Predicting breast tumor malignancy based on nuclear features

Introduction

Of all modern medical questions, cancer is still the most stubborn problem, compelling researchers and advocates in many different fields to pursue an answer. Diagnosing cancer is a long and difficult multi-disciplinary problem, and advancements in both medical knowledge and diagnostic technology are crucial to improving our understanding and our ability to treat and prevent cancer.

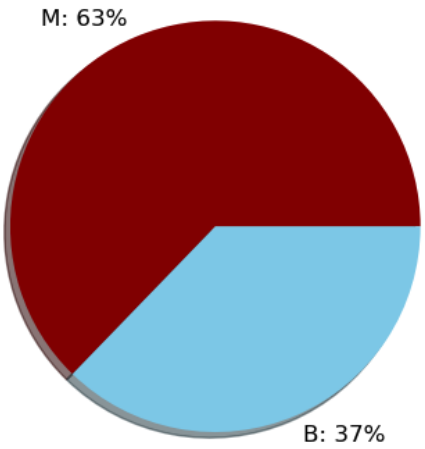
My goal with this project is to build a predictive model for breast cancer diagnosis based on the features of cancer cell nuclei. I intend for this model to be used as a tool for triaging breast cancer, ideally as one indicator in a set of other tools and methods. Triage is a crucial part of hospital intake, and is an especially difficult task for ER and ICU admissions, and providing more diagnostic tools will hopefully save lives as well as ensure the most effective use of limited hospital resources, including time, space, and personnel.

Dataset

These data were sourced from the Diagnostic Wisconsin Breast Cancer Database and were originally collected in 1993. The data were stored in a .csv and contained the mean, extreme, and standard error values for all 10 nuclear features. Each sample consists of nuclear feature data extracted from a fraction of a fine needle aspiration biopsy, which have been labeled either malignant (positive) or benign (negative).

I first ensured there were no duplicate samples and checked the class balance for malignant and benign diagnoses (Fig. 1).

Fig. 1: Class distributions



The set was slightly imbalanced but the imbalance was in favor of the class of interest (malignant). 13 cases contained 0 values for concavity and concave point features but all of these were negative (benign) samples so I kept them. Data for all other variables were present.

Feature analysis

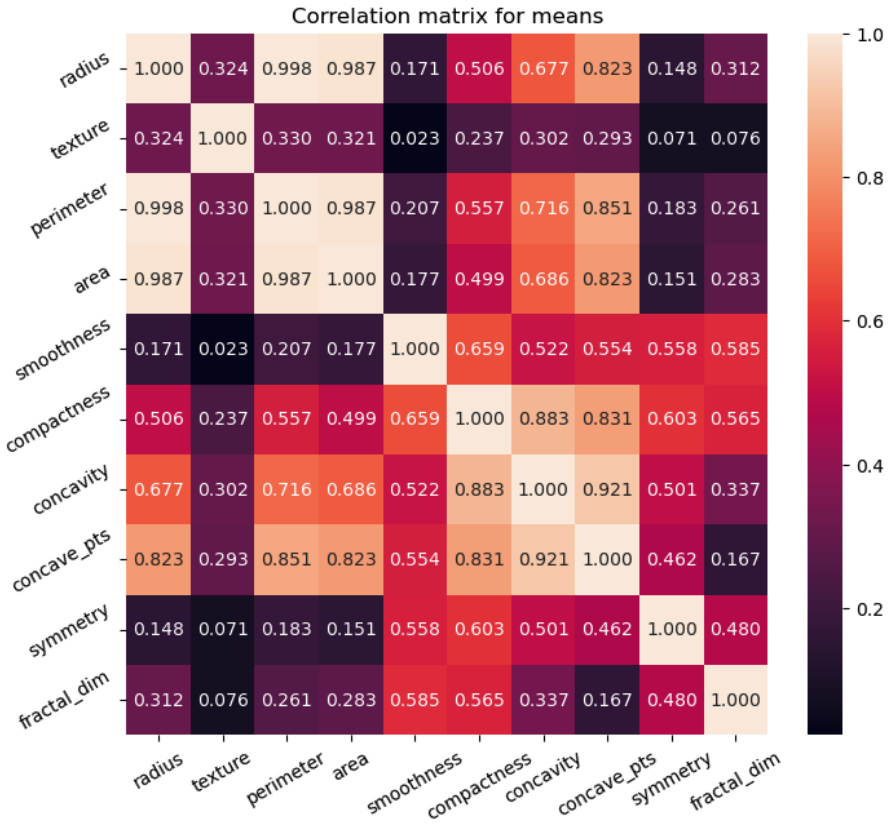
To explore the data, I first examined the distributions of each feature. My only concern was that the area measurements contained many outliers, so I intended to test the final model with and without the area features to see how it affected performance. None of the other features had unusual distributions. I scaled the data with StandardScaler and saved it separately for use in modeling.

