Bipin Sharma

Breast Cancer Detection - Final Report

Problem Statement

One of the main problems with breast cancer (or any cancer for that matter) is that of determining whether the associated tumor is benign (non-cancerous) or malignant (cancerous). Determining this quickly and accurately can be the difference between a patient getting immediate care and waiting till its too late. It is thus important, given a set of features of the cells contained in the tumor, to determine whether a tumor is benign or malignant.

To this end, I have used the readily available Wisconsin Breast Cancer data (diagnostic) from the UCI Machine Learning Repository and created a model with a high degree of accuracy of predicting whether or not a tumor is malignant. The data contained many redundant features, and after feature selection (bringing the number of features down from 33 to 8 independent variables and 1 dependent variable), I was able to create a Gradient Boosting classification model with a prediction accuracy of 0.95. This model can be easily used for predicting the outcome (benign:0 and malignant:1) given the other features of a tumor cell.

Data Wrangling

The raw data from UCI's Machine Learning Repository had quite a number of features. With the fear of collinearity and overfitting in mind, the number of features needed to be reduced to as little as possible without losing important information. This was easy to do once I figured out what each feature was and whether or not they could be eliminated for consideration. For example, we had multiple columns relating to the radius of each cell: the mean radius, the perimeter of the cell, the area of the cell. I started with taking the most relevant of such features (the mean radius in the previous example, as radius will cover the perimeter and the area as well). It did help that the extra features I eliminated were strongly correlated with each other.

All the data types in the features were of the correct type. Hence there were no changes needed on that front. Another basic thing to look at was whether the data had many missing values. On a quick look, the dataset was very clean in that sense - it has no missing values at all.

Exploratory Data Analysis

In my analysis, I wanted to look at all the features and try to see which ones are correlated to the others and to what degree. The easiest way for me to do this was to take a quick look at the “pairplot” feature of seaborn. This is shown below:

