

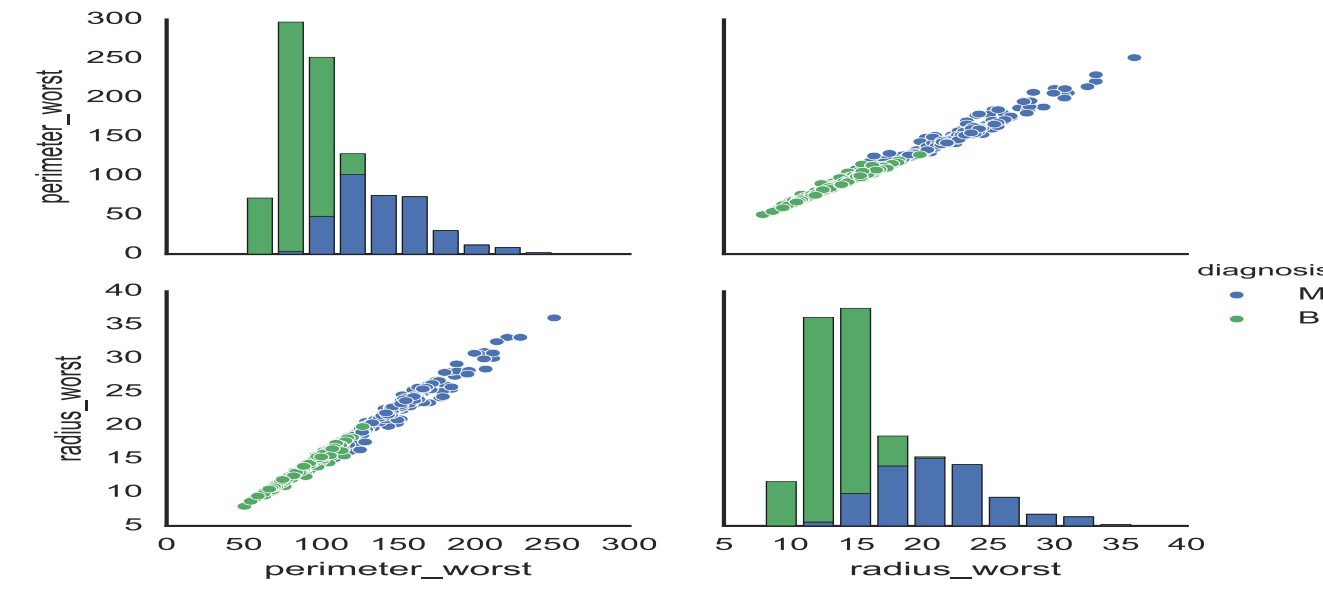
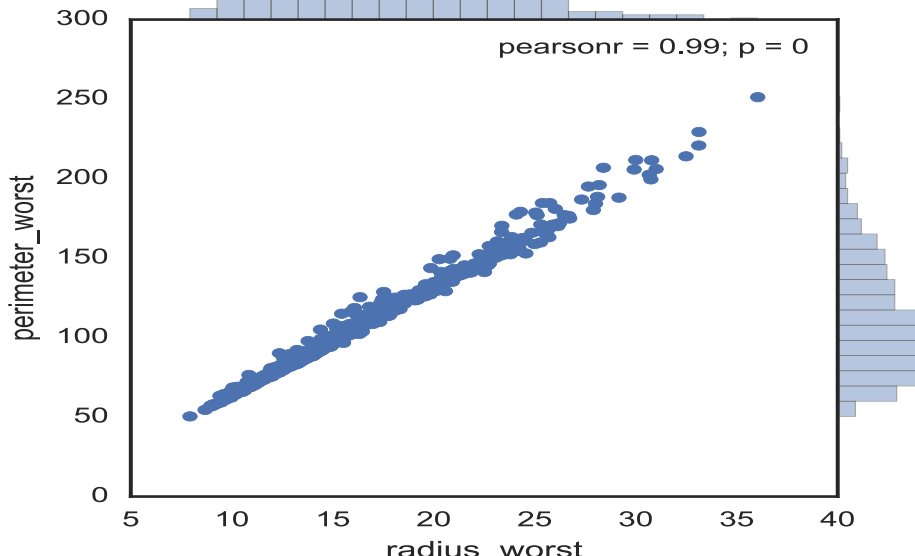
# Machine Learning in Healthcare: A Comparison of Random Forests and k-NN Algorithm Performance in Breast Cancer Classification