Initially I also removed the standard error features, but this led to poor baseline performance in all three of the algorithms I tested, so I returned to the original feature space and ran the modeling process again, with better results. Although none of the standard error features stood out when looking at feature importances later on, it stood to reason at this point that knowing the variability of certain features could be helpful to the model.

Correlation matrix

Next, I made a correlation matrix to explore possible relationships between features. I found that area, perimeter, and radius were highly correlated, which made sense geometrically (Fig. 2).

Fig. 2: Correlation matrix, means



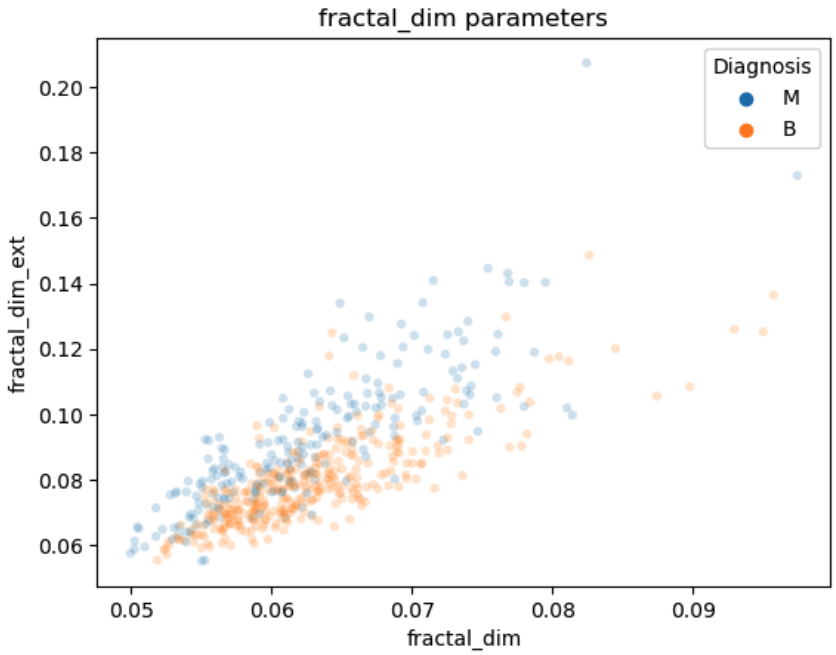
These features were also correlated with compactness, concavity, and concave points. The original research actually indicated that compactness was mathematically derived from perimeter and area

(perimeter^2 / area) so I planned to test subsets with and without these features later. Concavity and concave points were strongly correlated as well, which is not surprising since they are different measures of the same nuclear feature. Concave points is the number of contour cavities, concavity is the magnitude of the contour cavities as well as a measure of how they affect the overall shape of the nucleus.

I then made a second matrix with the most strongly correlated features removed so I could look into the possibility of more subtle relationships to help narrow down other subsets for modeling. This closer look showed that several features seemed to have some collinearity between their mean, extreme, and standard error measurements.

Scatter plotting these features did not show any unusually linear relationships but it did reveal that fractal dimension had quite a strong delineation between positive and negative classes (Fig. 3). Negative (benign) cases clearly tended to have lower worst fractal dimensions and vice versa for positive (malignant) cases.

Fig. 3: Fractal dimension, means vs. extremes



It was surprising at first to see that fractal dimension, which is a measure of contour irregularity using a coastline approximation of the perimeter, was not more closely correlated with perimeter. It is plausible that they could be independent; for example, a nucleus with more extreme irregularity would have a higher fractal dimension than another nucleus with the same perimeter, but less extreme irregularity. The second nucleus might actually be larger, but would have the same perimeter measurement. One is not necessarily related to the other, and looking at the correlation seemed to confirm this logic.