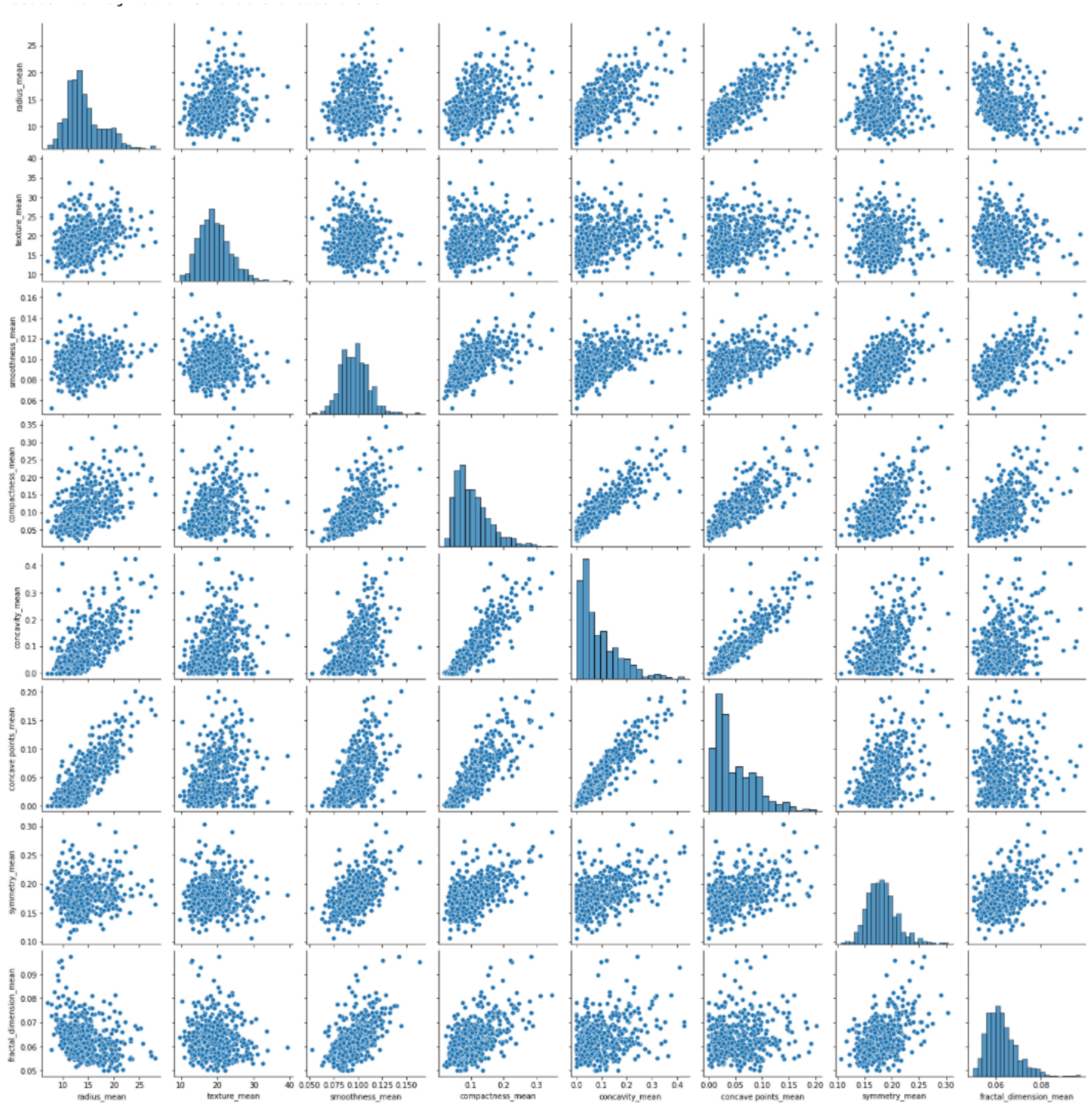
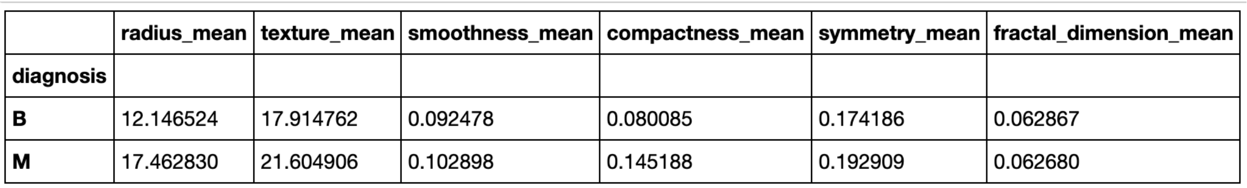


Fig 1. Pairplot showing all the features plotted against each other along with their distribution.

This is a very good way to look at the distributions of all the features, as well as their correlation with each other (demonstrated by the scatter plots). From this plot, it’s clear that a lot of the features that I had selected do not correlate strongly among themselves, which is a great sign! In this particular pair plot, I have left out the dependent variable.

After deciding that the selected features were well distributed (close to normal distribution) and they weren’t prone to overfitting (no collinearity), it was time for further analysis. Do the malignant cells differ fundamentally from the benign cells? Is there a way that we can just look at the cells and maybe get an idea of what kind of a tumor cell it is? With this in mind, I grouped the dataset by the class of the tumor and looked at the means of all the numerical columns.



	radius_mean	texture_mean	smoothness_mean	compactness_mean	symmetry_mean	fractal_dimension_mean
diagnosis						
B	12.146524	17.914762	0.092478	0.080085	0.174186	0.062867
M	17.462830	21.604906	0.102898	0.145188	0.192909	0.062680

Fundamentally, it looks as if the mean radius of the malignant cells is larger than the benign cells. So when we look at the tumor cells, we can tell, judging by the measurement of the radius of the cell from the images, whether the tumor cell *might* be malignant. Taking the example of these radius, if we plotted the radius with respect to the type of tumor, we can see a clear difference:

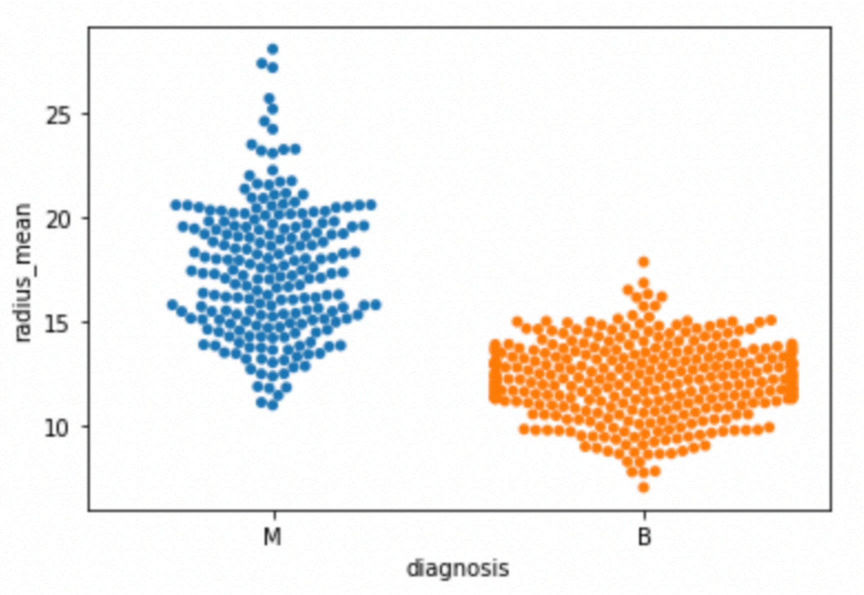


Fig 2. Mean radius vs the tumor type.

Data Preprocessing

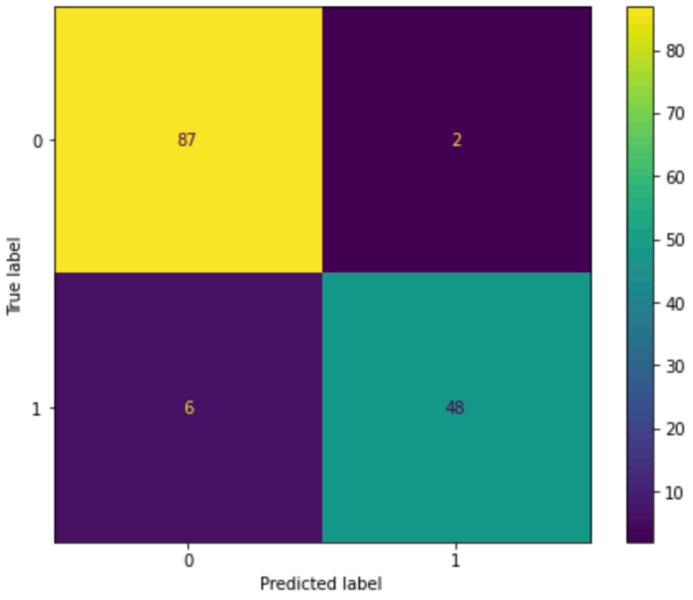
Data pre-processing is a very important step in the model development. I scaled the data so that all the features are on a similar scale, followed by normalizing (standardizing) the data. This got the dataset ready for the next step of splitting into training and testing sets. 25% of the data was set aside for testing and the rest was selected for training our algorithm.

