# Binary Classification for Cancer - Group 32

Mukta Jaiswal, Nathan D'Cruz, Dang Ho, Jonathan Yeung

June 6, 2023

Github repository

https://github.com/nathan-dcruz/ECS171-G32

## 1 Group Roles

- Logistic Regression Model: Nathan and Mukta
- Neural Network Model: Jonathan and Dang
- Support Vector Machine Model: Nathan, Jonathan, Dang, and Mukta
- Frontend Development: Dang
- Report: Nathan, Jonathan, Dang, and Mukta
- Poster: Nathan, Jonathan, and Mukta

## 2 Intro & Background

Cancer is a disease in which human cells grow abnormally, spread to other parts of the body, and cause health complications. According to the National Cancer Institute, 2 million people will be diagnosed with cancer and roughly 609,000 will die this year. Furthermore, according to Early Detection of CancerPatient survival is heavily correlated to early detection and identification but some types of cancer can be hard to correctly diagnose. Machine learning can be used to identify the malignancy of tumors and research is ongoing to create models that can identify such cancers. These models have the potential to enable greater automation in diagnosis and play a pivotal role in early cancer detection to save lives.

## 3 Model Decision

We decided on three different models after evaluating and analyzing the binary classification problem for benign or malignant cancers. In our study, a feed-forward neural network, logistic regression, and support vector machine were implemented and compared to gain a better understanding of the models and their accuracies. The main objective of our model decision was to classify tumors accurately using the numerical data we have. For the purpose of analyzing the binary classification of benign or malignant tumors, we decided to utilize these models due to their following distinctive characteristics and advantages.

To begin with, we chose the feed-forward neural network because of its ability to recognize intricate patterns and non-linear relationships in the data. The neural network's capacity to simulate these extensive linkages may enable it to do classification tasks with greater accuracy because tumor classification frequently entails complex interactions between multiple features. Neural networks can also easily adapt to high dimensional data and produce great classification analysis results.

Next, we chose the logistic regression model, a popular and understandable model that is very commonly used for binary classification. When the goal is to classify the data points, in our case classifying between benign and malignant cancer, it is better to use logistic regression. The model offers precise insights into the role played by each feature and may be useful in drawing connections between the features of the tumor and malignancy.

Lastly, we choose the support vector machine model because of its ability to generalize large datasets, making them ideal for binary classification tasks for projects such as ours. The relative simplicity of this model compared to the feed-forward neural network and logistic regression model also means that it should be less prone to overfitting and subsequently provide a more accurate model with fewer parameters to tune and with less training time.

We tested all three models to examine their performance to determine which was more appropriate for classifying tumors. Using this method, we evaluated the benefits and drawbacks of each model and came to a well-informed conclusion.

## 4 Literature Review

Machine learning has become an effective diagnostic and image analysis tool in medicine, especially for the identification of cancer. Recent research has demonstrated encouraging outcomes for binary classification models for predicting malignancy in various forms of cancer.

In the research done by Chao et al. (2014) , the authors' '95.22%' accuracy rate for SVM above '95.1%' accuracy rate for logistic regression exemplifies the value of machine learning methods for classifying cancer kinds. These results are comparable to those we obtained, where even our SVM model outperformed Logistic Regression to provide the best classification results.

Similar to this, according to Matthew (2020), the authors used characteristics including tumor size, shape, and margins, among others, to reach an accuracy of '98.87'% for the SVM model.

These studies show the promise of machine learning methods for raising the success and precision of cancer detection. Machine learning models can help physicians make more precise and quicker diagnosis, which will improve patient outcomes. They do this by utilizing image-based and clinical features.

Overall, these findings suggest that machine learning algorithms are capable of classifying tumors into two categories: benign or malignant. Larger datasets and more sophisticated machine learning models may be used in this field's future study.

# 5 Dataset Description and Analysis

We are using this dataset: https://www.kaggle.com/datasets/erdemtaha/cancer-data.

The dataset includes 570 records of tumors and whether they were classified as benign or malignant (non-cancerous or cancerous). Each record includes up to 30 attributes describing the appearance of the tumor with characteristics such as concavity and smoothness, in addition to specific measurements such as radius, the number of concave points, and symmetry. 63% of the dataset consists of records for benign tumors and 37% for malignant tumors and there is a relatively uniform spread of data for each specific attribute. After using exploratory data analysis and generating a correlation matrix for all the values, we were able to analyze that the variables: radius, perimeter, and area are very closely correlated as shown in the Figure 1 below, thus we decided to drop the area and perimeter columns and only utilized the radius column. This was done to decrease the size of the data set attributes so that it becomes less complex. The column ID was also dropped from our data set as it did not provide any relative information regarding the prediction of the type of cancer.