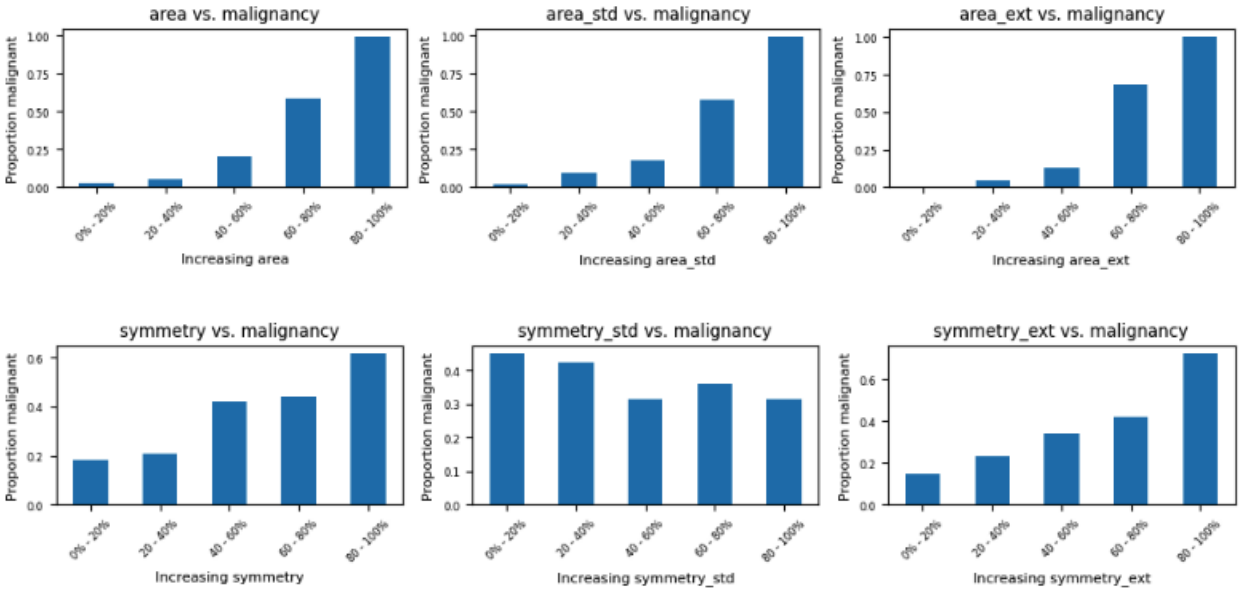
Discretization

Next, I analyzed the relationship between individual features and the target variable, diagnosis. This did show that most of the standard error measurements were positively correlated with malignancy, so it

verified that these would be useful data for the model to have. It also confirmed that malignancy increased consistently with magnitude for almost all features like the original research indicated.

Some features were exceptions. For example, while all three measurements for nuclear area showed strong positive correlations with malignancy, the standard error measurements for symmetry were only weakly correlated (Fig. 4). This was also true for texture and smoothness. The mean and worst measurements for these features remained positively correlated with malignancy.

Fig. 4: Discretization using pandas qcut: Area vs. symmetry



The fact that standard error remained consistent regardless of diagnosis for some features, like symmetry, seems to indicate that within a sample, these features tended to be the same for all nuclei. In other words, whether these features were homogenous (low standard error) or highly varied (high standard error) in a sample had no relation to malignancy. Since actual measurements for these features were still positively correlated with malignancy, it stands to reason that they might prove to be more reliable indicators of whether a particular biopsy is cancerous or not. If this is true, I would expect these features to be rank highly in feature importance and appear in the final feature subset.

For other features, like area, the greatest proportion of malignant cases were in samples that also exhibited a high standard error. This could mean that high variance is related to malignancy, or simply that these features tend to vary greatly regardless of individual sample properties. If the first case is true, I would expect the standard error measurements for these features to be quite important for the model to know. Again, feature selection later on will settle this question.

