IST 718 Final Project Report

Group 1

Mike DeMaria, Lu Guo, Haotian Shen, Casey Walsh

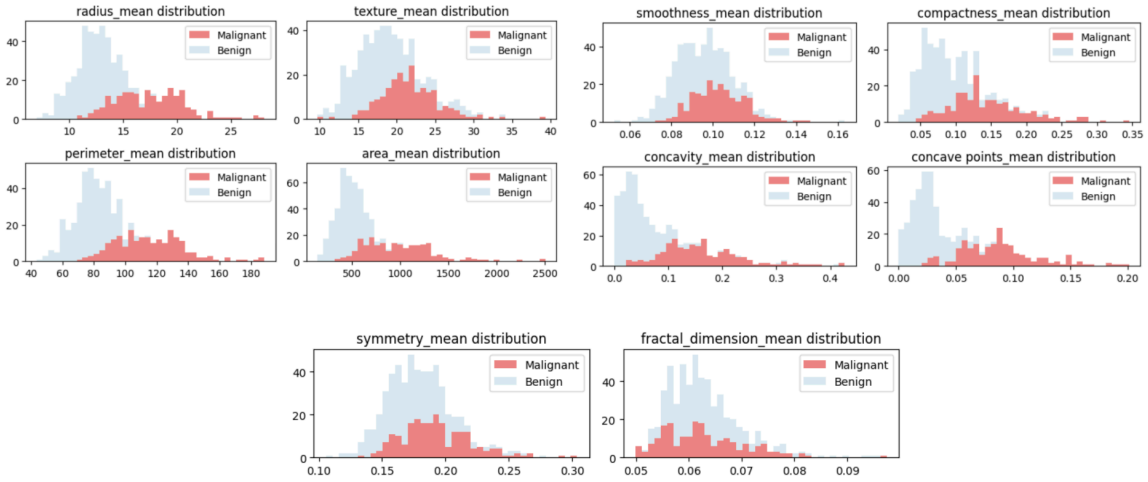
Introduction

In 1993, a new technique was discovered wherein machine learning could be used on fine needle aspirations to detect cancer cells. The cells in a sample are digitized, scanned, measured, and averaged. Our goal was to see if we could predict malignancy based on these values, matching or beating the originator's accuracy of 97%. Their model was initially published in a paper written by Street et al. (1993), The link to access the paper is https://minds.wisconsin.edu/bitstream/handle/1793/59692/TR1131.pdf;jsessionid=F231854E80A00FB8803122FDF9847940?sequence=1.

