Ben Gelman
Professor Hanke
SML 310
10/30/2021

1. Comparing Classification Models (70%)
1A.*Please use the scikit-learn breast cancer dataset (e.g. with the Python sklearn.datasets.load_breast_cancer() command), and perform a randomized (20%/80%) test/train split of the data to allow for model cross-validation in later steps.*

See Notebook

1B. *Please perform an exploratory data analysis on your training set to gain intuition for your modelling efforts. Do you notice anything interesting? Which features seem most important? Why?*

When creating linear models for each of the features, R^2 values vary widely. There are those with relatively high values between 50 and 60%, intermediate values of around 30-40%, lower values of 10-20%, and those whose R^2 hovers around 0. “Mean” features have relatively high R^2 values, and most of them have a relatively normal distribution or a slightly left skew. “Worst” features also have high R^2 values, slightly higher than those of “Mean,” and appear to have similar distributions to the “Mean” variables. The “Error” variables have the lowest R^2 values, and are less normal and more left skewed. Given these facts, I anticipate that ‘Worst’ variables will be the most important as an overall group, followed by ‘Mean’ and ‘Error.’

Almost all regression lines have a negative slope, although exceptions include Mean Fractal Dimension, Texture Error, Smoothness Error, and Symmetry Error, which has positive slopes close to 0. This leads me to anticipate that they will be less important to the model, as they show little relationship to the outcome variable.

The only variable with a positive slope of notable magnitude is Smoothness Error, but it has a very low R^2, so I do not think it will play a large role in the models.

There appear to be some Error variables whose overall values between the two classes of benign and malignant are similar, but whose distributions have several outliers that are driving much of the slope of the regression line. These include Texture Error, Compactness Error, Concavity Error, and Fractal Dimension Error. This may indicate that they will not end up being that significant to the models.

In terms of importance, Mean Radius, Mean Perimeter, Mean Area, Mean Concavity, Mean Concave Points, Worst Radius, Worst Perimeter, Worst Area, Worst Concave Points all have high R^2 values of around 50% or above, so I predict that they will be the most important to the models.

See Notebook.

1C. *Please create a binary classification model on your training set to predict which data samples are benign or malignant, and make a confusion matrix to report your results.*
See Notebook

Preliminary Random Forest Model
Accuracy: 100%
Confusion Matrix:

```
[[170,   0],
[  0, 285]])
```

1D. For this classification problem, what do you feel is the most appropriate way to measure the “goodness” of the model? Please carefully explain your reasoning!

This model would have to balance the need for accurately predicting the status of the tumor, as either benign or malignant, and the need for avoiding overfitting to this particular set of test observations. Thus, I would be skeptical of a model that has extremely high accuracy on this data but that only uses one approach to make predictions. I chose random forest as my first model for this reason, as it makes multiple decision trees and lets them vote on a prediction, making it less susceptible to overfitting.

Importantly, as this is a model used for detecting whether breast cancer tumours are benign or malignant, a 'good' model would maximize the rate of true positives. This would be defined as the recall, or True Positives divided by True Positives plus False Positives.

1E. Please repeat part 1c for each of the following types of classification models, choosing parameters that seem appropriate to give a good model. For each model you make, record the *confusion matrix and how well the model performs for the "goodness" metric you decided on in part 1d:*

Training Data Results:

i. Nearest Neighbors
Accuracy: 93.8%
Recall:95.5%
Confusion Matrix:

```
[[149  21]
 [  7 278]]
```

ii. Naive Bayes
Accuracy: 93%
Recall: 96.7%
Confusion Matrix:

```
[[143  27]
 [  5 280]]
```

iii. Logistic Regression
Accuracy: 95.6%
Recall: 95.2%
Confusion Matrix:

```
[[158  12]
 [  8 277]]
```

iv. Support Vector Machines
Accuracy: 97.4%
Recall: 96.5%
Confusion Matrix:

```
[[164   6]
 [  6 279]]
```

v. Decision Trees
Accuracy: 99.8%
Recall: 100%
Confusion Matrix:

```
[[169   1]
 [  0 285]]
```

vi. Random Forests
Accuracy: 99.8%
Recall: 100%
Confusion Matrix:

```
[[169   1]
 [  0 285]]
```

I first tested the models using default conditions, then created new ones with altered parameters. These models all have very high accuracy and recall on the training data. Their high accuracy makes me a bit concerned about overfitting, despite implementing some parameters aimed at mitigating that risk. My parameter tweaks were also aimed at increasing recall.