I also wanted to note that the original research indicated that all features "... are numerically modeled such that larger values will typically indicate a higher likelihood of malignancy." How this modeling was done was not explicitly described, so I was not able to make any decisions about feature scaling using this

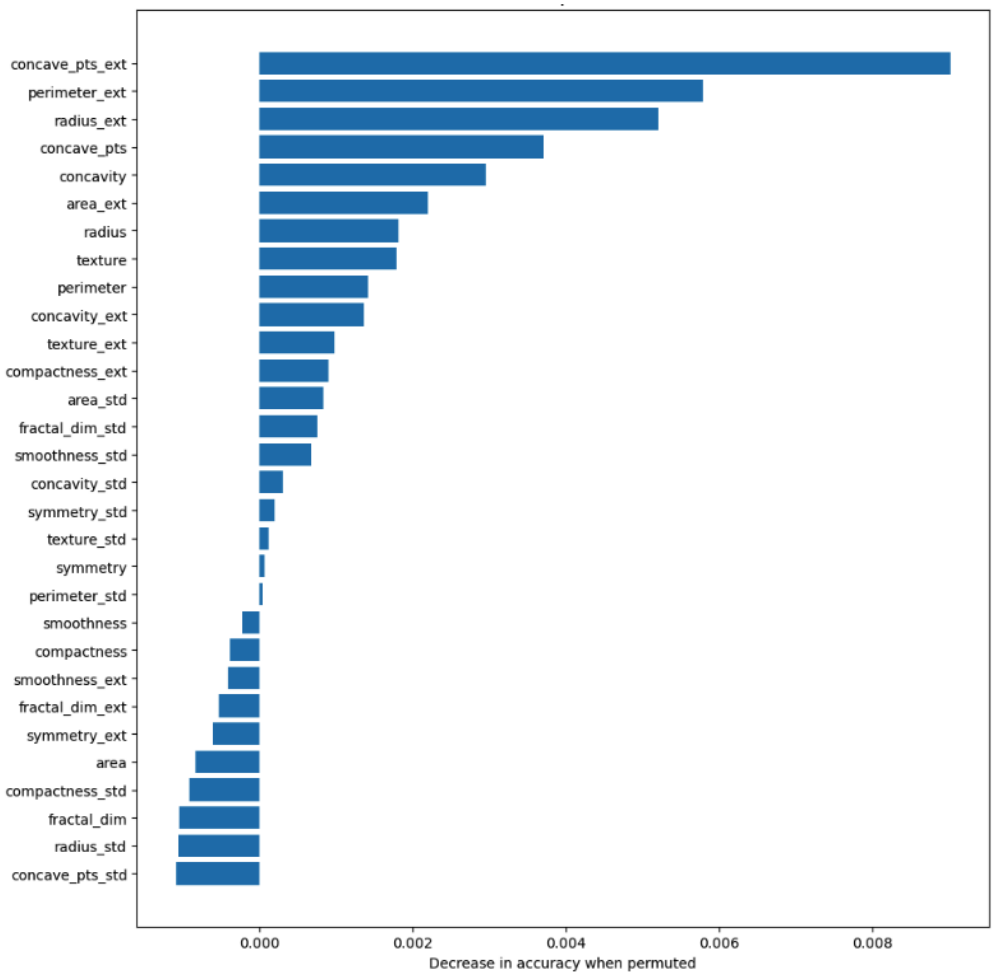
information, but the discretization of features still gave me a good idea of the behavior of each variable with relation to diagnosis.