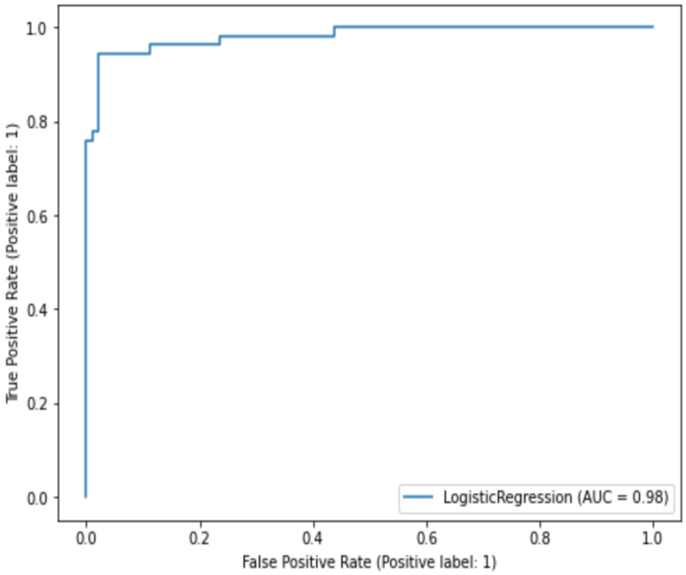
Modeling and Model Selection

For this part, I selected 5 different algorithms, so that I could select the best out of them in the end. I used the following algorithms:

- Logistic Regression
- K-Nearest Neighbors Classification
- Random Forest Classification
- Support Vector Machine (SVM)
- Gradient Boosting Classification

For classification algorithms (especially the binary classification problems), a popular way to evaluate a model is looking at the confusion matrix for the classifier. The metrics that define the accuracy of the model include precision, recall, F-1 score, and ROC-AUC. These are the metrics I used to evaluate my models.