My parameter changes included increasing the smoothing in Naive Bayes. I also increased the minimum number of samples required to split a node and limited tree depth for Decision Trees and Random Forest. I then increased the amount of neighbors to use for KNN calculations and limited the number of trees for Random Forest. Finally, I decreased the regularization parameter and increased the tolerance for stopping SVC, which was mainly aimed at increasing recall.

1F. Which model performed the best on the training data, and how well did they perform in general?

In terms of both accuracy and avoiding false negatives, the Random Forest and Decision Trees models performed best on the training data, achieving a score of 99.8% for accuracy and 100% for recall. In general, the models performed well, with the rest of the accuracy rates ranging from 93% for Naive Bayes to 97.4% for SVM. Apart from the perfect recall score for Random Forest and Decision Trees, recall scores ranged from 96.6% for Naive Bayes to 95.2% for Logistic Regression.

1G. Which model performed the best on the testing data, and how well did they perform in general?

Testing Data Results:

KNN
k-Neighbors Accuracy: 91.2%

```
Recall: 97.1%
Confusion Matrix:
[[33  9]
 [ 1 71]]

Naive Bayes
Accuracy: 89.5%
Recall: 96.9%
Confusion Matrix:
[[31 11]
 [ 1 71]]

Logistic Regression:
Accuracy: 95.6%
Recall: 100%
Confusion Matrix:
[[37  5]
 [ 0 72]]

SVM
Accuracy: 95.6%
Recall: 100%
Confusion Matrix:
[[37  5]
 [ 0 72]]

DT
Decision Trees Accuracy: Accuracy: 95.6%
Recall:97.4%
Confusion Matrix:
[[38  4]
 [ 1 71]]

RF:
Accuracy: 94.7%
Recall: 97.4%
Confusion Matrix
[[37  5]
 [ 1 71]]
```

Logistic Regression, SVM, and Decision Trees performed the best on accuracy, all with the highest accuracy of 95.6%. Logistic Regression and SVM achieved the highest recall of 100%.

All the other models also performed comparably well. Other accuracy rates ranged from 94.7% for Random Forest to 89.5% for Naive Bayes, and recall ranged from 97.4% for Decision Trees to 96.9% for Naive Bayes.

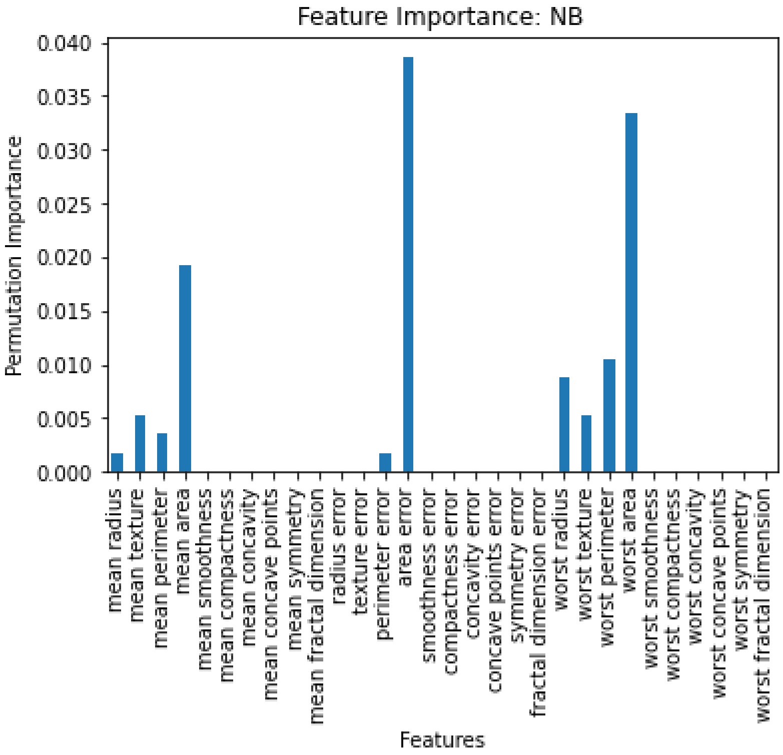
1H. Compare the model performance on the testing and training data to see if any of your models were overfit!