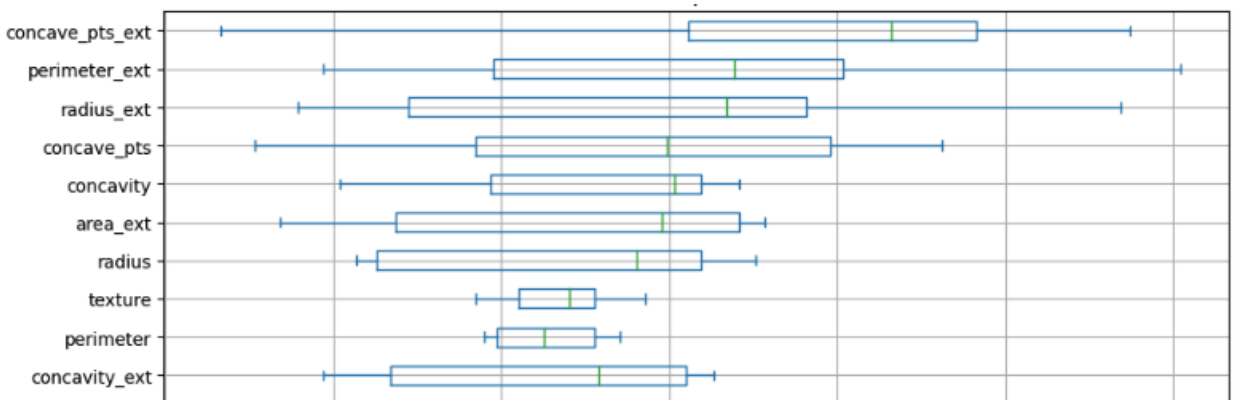
Random Forest feature importances, by permutation, using ROC-AUC scoring

Next, I used Random Forest to look at the feature importances. I used both feature importances and permutation importances. Feature importances are computed only on the fitted model and performance is not validated against a test set. Permutation importance, on the other hand, can measure how randomizing or shuffling a feature's values affects the model's prediction error. Since my goal was to build a predictive model, I decided that looking at permutation importance instead could be more useful for determining which features would best help the model generalize to unseen data.

I was able to split the training set (same as the set I used later on for modeling) into another training and validation set, which were smaller but still sufficient for EDA purposes. The training subset represented approximately 64% of the original data and the test subset represented about 4%. I used these to examine the fitted Random Forest model's prediction error then graphed the feature permutation importances to see which ones the model was relying on most heavily (Fig. 5).

Fig. 5: Permutation importances, using ROC-AUC scoring





Taking into account what I observed in the correlation matrices, the permutation importance test showed that for radius, perimeter, and area, the worst (extreme) measurements were much stronger predictors than the means and standard errors, so with this information I planned to test the final model using a subset with those features dropped.

Model selection

I divided the data into training and test sets using an 80/20 split. I chose three classifiers to test for model selection: logistic regression, stochastic gradient descent (SGD), and Random Forest. I used GridSearchCV to find the best parameters for each and compared their performances.

SGD was initially the worst performer and did not respond well to hyperparameter tuning until I fitted it again using the scaled data (scaled using StandardScaler earlier) and ran the grid search again.

Classifier	**Metrics**	**Hyperparameters**
Logistic Regression	roc-auc = **95.39** recall = **94.74** precision = **92.31**	C = 777.8 max_iter = 1000 penalty = l2 solver = newton-cg
Stochastic Gradient Descent (scaled using StandardScaler)	roc-auc: **93.42** recall: **89.47** precision: **94.44**	loss = modified_huber penalty = elasticnet
Random Forest	roc-auc = **95.39** recall = **94.74** precision = **92.31**	bootstrap = True criterion = entropy max_depth = 5 max_features = sqrt n_estimators = 20

The logistic regression and Random Forest models performed equally well, so to help decide between the two I plotted their precision-recall curves (Fig. 6a, 6b). Since I wanted a model that could most accurately diagnose malignant tumors, it was important to choose a model with strong recall. In a cancer diagnostic use case it is more important to ensure the model can correctly predict positive cases than to worry about false positives.

The precision-recall curves for the logistic regression model (Fig. 6a) and the Random Forest model (Fig. 6b) showed that while the logistic regression model had slightly higher recall, it also sacrificed much more precision than the Random Forest model (Fig. 6c). F1 scores were higher on the Random Forest model, and in considering the tradeoff I decided to move forward with the Random Forest model, since it was only outperformed by 2.6% on recall.

