MAIS 202 – Project Deliverable 2

Problem statement

Given features from breast tissue, the objective of this project is to predict whether the cancer is benign or malignant using Random Forest algorithm (that will be implemented from scratch if time permits).

Data Preprocessing

The dataset I will be working with is https://www.kaggle.com/uciml/breast-cancer-wisconsin-data. This dataset has 569 samples with 32 features each. The dataset contains a label: diagnosis of breast tissues.

The first step was to understand the data. To do this, I plotted a histogram, 31 violin plots, 31 box plots, a joint plot, 31 swarm plots, and correlation heat map.

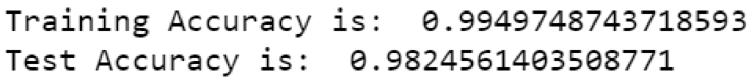
- Histogram: The histogram was plotted to know how many benign and malignant labels are contained in the set.
- 31 violin plots: These plots are interesting because they plot the normalized distribution of each label for each feature. They gave me an idea of the features that can be used to distinguish benign tissues from malignant tissues. To do this, you must observe the mean and standard deviation of two associated plots for one feature. If the mean plus and minus 2 standard deviations do not overlap between malignant and benign, you can infer that they might be good features to use. You can eliminate all others.
- 31 box plots: Same idea as above but a different way of showing the data. In this case, we use the Q1, the Median, Q3 to define a box. Again, we try to minimize overlap.
- Join Plot: A joint plot is just a graph to analyze the correlation of two features. Since it required a lot of work to do it for all features, I decided to use a heat map (after).
- 31 swarm plots: same idea as 2) and 3) but you get an idea of individual values. From these graphs, I created a possible features data frame and removed all “bad” features; which are features that will not allow the distinction between malignant and benign.
- Correlation heat map: I did 3 heat maps with the correlation values: one for the mean values, one for these values and one for the worst values. Then, I made the final decision of which feature to keep, and which feature to drop. Although I am not certain I made the right decision, I decided to keep concave points worst because it was the feature that distinguished the best (from the graphs).

Machine learning model

I decided to use Random Forest from Sklearn (for now) as my machine learning model. I followed the usual 70-30 train-test split. Hyper-parameters were selected by running a for loop and searching for the best combination. I adjusted 2 parameters: n_estimators and min_samples_split). I don’t believe my model is overfitting since I am getting a good accuracy for my test set.

Note: I am setting a random state to fine tune my parameters.

Preliminary results



```
Training Accuracy is:  0.9949748743718593
Test Accuracy is:  0.9824561403508771
```

Figure 1. The accuracy of the machine learning model obtained by Random Forest Classification with n_estimators=8, min_samples_split=2, and random_state=20

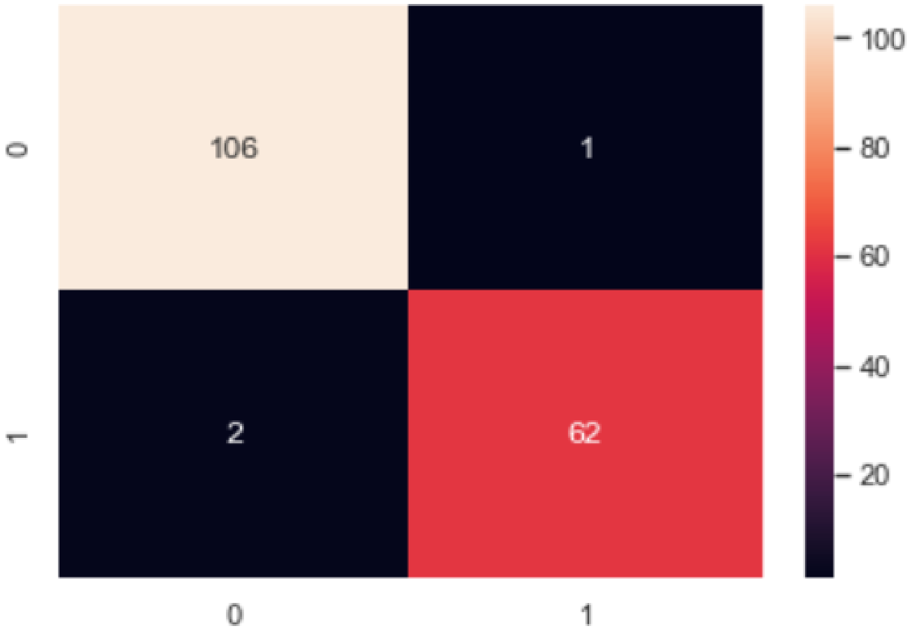


Figure 2. Confusion matrix obtained by Random Forest Classification with n_estimators=8, min_samples_split=2 and random_state=20

Concerning figure 1, we obtained a reasonable accuracy. However, the reader has to note that random_state is fixed to a constant.

Concerning figure 2, the columns represent the true value while the rows correspond to the predicted values by the model. 0 indicates that the tissue is benign and 1 indicates that the tissue is malignant.

With the results shown in the confusion matrix above, we can believe that the problem is solved. However, once I remove the random state parameters in both the train test split and the random forest classifier, the accuracy drops to around 92% and fluctuates much more. Since my project has its use in medicine, it is crucial that it has a precision close to 99.5%. 92% is not acceptable. Further, I would like to find a way to "transfer" false-negatives to false-positive because it is better to say to someone that he is sick when in fact he isn't instead of telling him that he is not sick when he is.

Next steps
Pros: Simple approach to the problem that gives a relatively good result.

Cons: Too simple approach to the problem that is not precise enough to be used in real life. 😊
Future work

First, I am looking forward to receiving your feedback for new ideas to improve my model (either pre-processing strategy, the model itself, the way I selected my parameters)
Second, I believe I made a mistake by not creating a validation set. So, for the next iteration, I will create a validation set and use it accordingly (cross-validation, for instance).
Third, as discussed above, I will try to find a way to transfer the false-negative to false-positive. I am open to suggestions.
Fourth, I will try to implement the random forest manually.
Fifth, I have to figure out how to create a website.