Models performed similarly or even improved between training and testing data. Logistic Regression stayed about the same on accuracy and improved on recall. Naive Bayes, KNN, and SVM lost some of their accuracy but gained slightly in recall. Decision Trees and Random Forest lost both accuracy and recall, but not by very significant amounts.

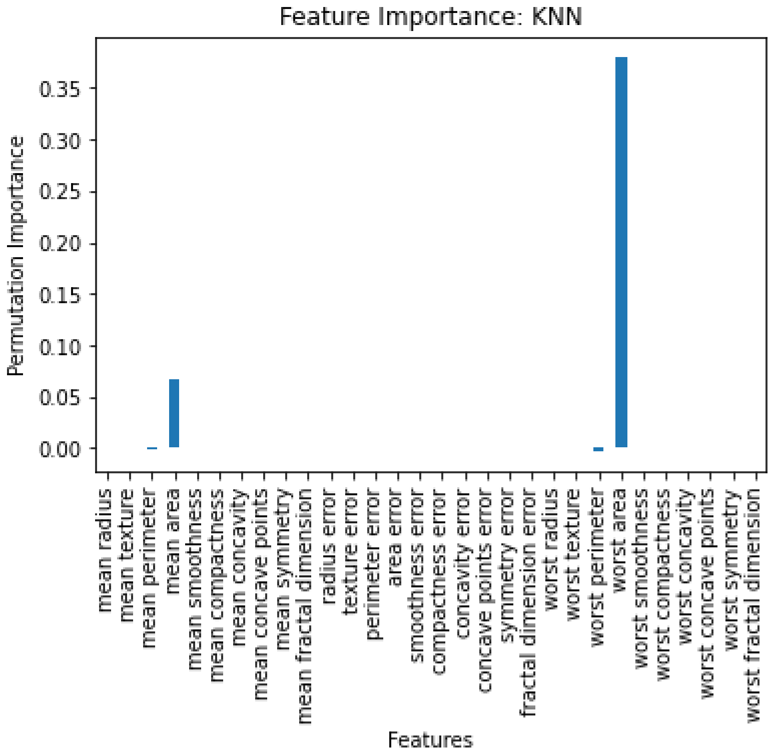
I was concerned that the Random Forest and Decision tree models might be highly overfit, given that they had perfect or close to perfect scores on both recall and accuracy on the training data. However, with their high accuracy and recall on the test set, I am satisfied with their performance on the testing data.

2. Explaining your Model (20%)
2A. Please explain which features are the most important in each of your models in part 1e? Is this similar across all models? How does this agree with your expectations from part 1b? Please carefully explain your reasoning

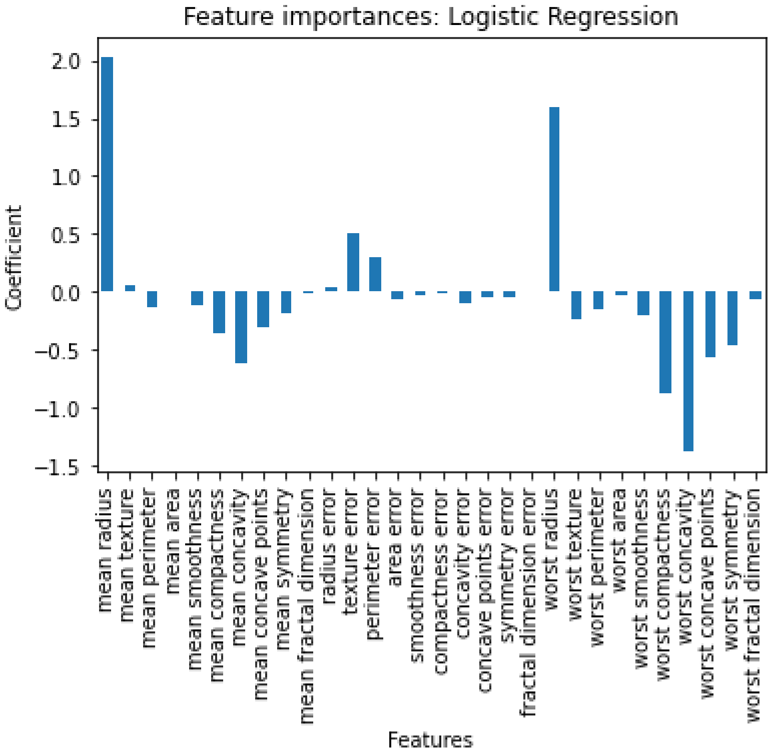
For Naive Bayes, when measuring by permutation importance, which illustrates how much worse the model performs if a given feature is randomly shuffled, the most important features were Area Error, Worst Area, and Mean Area.