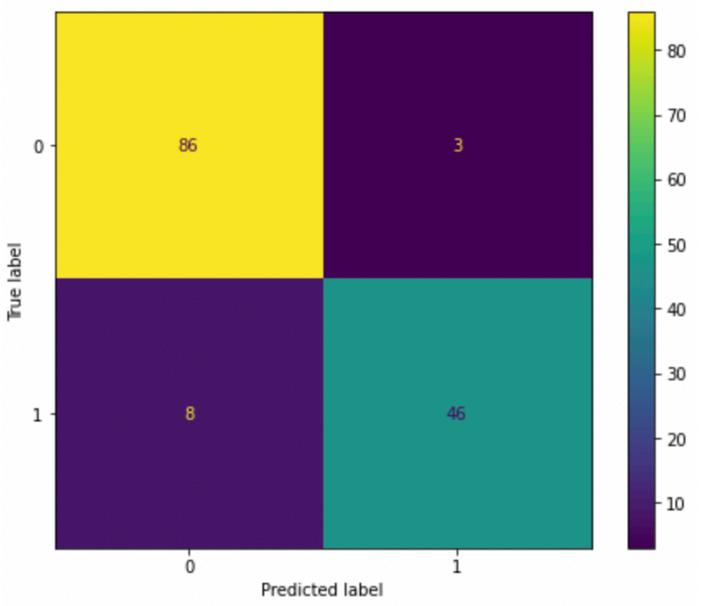




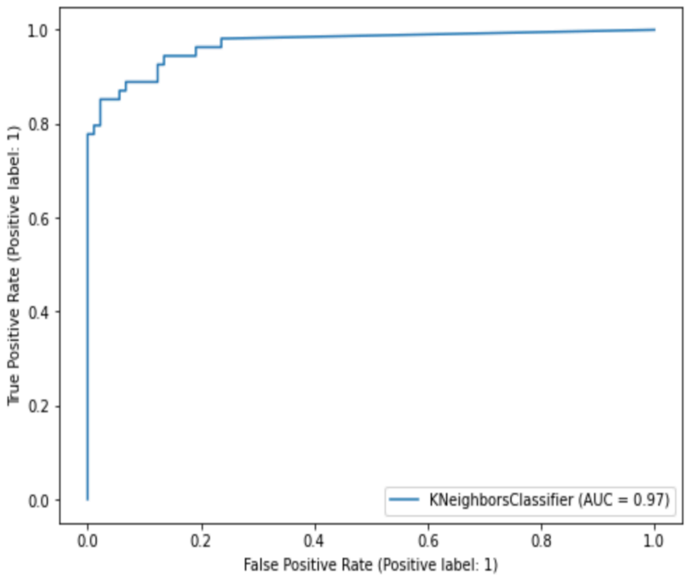
Logistic Regressor confusion matrix

ROC - AUC: **0.98**

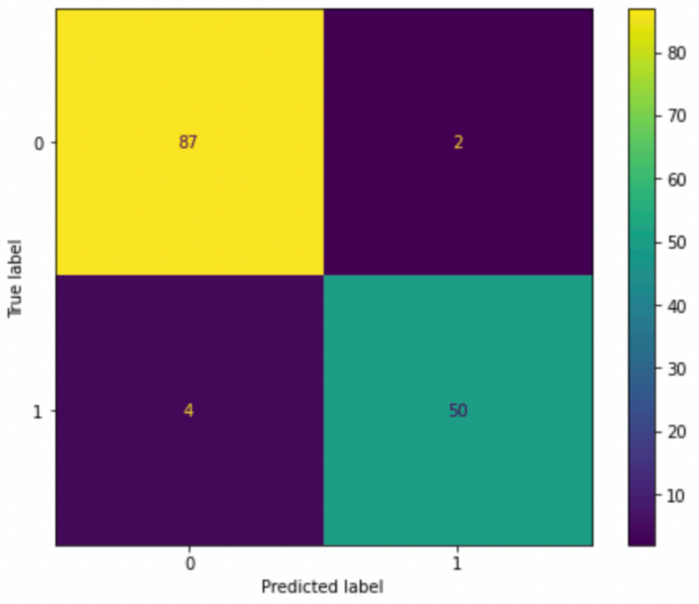
Bipin Sharma



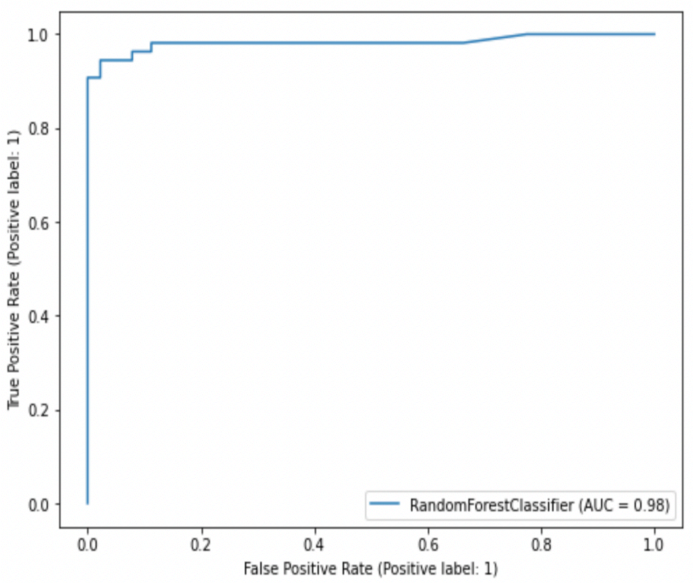
K-Nearest Neighbors Classifier confusion matrix