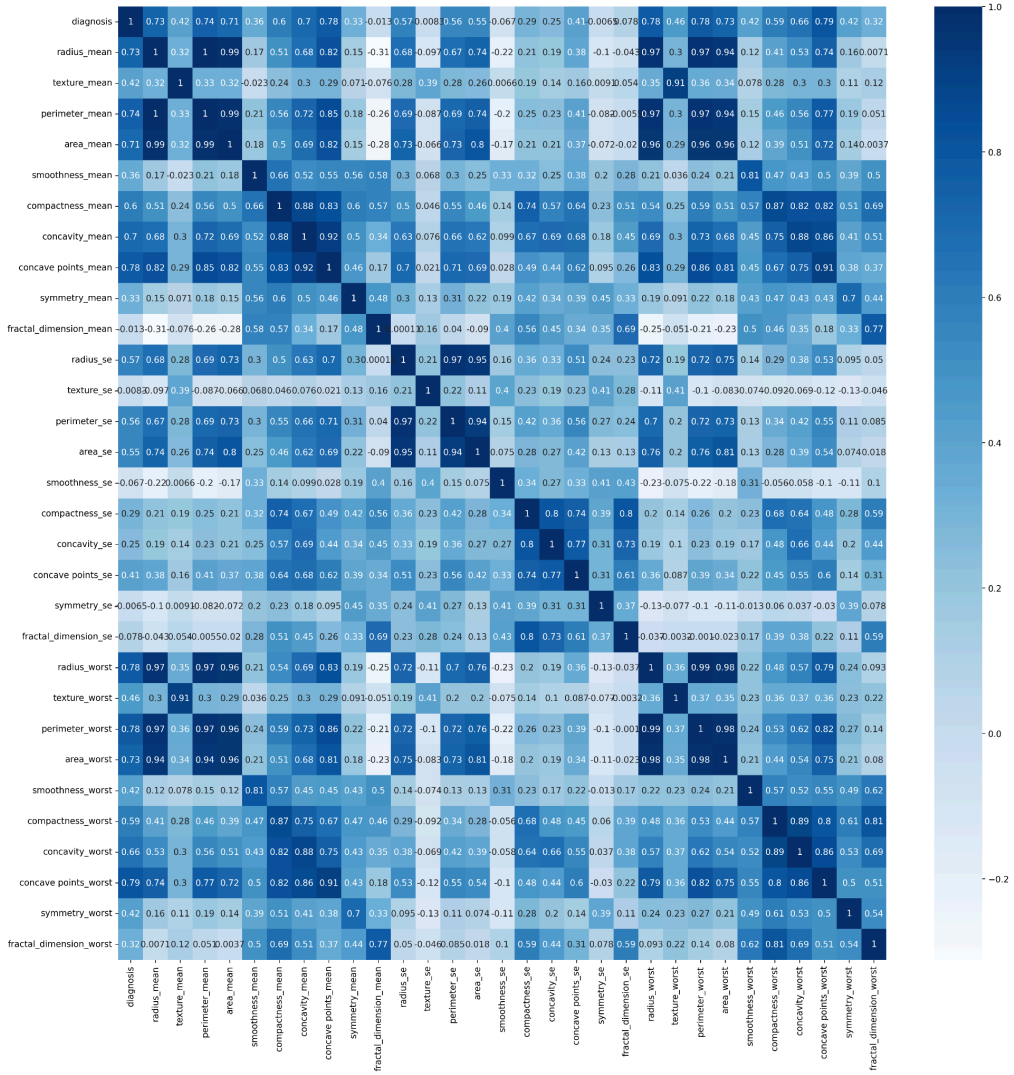
Data Exploration

The data is a benchmark dataset, which is still used in recent research projects (Vijayakumar et al., 2021). We obtained the Breast Cancer Wisconsin (Diagnostic) data set from Kaggle: https://www.kaggle.com/datasets/uciml/breast-cancer-wisconsin-data/data. This data set consists of 569 observations, 30 input variables, 1 output variable, and a patient ID column. All variables are numeric with no outliers, no data type mismatches and no missing values. The column "diagnosis" is a categorical prediction variable: it is B for benign or M for malignant. The data is split about 63% benign and 37% malignant. The other input variables consist of 10 types of measurements broken down into mean values, standard deviation and worst values (the mean of the 3 largest cells in a given sample).

We divide the data into two groups, one is benign, and the other is malignant. We want to see if there is any difference between the two groups. From the below images, we can see an obvious difference between the two groups on some variables like radius_mean. However, there is no significant difference between the two groups on symmetry_mean and fractual_dimension_mean.



