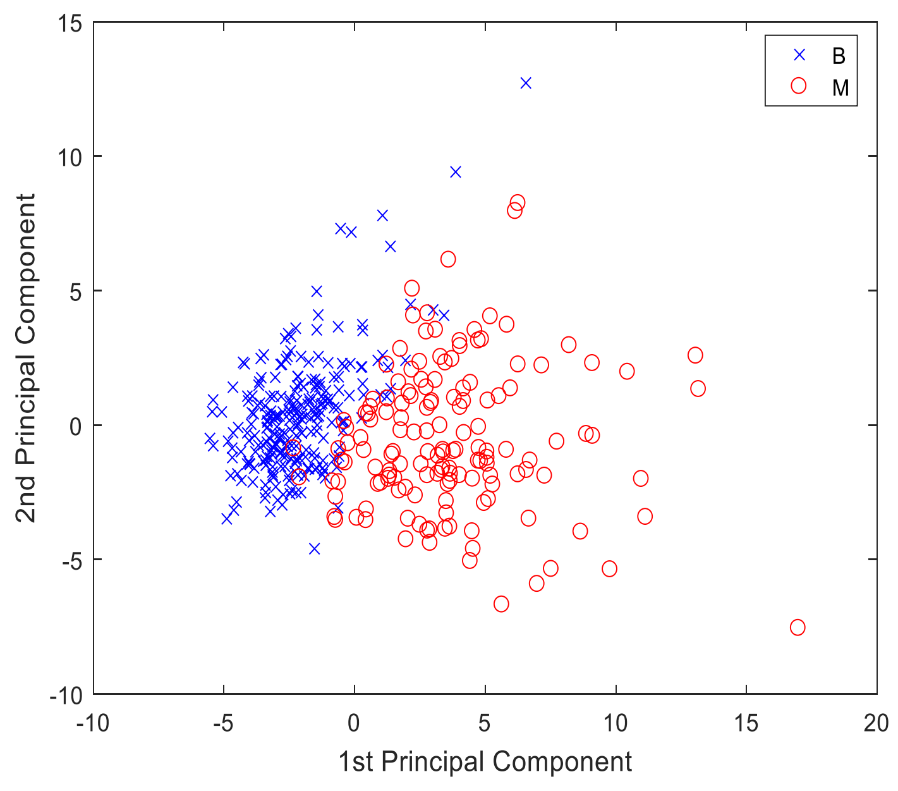


Figure 4: PCA 2D

Comparison and Critical Evaluation of ML models

Without PCA

- Both algorithms resulted in very high performance overall with relatively similar accuracy and F1 scores. However, what is surprising is that k-NN outperformed (training/test accuracy and f1 score) RF on this data set even though sometimes often criticised for its mediocre performance [4]. A reason for this could be that RF can suffer from too few informative features relative to noisy ones [5]. K-NN had an additional advantage on this data set because tuning as well as testing were much faster compared to RF. However, parallelising the tree building process in the RF ensemble might reverse this effect and could make RF more suitable for scaling to larger datasets.

With PCA

- PCA led to interesting results. When plotting the first two components (Figure 4) it can be observed, that the classes are almost linearly separable with only a few overlapping samples. Using the first 10 components, the performance of RF improves and k-NN stays the same. As described in the initial statistical analysis, we observe many highly (linearly) correlated features in our data which entail the same predictive power. Therefore, the original feature set not only slows down the training process, but might also contain more noise compared to the PCA features. This is probably the reason why RF was outperformed by k-NN on the full feature set which overall still performed best with 96.7% accuracy (F1: 0.959).

Other Observations

- In all clinical applications, the number of false negatives (FN) should be as close to zero so there are no misdiagnosed cases. In this aspect RF was the more robust choice compared to k-NN (only 2 FN).
- For the size of our dataset and choice of HP, the training time was not an issue. However for larger datasets, HP tuning can be time consuming. Random Forests allow parallelisation to be employed for scalability. In other cases, a balance must be achieved between computational time and model accuracy (e.g. tree depth, k value).

Conclusion

- The performance of an algorithm depends only on the data set. A more powerful algorithm might not always outperform a weaker one [4]. Both models perform well for classification of breast cancer. However, k-NN slightly outperforms RF (accuracy/ f1). PCA has more of an effect in improving RF performance relative to k-NN.
- Choosing the "right" performance metric is problem specific. In the health domain, recall and f1 score are more informative than just relying on the accuracy metric.

Future Work

- Considering the almost perfectly linear separability shown in the PCA plot, linear classifiers like logistic regression should perform extremely well and might even outperform k-NN and RF.
- Exploring the difference in just using the 10 original features rather than also including the additional correlated attributes as separate features. This would significantly reduce the dimensionality of the dataset.
- Explore the use of feature selection methods like Sequential Backward Selection to find possible smaller feature sets that lead to even higher performance.

1) Bishop, C. M. (2006). *Pattern recognition and machine learning*, vol. 1Springer. New York, (4), 12.
2) "Breast Cancer Wisconsin (Diagnostic) Data Set | Kaggle". *Kaggle.com*. N.p., 2016. Web. 16 Nov. 2016.
3) Breiman, L. (2001). Random forests. *Machine learning*, *45*(1), 5-32.
4) Caruana, R., & Niculescu-Mizil, A. (2006, June). An empirical comparison of supervised learning algorithms. In *Proceedings of the 23rd international conference on Machine learning* (pp. 161-168). ACM.
5) Friedman, J., Hastie, T., & Tibshirani, R. (2009). *The elements of statistical learning* (Vol. 1). Springer series in statistics Springer, Berlin.
6) Ruben Martinez-Cantin, BayesOpt: A Bayesian Optimization Library for Nonlinear Optimization, Experimental Design and Bandits. Journal of Machine Learning Research, 15(Nov):3735–3739, 2014.
7) Snoek, J., Larochelle, H., & Adams, R. P. (2012). Practical bayesian optimization of machine learning algorithms. In *Advances in neural information processing systems* (pp. 2951-2959)
8) Street, W. N., Wolberg, W. H., & Mangasarian, O. L. (1993, July). Nuclear feature extraction for breast tumor diagnosis. In *IS&T/SPIE's Symposium on Electronic Imaging: Science and Technology* (pp. 861-870). International Society for Optics and Photonics.
9) UCI Machine Learning Repository (https://archive.ics.uci.edu/ml/datasets/Breast+Cancer+Wisconsin+(Diagnostic)).Irvine, CA: University of California, School of Information and Computer Science.