Fig. 6a: Logistic regression model precision-recall

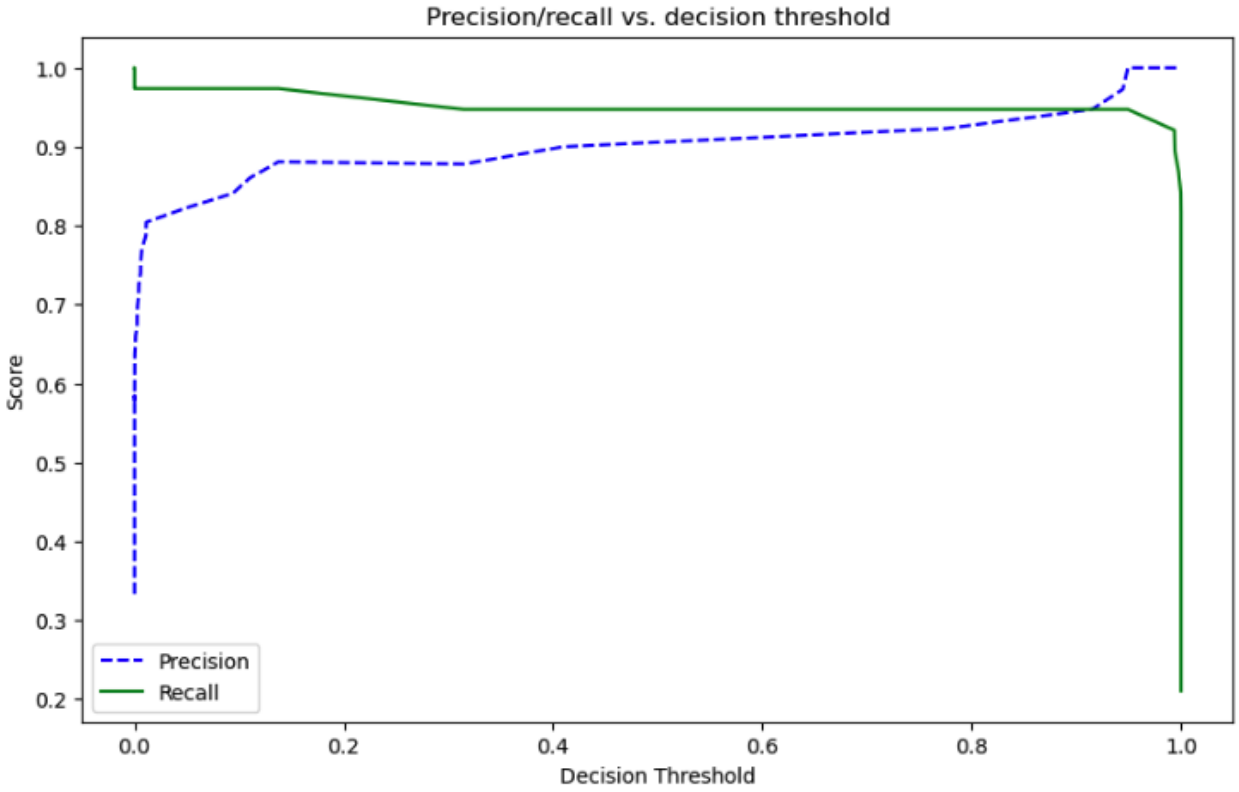


Fig. 5b: Random Forest model precision-recall

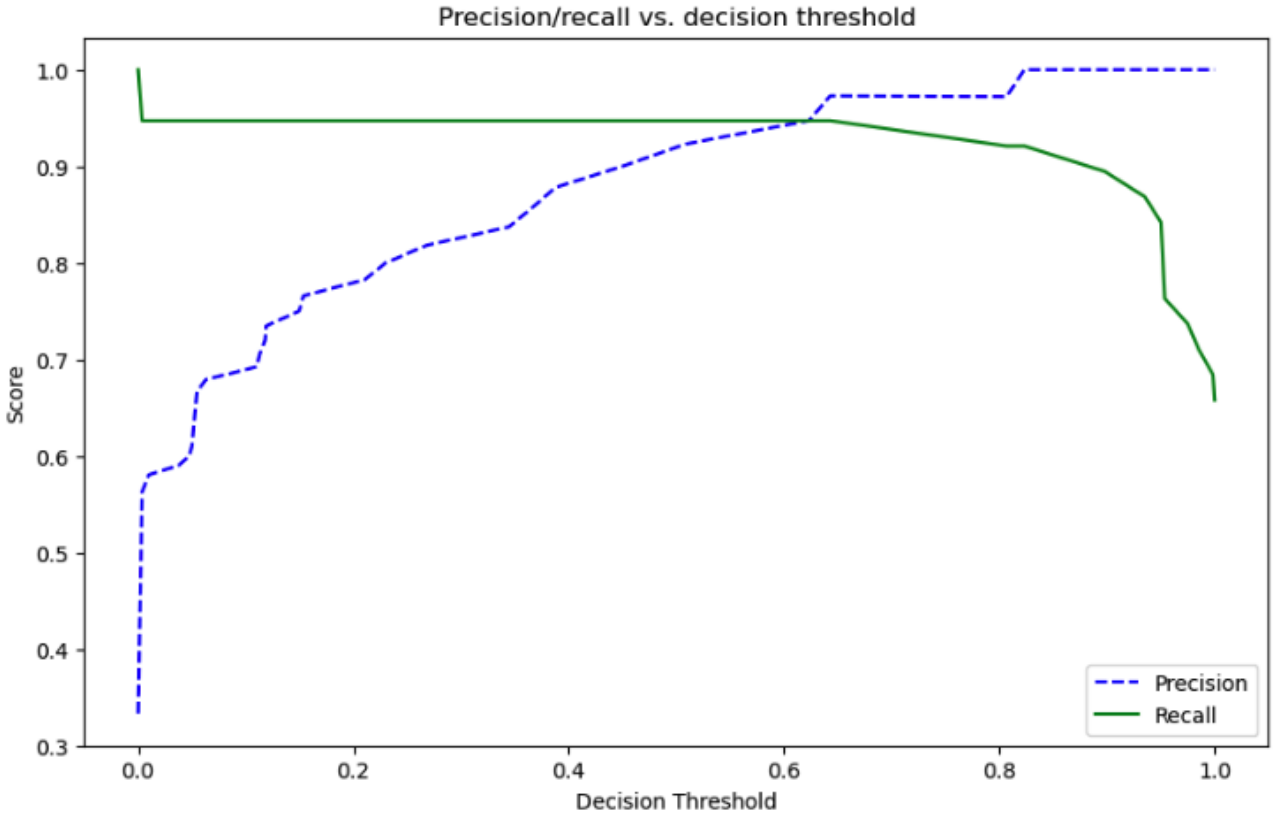


Fig. 5c: Classification reports

Logistic regression model at 0.13 threshold

	False	True	accuracy
precision	0.986111	0.880952	0.947368
recall	0.934211	0.973684	0.947368
f1-score	0.959459	0.925000	0.947368
support	76.000000	38.000000	0.947368

Random Forest model at 0.64 threshold

	False	True	accuracy
precision	0.974026	0.972973	0.973684
recall	0.986842	0.947368	0.973684
f1-score	0.980392	0.960000	0.973684
support	76.000000	38.000000	0.973684

Adjusting the final model