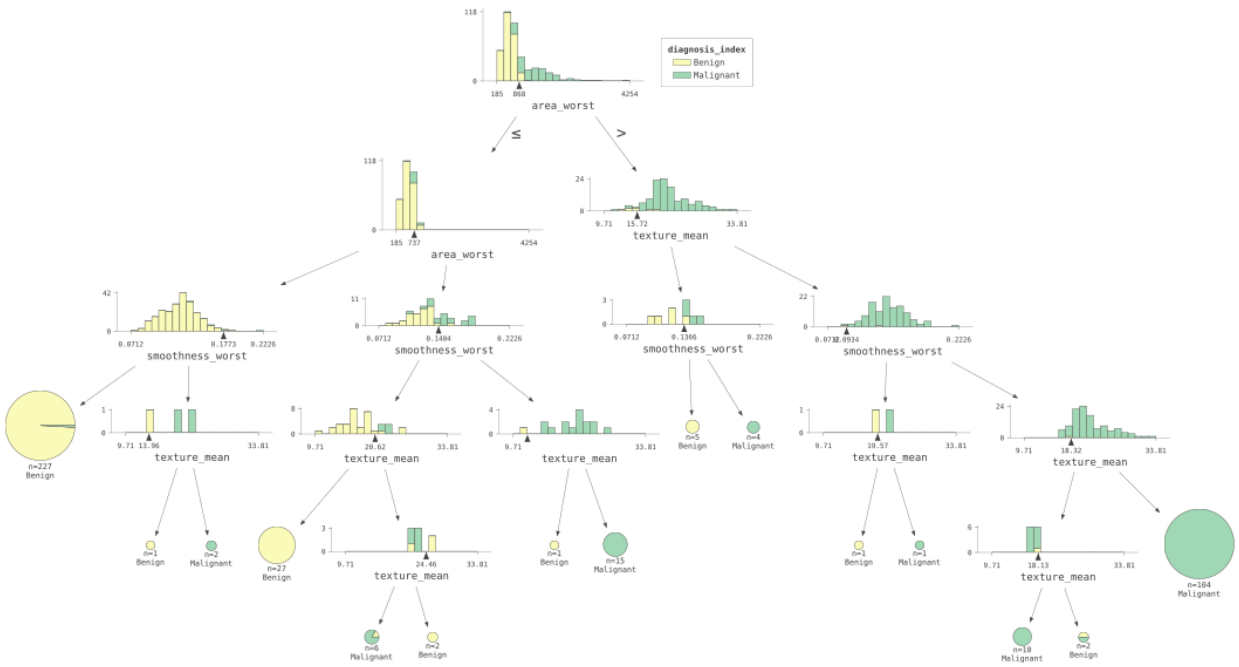
We produced a correlation matrix. It looks like radius, perimeter, area, and concave had the highest correlation to diagnosis, while fractal dimension, texture, smoothness and symmetry had the least. The standard deviation measurements of these metrics are less correlated than their mean values.



Original Experiment Reproduction

We reproduced the original experiment by creating a decision tree with texture mean, area worst and smoothness worst as the input variables. The paper states that small subsets of features were chosen to analyze, likely due to a lack of compute power or availability in the early 90s. The parameters used to train the model, or what choices the resulting tree made were not documented in the paper. As such, the model was trained with the default hyperparameters for decision trees set in PySpark. The dataset was split into 70% training and 30% testing. A model

was trained on the resulting training set with an accuracy of 94% and an AUC ROC of 0.92. A visualization of the this decision tree is below:

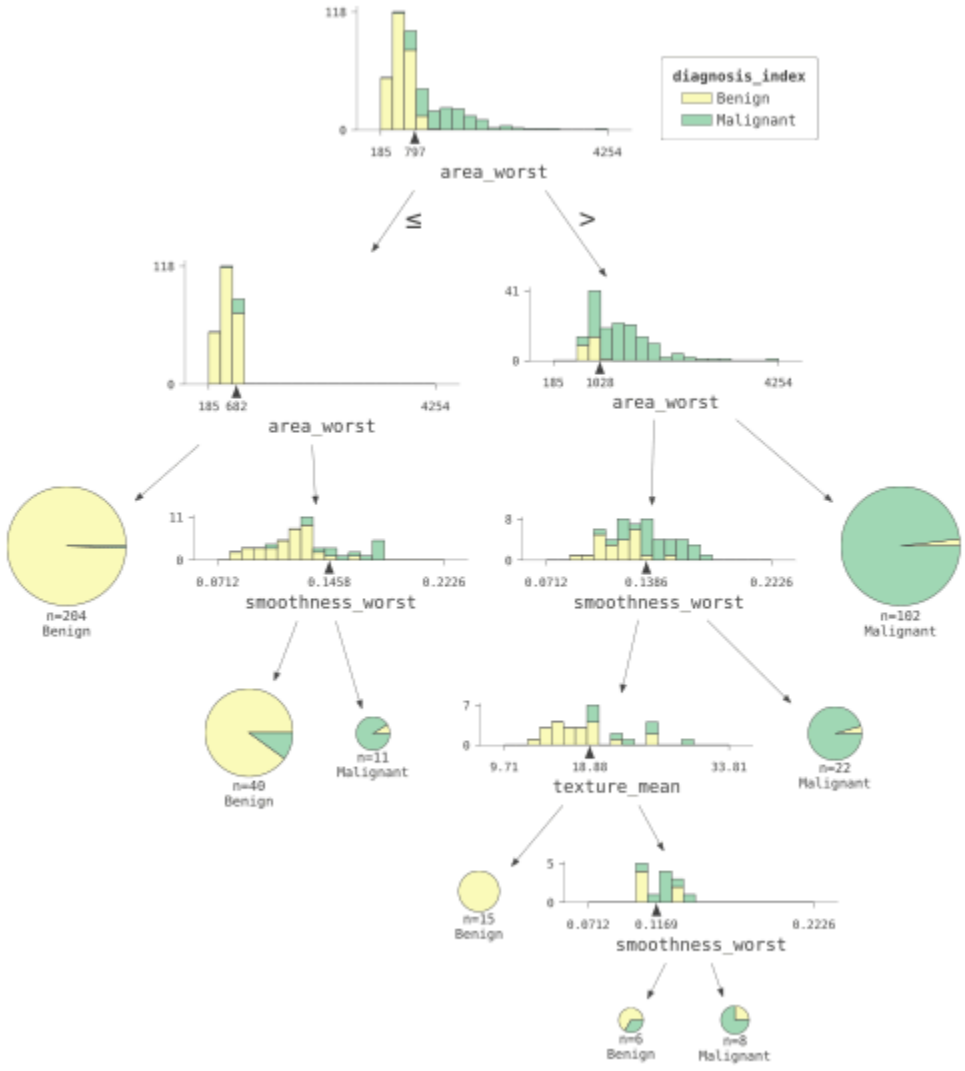


In an attempt to improve the accuracy of the model, we did a gridsearch of hyperparameters defined in the below table:

Parameter	Values
maxDepth	[2, 3, 5, 7, 9, 11]
maxBins	[8, 10, 32]
minInstancesPerNode	[1, 3, 5, 7, 9]

These 90 models were run at 5 fold cross validation, specifying a range of depth, instances per node and max bins from the table above. These models took the most time to train, at 556

seconds. The best model had a depth of 5, 15 nodes, 2 classes and 3 features, producing a model with 96.4% accuracy. This is close to the original experiment's 97.0%. The resulting decision tree is visualized below:



Neural Network

With the advances in computing power available today, we opted to build an artificial neural network to see if we could improve accuracy. A network was constructed consisting of two hidden layers: one with 32 nodes, the second with 16 nodes. Only the mean input columns

were used. The network took only 12 seconds to train. The output was a model with 93.08% accuracy, 93.14% precision, 93.08% recall and an F1-score of 93.01%. A second network was constructed, this time with just the standard error columns. The accuracy dropped to 90.57%, the F1-score of 90.36%. A third network was constructed with the “worst case” values. This model had 93.71% accuracy and 93.71% F1-score. Finally, a network was constructed with all input parameters. This model had an accuracy of 89.44%, precision of 89.47%, recall of 89.44% and F-score of 89.45%.

Then we tried creating one more complex neural network since the number of input features increases with all 30 feature columns. This time we added an additional hidden layer. The 64, 32, and 16 layer networks increased accuracy to 89.94%, precision to 89.95%, recall to 89.94% and F-score to 89.80%. The performance got a little better with a more complex neural network.