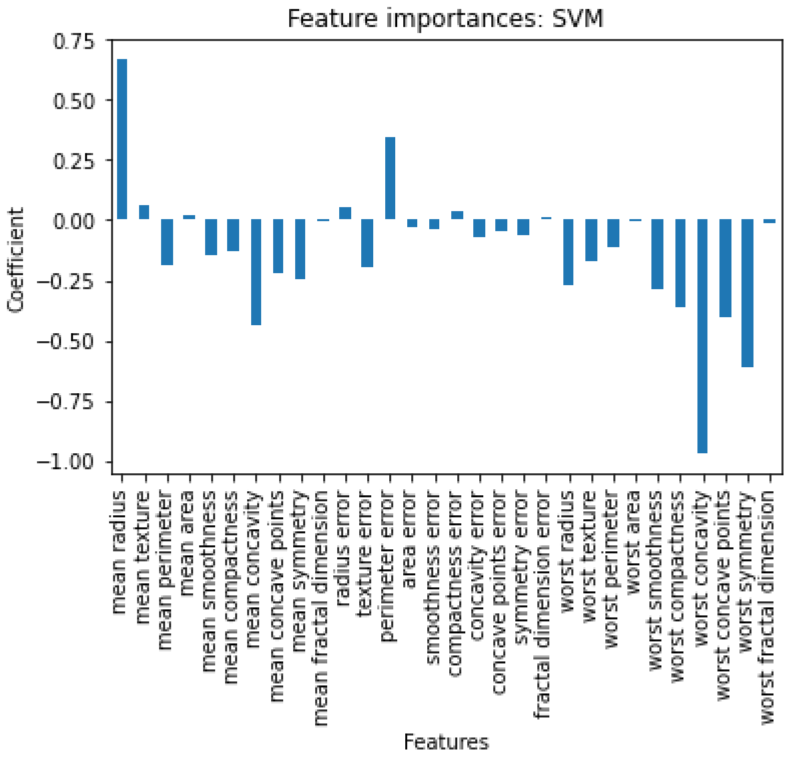
By the same metric, Worst Area was by far the most important for KNN.



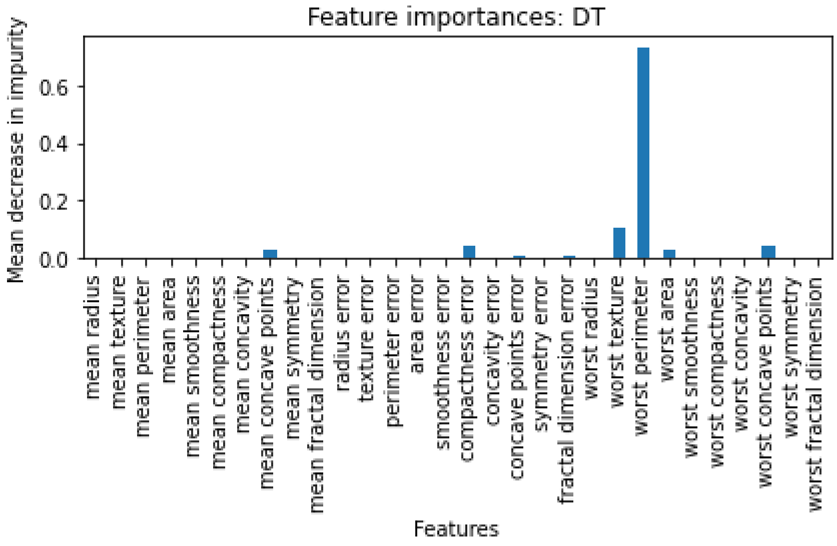
I measured importance for Logistic Regression and SVM by comparing the size of the different variables' coefficients in the models. For Logistic Regression, Mean Radius, Worst Radius, and Worst Concavity played large roles, along with Worst Compactness, Mean Concavity, Worst Concave Points, and Worst Symmetry.