After selecting the final model, I explored different subsets of features to apply. In EDA I observed that many features were highly correlated, which means they can potentially skew the model. I tested the final model's performance on a variety of subsets to see which resulted in the best predictions.

I performed cross-validation on the training set with 6 different subsets of features, one of which was based on Random Forest permutation importances obtained earlier using ROC-AUC scoring and one of

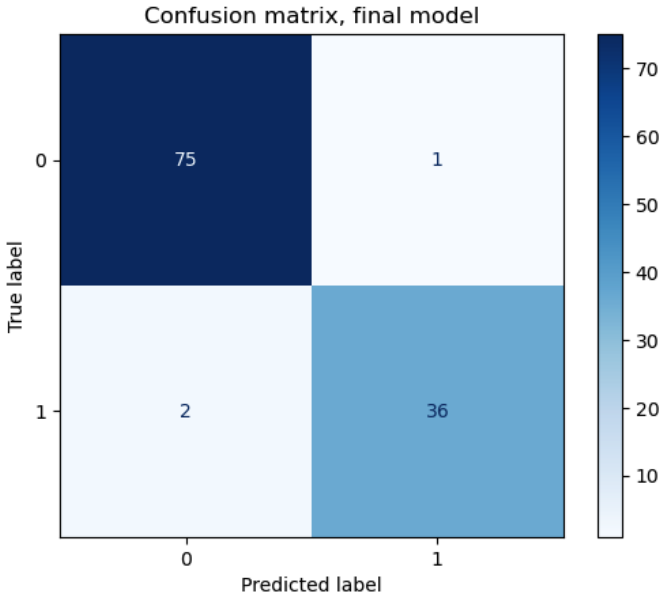
which was based on a PCA, which was not useful in the end. I narrowed down the feature space to 12 features: mean radius, mean texture, mean concavity, mean concave points, worst radius, worst perimeter, worst area, worst texture, worst smoothness, worst compactness, worst concavity, and worst concave points.