Ryan Nazareth and Hannes Draxl



## Motivation and Hypothesis

- 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) and Random Forests (RF) [3].
- Whilst we expect both algorithms to produce good results, we hypothesize that RF will perform better based on the ability to achieve a lower bias to outweigh the variance. This is achieved through numerous techniques like bootstrapping, random feature selection and majority voting [3]. 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		Initial Statistical Analysis of Dataset																																
<b>Random Forest</b> <ul style="list-style-type: none"><li>An ensemble of deep individual and independent trees built on bootstrap samples of the original data.</li><li>Instead of providing each feature at a split node, only a maximum amount of randomly sampled features are given.</li><li>This has a decorrelating effect on the individual trees in the ensemble.</li></ul> <b>Pros</b> <ul style="list-style-type: none"><li>Achieves high predictive performance on numerous tasks and is robust against overfitting.</li><li>Speeds up training process through parallelization.</li><li>Provides a real probability measure.</li></ul> <b>Cons</b> <ul style="list-style-type: none"><li>If the majority of random features are noisy, RF are likely to perform poorly [5].</li><li>Large amount of trees can be computationally expensive.</li></ul>	<b>K-NN</b> <ul style="list-style-type: none"><li>Memorizes training data instead of learning a discriminative function.</li><li>Uses a distance metric to find k-samples in the training data closest to a validation/test sample.</li><li>A majority vote assigns the class label.</li></ul> <b>Pros</b> <ul style="list-style-type: none"><li>Relatively simple to understand.</li><li>The memory approach of the k-NN-Classifier allows for an immediate adaption to additional and new training data.</li></ul> <b>Cons</b> <ul style="list-style-type: none"><li>Prone to the Curse of Dimensionality as samples become very distant to each other in higher dimensions.</li><li>Memory intensive as training samples have to be permanently stored [1].</li><li>Requires tuning optimal k and numerous distance metrics.</li></ul>	<ul style="list-style-type: none"><li>Breast Cancer Wisconsin (Diagnostic) Data Set [2][9] consisting of 569 samples and 32 columns.</li><li>The ID column was dropped from the feature set. The remaining 30 features can be broadly classified into 10 physical cell dimension measurements. The standard error and "worst" or largest (mean of the three largest values) of each of these measurements were computed for each image, resulting in 30 highly correlated features.</li><li>The target variable is binary consisting of 'benign' and 'malignant' tumor classes. The class distribution is 357 for benign and 212 for the malignant cancer class, representing a slight class imbalance.</li><li>The mean and standard deviation of the five highest correlated features (point biserial) with the target variable are summarised in the table on the right. The figures below display scatterplots and show the high correlation between features (0.99 Pearson) indicating redundant information. Most features are positively skewed. A similar behaviour in terms of the class distribution with regard to the feature value can be observed: the higher the feature value, the more likely it is that the cancer is malignant.</li></ul>																																
		<table><tr><th>Features</th><th>Mean B</th><th>Mean M</th><th>Std B</th><th>Std M</th></tr><tr><td>concave points_worst</td><td>0.07</td><td>0.18</td><td>0.03</td><td>0.04</td></tr><tr><td>perimeter_worst</td><td>87.00</td><td>141.37</td><td>13.52</td><td>29.45</td></tr><tr><td>concave points_mean</td><td>0.02</td><td>0.08</td><td>0.01</td><td>0.03</td></tr><tr><td>radius_worst</td><td>13.37</td><td>21.13</td><td>1.98</td><td>4.28</td></tr><tr><td>perimeter_mean</td><td>78.07</td><td>115.36</td><td>11.80</td><td>21.85</td></tr></table> <div></div>					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
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 MATLAB Bayesian optimization [6] approach for Hyperparameter Tuning (HP). This automatically models the generalisation performance "as a sample from a Gaussian process" [7].
- The performance of both k-NN and RF with and without the use of PCA dimensionality reduction was 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.