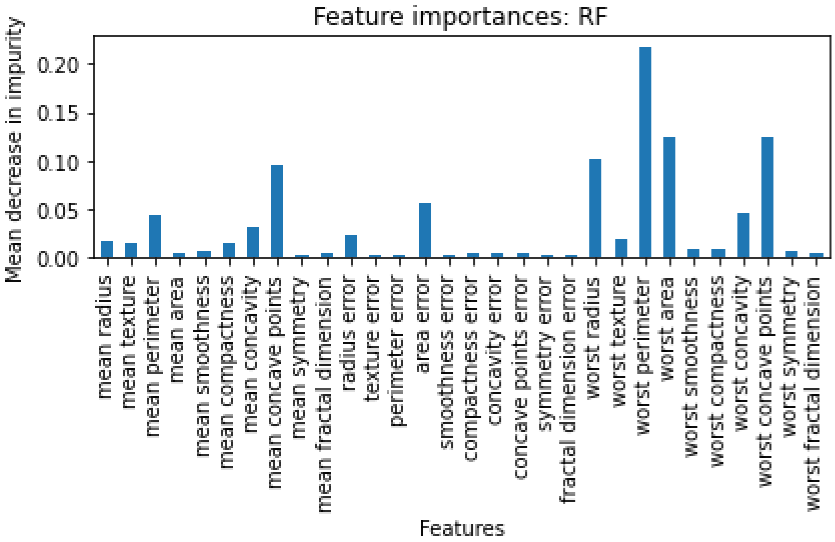
In SVM, Mean Radius, Worst Concavity, Mean Concavity, Perimeter Error, Worst Symmetry, Worst Concave Points, and Worst Compactness were the most important.



For the Decision Tree and Random Forest Models, importance is measured by the decrease in impurity when splitting along a given variable. Worst Perimeter was by far the most influential for Decision Tree.



Worst Perimeter was also the most important for Random Forest, though in this model Worst Concave Points, Worst Area, Worst Radius, and Mean Concave Points also play large roles.



Many important features are shared between models, such as Worst Area in both Naive Bayes and KNN, or Mean Radius between Logistic Regression and Mean Radius. Some features are important between 3 models, such as Worst Area and Worst Concave Points which are prominent in Naive Bayes, KNN, and Random Forest. Nontheless, the models are still quite heterogeneous in terms of which features are important, with no feature being very important to more than 3 models.

I predicted that variables with high R^2 values would end up being more important to the models, and both Worst Area and Worst Concave Points, which are the most broadly shared important features, have high R^2 values (54% and 63% respectively), so in that sense this does line up with my predictions.