With these 12 features, the model achieved a recall of 94.7%, a precision of 97.3%, and an ROC-AUC score of 96.7% on the class of interest (Fig. 7a). The final model did produce 2 false negatives, but also predicted 36 true positives and 75 out of 76 true negatives (Fig. 7b).

Fig. 7a: Classification report on final model

	0	1	accuracy
precision	0.974026	0.972973	0.973684
recall	0.986842	0.947368	0.973684
f1-score	0.980392	0.960000	0.973684
support	76.000000	38.000000	0.973684

Fig. 7b: Confusion matrix on final model



Conclusion

One of the most interesting questions that came up in this problem was the usefulness of standard error as a feature. None of the standard error measurements showed up in permutation importances or in the final subset. I tested whether adding any of them would improve the model's performance but it had no effect. Yet when I initially started model selection, base performance (with default parameters) was stronger with the standard errors than without.

Even taking into the consideration the numerical modeling that the original researchers mentioned, the values for these features would have undergone the same processing as mean and worst measurements, so it still seems logical that this information should be informative for the classification. It was interesting to me that this was not the case, and even after repeatedly returning to the issue in EDA and modeling, the data kept giving me the same answer.

Returning to the discretization done earlier, the final feature space also did not include any standard error features and when I experimented with adding them, performance did not improve. I would therefore conclude, based on my analysis, that the measurements for the more homogenous features (texture,

smoothness, symmetry) were actually more useful for the model than the features that showed greater variability in each sample. In fact, both mean texture and worst smoothness appeared in the final subset of features on which the model made its best predictions.

After tuning and feature selection, the final model achieved an ROC score of 96.7%, a recall score of 94.7%, and a precision of 97.3%. In model selection, the logistic regression model did actually score about 2% higher in recall, but it also scored 9.2% lower in precision. Considering the nature of this problem it could be argued that the 2% gain in recall (in other words, the reduction of false negatives from 5% to 3%) would be worth the loss in precision.

This would come down to what purpose the model is put to—if it is used for diagnosis, then I think I would return to the logistic regression model, sacrifice the precision, and deal with the 9.2% increase in false positives. However, if it is used for another kind of task, for example estimating business costs for a hospital, I would again choose the Random Forest model, which preserves much more precision and would be more useful in making a budgetary decision.

In my initial approach, I was looking to create a tool that would help in the difficult task of hospital triage, so I chose the Random Forest model—while it is important to correctly triage positive cases, it is also not resource-efficient to sacrifice nearly 10% in precision. Cancer patients who are admitted to the ICU are often there for more than 30 days, and a triage nurse would have no way of predicting when patients come in or seeing the whole population per se. It is therefore more logical to preserve the precision of this model and rely on other diagnostic tools to compensate for the 2% loss in recall. This way, limited space can be reserved for patients with the most dire need, so that when they come in, they can be treated as quickly as possible.

If I were to return to this problem, I would be interested in also building an SVM to see if that might produce a better separation. I would also like to try adjusting for the class imbalance and going through the problem again with a stratified training sample to see if that produces different results in model selection or in performance.

Expanding on the problem, this model could also be used as part of a larger system for gauging the success of different treatments. Given more historical data, it could also be used to analyze changes or patterns in a patient's cancer over time, which could help signal recurrence or relapses. Also, though the data used in this problem were drawn only from a specific type of tumor (breast) there is no reason why the model could not be expanded to other kinds of cancer as well, which might improve the model's performance or open the door to creating a multi-class problem, where we would not only be able to predict whether a tumor is benign or malignant, but even to predict the severity of the case or the type of cancer we are looking at. There are many possibilities, and with such an important medical question, it is crucial that researchers in all disciplines explore as many of these avenues as possible.