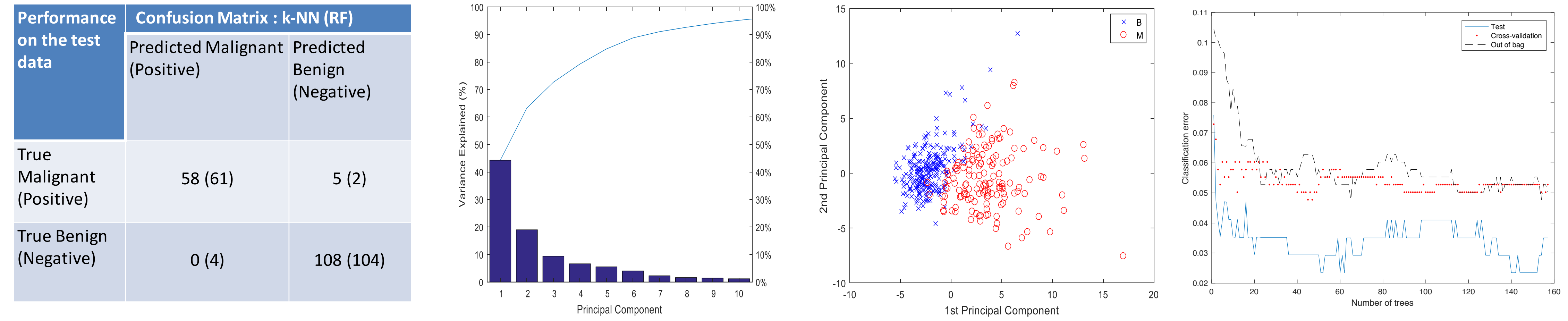
## Choice of Optimisation Parameters and Results

### 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. A classification error plot on the right displays how the CV, OOB 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 for PCA analysis shows that 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.



## Comparison and Critical Evaluation of ML models

- 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.
- PCA led to interesting results. When plotting the first two components (see Figure above) 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 but k-NN decreases slightly. As described in the initial statistical analysis, we observe many highly correlated (linear) 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).
- 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 HPs, 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).

Conclusions and Future Work	References
<b>Conclusion</b> <ul style="list-style-type: none"><li>The performance of an algorithm depends only on the data set. A usually 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). The use of PCA has more of an effect in improving RF performance relative to k-NN.</li><li>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.</li></ul> <b>Future Work</b> <ul style="list-style-type: none"><li>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.</li><li>Exploring the difference in using the mean of the features of the 10 groups rather than also including “standard error” or “worse” measurement as a separate attribute. This would significantly reduce the dimensionality of the dataset.</li><li>Explore the use of other feature selection methods : filter (K-best) or wrapper methods (Sequential Backward Selection).</li><li>Consider combining the two algorithms into a voting classifier.</li></ul>	<ol style="list-style-type: none"><li>Bishop, C. M. (2006). <i>Pattern recognition and machine learning</i>, vol. 1Springer. New York, (4), 12.</li><li>"Breast Cancer Wisconsin (Diagnostic) Data Set   Kaggle". <i>Kaggle.com</i>. N.p., 2016. Web. 16 Nov. 2016.</li><li>Breiman, L. (2001). Random forests. <i>Machine learning</i>, 45(1), 5-32.</li><li>Caruana, R., &amp; Niculescu-Mizil, A. (2006, June). An empirical comparison of supervised learning algorithms. In <i>Proceedings of the 23rd international conference on Machine learning</i> (pp. 161-168). ACM.</li><li>Friedman, J., Hastie, T., &amp; Tibshirani, R. (2009). <i>The elements of statistical learning</i> (Vol. 1). Springer series in statistics Springer, Berlin.</li><li>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.</li><li>Snoek, J., Larochelle, H., &amp; Adams, R. P. (2012). Practical bayesian optimization of machine learning algorithms. In <i>Advances in neural information processing systems</i> (pp. 2951-2959)</li><li>Street, W. N., Wolberg, W. H., &amp; Mangasarian, O. L. (1993, July). Nuclear feature extraction for breast tumor diagnosis. In <i>IS&amp;T/SPIE's Symposium on Electronic Imaging: Science and Technology</i> (pp. 861-870). International Society for Optics and Photonics.</li><li>UCI Machine Learning Repository (<a href="https://archive.ics.uci.edu/ml/datasets/Breast+Cancer+Wisconsin+(Diagnostic)">https://archive.ics.uci.edu/ml/datasets/Breast+Cancer+Wisconsin+(Diagnostic)</a>).Irvine, CA: University of California, School of Information and Computer Science.</li></ol>