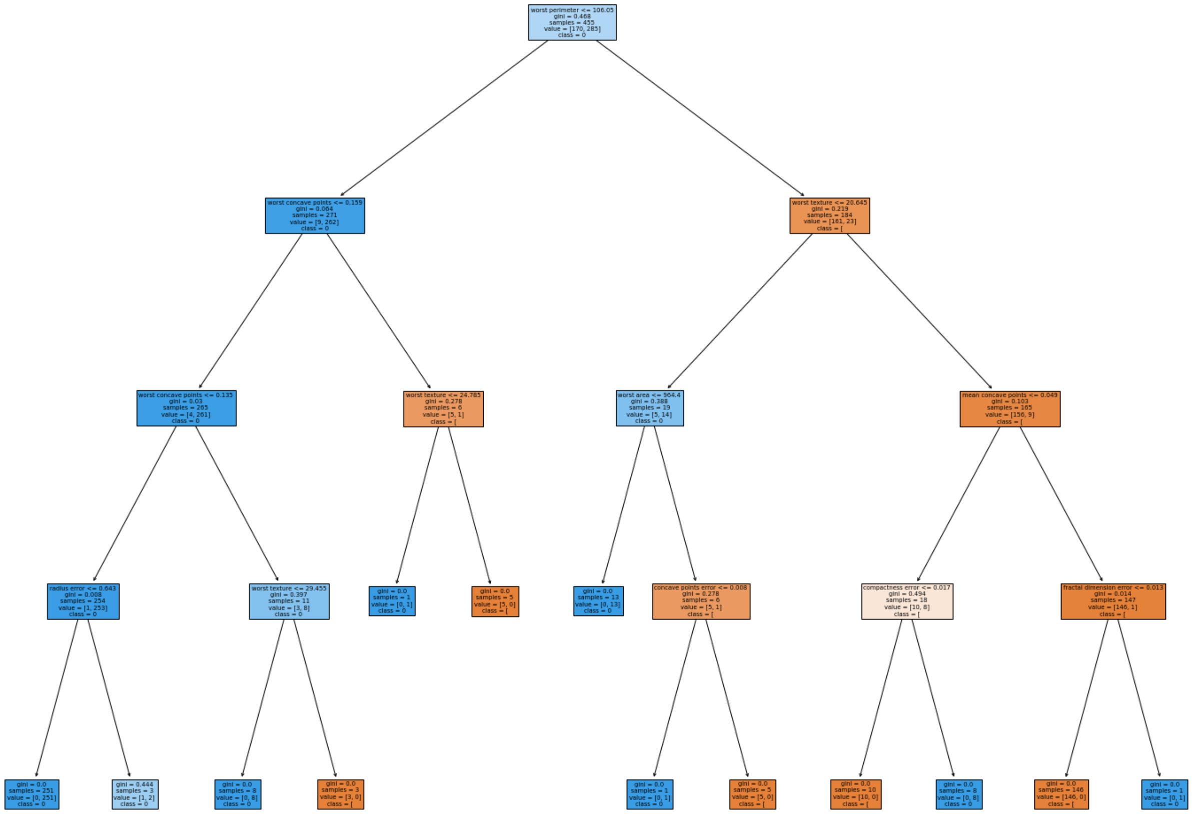
Very few of the variables with low R^2 values ended up being very imporatant to any of the models, with one exception being Worst Symmetry which was important in both Logistic Regression and SVM, so that also supports my prediction.

Another pattern I noticed is that the important variables tend to belong to the 'Worst' or 'Mean' categories, with only 'Area Error' and 'Perimeter Error' being prominent in Naive Bayes and SVM. This also aligns with my prediction, as I noted that variables of the 'Error' category had lower R^2 values than 'Worst' or 'Mean.' 'Worst' variables are also generally more important than 'Mean' ones, following the trend of R^2 values predicting importance.

Lastly, I predicted that Mean Fractal Dimension, Texture Error, Symmetry Error, which have slopes close to 0, would not be important to the models, which was correct.

2B. Of all of the models in 1e, which was the most explainable? Why?

The Decision Tree model is the most explainable, as it has the most disproportionate ratio of importance among its features, with Worst Perimeter leading to a far greater decrease in impurity than any other attribute. It is thus much easier to understand how the model comes to its decision, It is also easy to visualize and understand exactly how the tree comes to its conclusions. This contrasts with models such as Random Forest, Logistic Regression, or SVM which use a more diverse set of variables to make their predictions.