Since we built the heat map of the feature columns and the output column. According to the map, we found there were 5 features who has a low correlation with the output column. So we dropped them (fractal_dimension_mean, texture_se, smoothness_se, symmetry_se, fractal_dimension_se), and the performance of the complex model increased a lot with 92.45% of accuracy, 92.70% of precision, 92.47% of recall, and 92.31% of F1-score.

With some feature engineering, we improved the performance of the complex model. Although it’s not the best one, this is a heuristic way since the data scale is quite small. And we believe if there are more rows of data, the performance will continue to be better.

	accuracy	precision	recall	f1-score

All mean features	0.9308	0.9314	0.9308	0.9301
All standard error features	0.9057	0.9082	0.9057	0.9036
All worst features	**0.9371**	**0.9371**	**0.9371**	**0.9371**
All features	0.8944	0.8947	0.8944	0.8945
All features(large)	0.8994	0.8995	0.8994	0.8980
All features+Drop(large)	0.9245	0.9270	0.9245	0.9231

NOTE: *(large)* means the more complex neural network, *Drop* means dropping 5 low correlation value columns.

Additional Models

We decided to expand our supervised learning by trying out several other models, then iterating on the best ones. The first model was a simple logistic regression. This had an AUC of 99.7%. We tried a random forest. This model performed worse with an AUC of 98.0%. Finally, we tried gradient boosting trees. This model performed the worst with an AUC of 98.9%. It also took significantly longer to run, nearly 4 to 6 times as long as the others.

Given the higher AUC of the logistic regression, we iterated on that model. We tried scaling the inputs. The accuracy did not change. Next, we tried using only the “worst case” columns. This model did not perform better, with an AUC of 95.3% and an accuracy of 94%.

We tried the inverse, discarding just the “worst case” columns. This model performed significantly better, producing an AUC of 99.1% but had an accuracy of just 94%. We then

increased the number of iterations to a max of 10,000. The AUC went down to 98.3% but accuracy increased to 96%. The training time remained low at 5.7 seconds.

Finally, we tried a linear support vector classification. We split the data as 70% training, 30% testing. We used all available columns as input variables. The max iterations were set to 5,000. This model took 28.1 seconds to train, which is significantly longer than our other logistic regression models. However, this model also performed better than most of the other models. We had an AUC of 99.6%.

We narrowed down focus to our top two models: logistic regression and Linear SVC. We found that both models had similar top and bottom coefficients. Fractal dimension standard deviation was the biggest influencer at benignity, with a negative value that was far larger than any other variable. On the malignant side, concave points standard deviation was the most impactful.

We looked at the quality of the resulting models. Both had near identical AUCs, and identical accuracy of 98%. Linear regression has slightly worse precision (96% vs 98%), while the recall was better (100% vs 99%). The F-measure was 98% for linear regression and 99% for LinearSVC.

From our testing data set, linear regression produced 4 false positives and 0 false negatives. LinearSVC produced 2 false positives and 1 false negative. Comparing these two models, on the surface, LinearSVC produced fewer errors. However, it did produce a false negative. When it comes to a cancer diagnosis, we would err on the side of caution and prefer the model that produces less false negatives (Vomweg). Thus, we consider the logistic regression model the superior algorithm.

References:

Street, W. N., Wolberg, W. H., & Mangasarian, O. L. (1993, July). Nuclear feature extraction for breast tumor diagnosis. In Biomedical image processing and biomedical visualization (Vol. 1905, pp. 861-870). SPIE.

Vijayakumar, K., Kadam, V. J., & Sharma, S. K. (2021). Breast cancer diagnosis using multiple activation deep neural network. Concurrent Engineering, 29(3), 275-284.

Vomweg, Toni. (2021). Women Prefer False Positives Over Missed Breast Cancer, Survey Finds. Radiological Society of North America, Inc. Daily Bulletin.