ROC - AUC: **0.97**

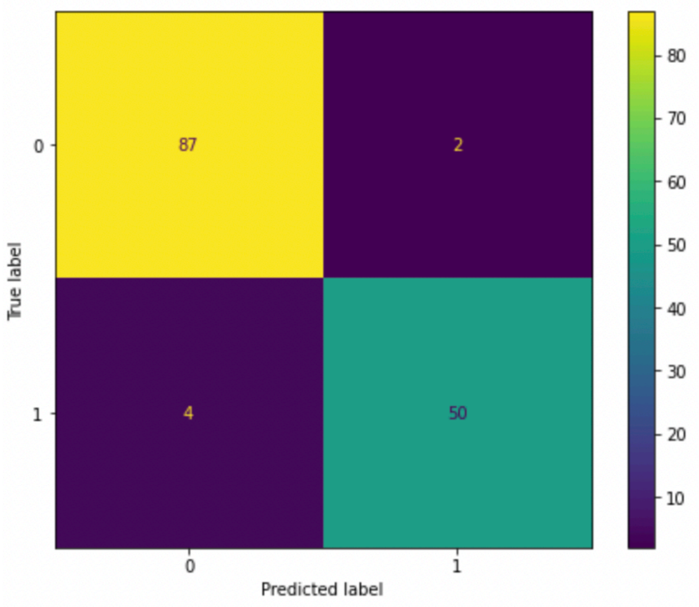


Random Forest Classifier confusion matrix

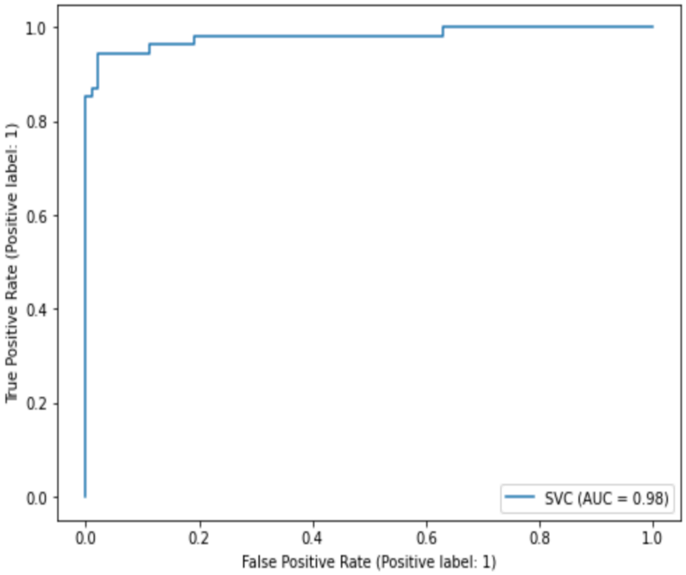


ROC - AUC: **0.98**

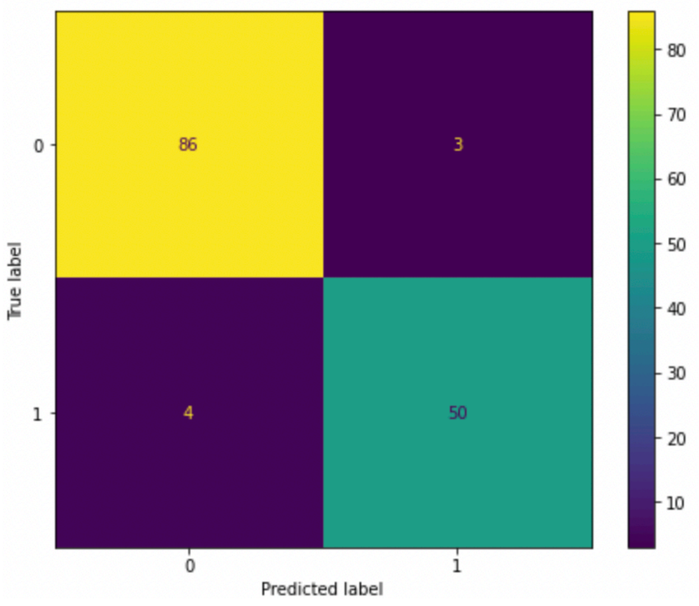
Bipin Sharma



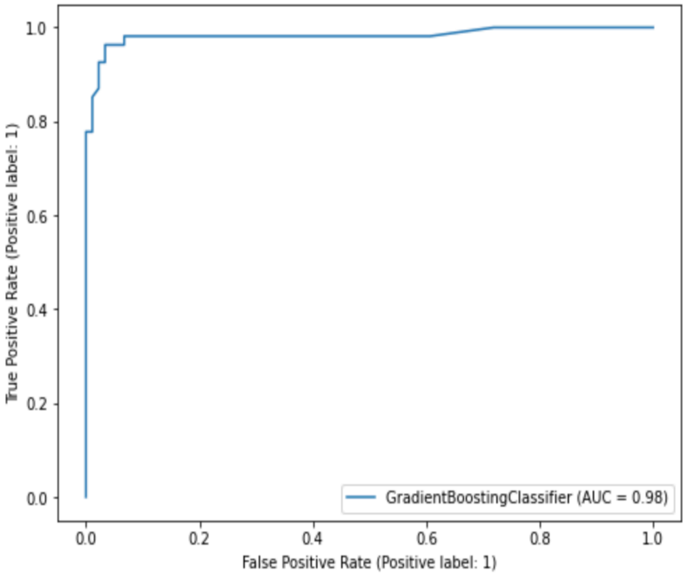
Support Vector Machine confusion matrix



ROC - AUC: **0.98**

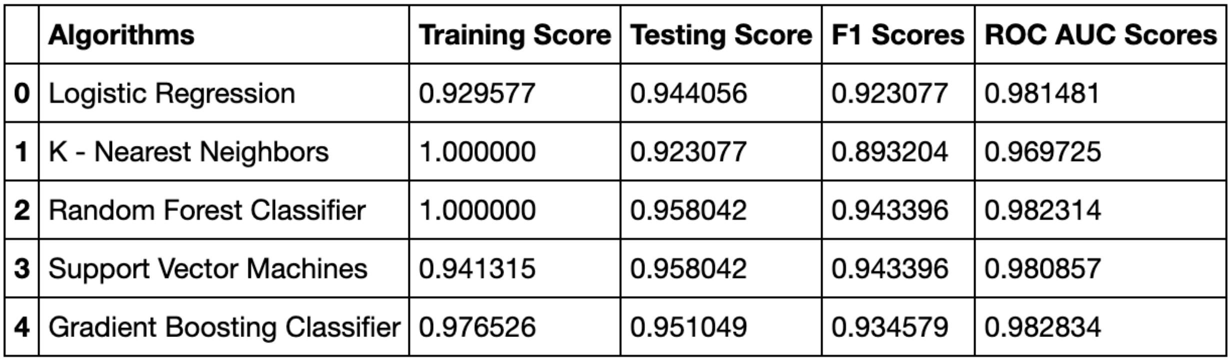


Gradient Boosting Classifier confusion matrix



ROC - AUC: **0.98**

We can see that the Receiver Operating Characteristic area under the curve (ROC-AUC) are very similar for all the models. In addition, the true positives, false negatives, true negatives and false positives are also similar in number for almost all the models. How would we then choose the most appropriate model? An easy thing to do is to take all these above said factors into consideration *along with* the training and testing accuracies. I created a comparison table for this very purpose:



	Algorithms	Training Score	Testing Score	F1 Scores	ROC AUC Scores
0	Logistic Regression	0.929577	0.944056	0.923077	0.981481
1	K - Nearest Neighbors	1.000000	0.923077	0.893204	0.969725
2	Random Forest Classifier	1.000000	0.958042	0.943396	0.982314
3	Support Vector Machines	0.941315	0.958042	0.943396	0.980857
4	Gradient Boosting Classifier	0.976526	0.951049	0.934579	0.982834

Here we go then! Considering all the factors, I think the **Gradient Boosting Classifier** is the best model for our application. I haven't considered a training score of 1.00 to be of as great an importance because it indicates overfitting to the training data.

Key Takeaways

One of the main takeaways for me in this capstone was the fact that we may have many features corresponding to a target variable, but not all the features are always useful. Most of the time, there are redundant variables in the dataset that need to be cast aside. A lot of them are also collinear and thus, do not add much to the model and must be removed as well. We hear a lot about dimensionality reduction and the curse of dimensionality - this was a great introduction to the concept. There will come times where there will be 100's of features - this lesson will be important in feature selection.

The dataset gave some insight as to what changes can we notice from photos of tumor cells to assess them - before doing extensive tests. This model will be very useful in detection of malignant breast cancer cells, given the rest of the features.

Future Research

As always, more the data, the better. If we can somehow gather more information about the malignant or benign tumor cells, we will probably be able to build a better model. I must mention that I mean increasing the *number of data points* as opposed to increasing the number of features. Collection of more data from more patients is very important in spotting some anomalies or some exceptions. In addition, more data lets the model train better, ultimately increasing it’s predicting accuracy, which is our ultimate goal.