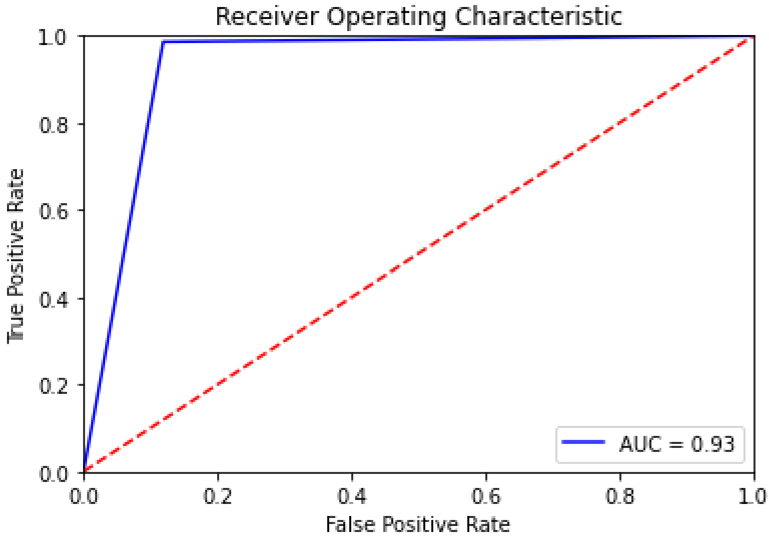

worst perimeter <= 106.05 gini = 0.468 samples = 455 value = [170, 285] class = 0
worst concave points <= 0.159 gini = 0.064 samples = 271 value = [9, 262] class = 0
worst texture <= 20.645 gini = 0.219 samples = 184 value = [161, 23] class = [
worst concave points <= 0.135 gini = 0.03 samples = 265 value = [4, 261] class = 0
worst texture <= 24.785 gini = 0.278 samples = 6 value = [5, 1] class = [
worst area <= 964.4 gini = 0.388 samples = 19 value = [5, 14] class = 0
mean concave points <= 0.049 gini = 0.103 samples = 165 value = [156, 9] class = [
radius error <= 0.643 gini = 0.008 samples = 254 value = [1, 253] class = 0
worst texture <= 29.455 gini = 0.397 samples = 11 value = [3, 8] class = 0
gini = 0.0 samples = 1 value = [0, 1] class = 0
gini = 0.0 samples = 5 value = [5, 0] class = [
gini = 0.0 samples = 13 value = [0, 13] class = 0
concave points error <= 0.008 gini = 0.278 samples = 6 value = [5, 1] class = [
compactness error <= 0.017 gini = 0.494 samples = 18 value = [10, 8] class = [
fractal dimension error <= 0.013 gini = 0.014 samples = 147 value = [146, 1] class = [
gini = 0.0 samples = 251 value = [0, 251] class = 0
gini = 0.444 samples = 3 value = [1, 2] class = 0
gini = 0.0 samples = 8 value = [0, 8] class = 0
gini = 0.0 samples = 3 value = [3, 0] class = [
gini = 0.0 samples = 1 value = [0, 1] class = 0
gini = 0.0 samples = 5 value = [5, 0] class = [
gini = 0.0 samples = 10 value = [10, 0] class = [
gini = 0.0 samples = 8 value = [0, 8] class = 0
gini = 0.0 samples = 146 value = [146, 0] class = [
gini = 0.0 samples = 1 value = [0, 1] class = 0

2C. How might this information be useful in communicating to doctors performing screenings for cancer based on these images?

These doctors could learn from the Decision Tree model and study how the Worst Perimeter metric predicts breast cancer, knowing that it has a high level of predictive power in this specific model. This would be harder to do with other, less explainable models such as Random Forest or SVM, which have many more important features that they use to make their predictions and would be more difficult for a human to follow or replicate.

3. Varying the Decision Threshold (10%)

3A. For one of the models above that predicts the probabilities to perform the classification (i.e. Naive Bayes, Logistic Regression, Decision Trees, Random Forests), please construct/plot the Receiver Operating Characteristic (ROC) curve to show how well the model performs with across various choices of thresholds.



3B. What is the area under this ROC curve? What does this tell us about our model?

The area is 0.93. This is a very high value and means that this model is very capable of differentiating between benign and malignant tumours.

Extra Credit: What is the optimal choice of threshold for this model to optimize your "goodness" measure in part 1d? How did you find this?

I am interested in maximizing recall more than I am interested in maximizing accuracy. This is because I would rather a patient be misdiagnosed with cancer, than be misdiagnosed to be cancer free. A false cancer diagnosis would certainly be incredibly distressing for a patient, but would likely be resolved shortly after more testing. On the other hand, if the patient actually does have cancer but this is missed by the test, then they might not seek a second opinion and would then not receive treatment, which would be a truly catastrophic outcome.

In the interest of maximizing the True Positive rate, I would therefore set the threshold at the point at which it is maximized and close to 100%, i.e. the point at which the Y value is highest on this graph. This would lead to some False Positives, but I feel that is a tradeoff worth making in this context.