Here's a breakdown of our exploratory data analysis and data cleaning:

1. Dropped the redundant or unnecessary columns that gave us no relevant information for the prediction of the result (in our case: id column).
2. Generated correlation matrix so that we could either drop or combine the columns that are closely correlated to avoid the problems of encountering multicollinearity issues in the ML models. (in our case: radius, perimeter and area attributes).
3. Generated correlation matrix so that we could either drop or combine the columns that are closely correlated to avoid the problems of encountering multicollinearity issues in the ML models. We normalized that data so that very high and very low data points can be brought between 0 and 1, thus preventing them from creating a margin of error in the prediction model.
4. We binarized the M and B values of the prediction column to 1 and 0 that can be easily be used in our analysis. (in our case: the diagnosis column).

For most parts, our data set did not require a lot of pre-processing and cleaning because most of the columns were not closely correlated or redundant without any null values.

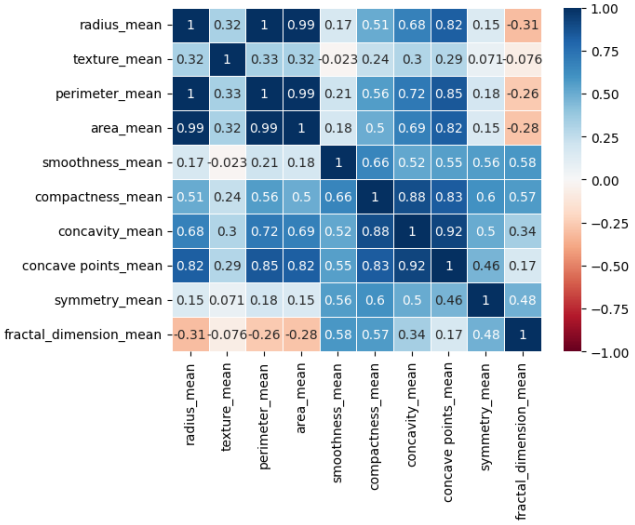


Figure 1: Correlation Matrix

Figure 1 refers to the correlation between attributes of the dataset. If we find some with a high correlation, we only keep one of them for training the model.

# 6 Proposed Methodology

This is a simple binary classification problem. We trained three models, a feed-forward neural network, a logistic regression binary classifier, and a support vector machine. We used K-Fold validation with 10 folds to find the average performance of each model type. We chose not to use the split corresponding to the most performant fold. Instead, we trained our final model on our entire dataset for reasons discussed below.

We chose the best model type based on accuracy, F-Score, and mean squared error metrics as discussed below. We deployed this model with a front end, where the user can enter in numerical data and receive a classification from the trained model. In a real scenario, this information might be passed to the model from a database or a sensor. Since our model used simple numerical data, we required little to no layers of processing between user input and the model.

# 7 Conclusion

We judged our models on three metrics: accuracy, F-Score, and mean-squared error. We chose to consider F-Score because our dataset has slightly more benign samples than malignant. However, in most cases the F-Score was close to the accuracy metric, so we believe that the sample imbalance was not a major factor in this case.

Interestingly enough, at least one fold for nearly every model type produced 100% accuracy. This seems oddly unrealistic and may point to some faults in the dataset or the model overfitting. To avoid using an overfitting model, we chose not to use the parameters corresponding to the most performant model because we did not want them to overfit.

The Support Vector Machine model produced the highest average accuracy and was the least overfitting, so we chose to deploy that model type for our final product.

## 7.1 Feed-Forward Neural Network

Our Neural Network had an average accuracy of 0.8875, an average F-Score of 0.852, and an average MSE of 0.063. The SSE was 6.308 and the training SSE was 11.244. The training variance was -10.12 and the new variance was -5.677. This model was overfitting.

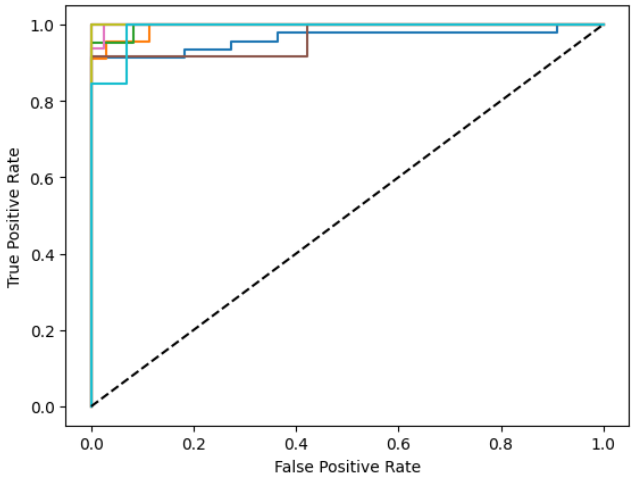


Figure 2: ROC Curve for Feed-Forward Neural Network

Figure 2 refers to the accuracy of 10 folds of Feed-Forward Neural Network.

## 7.2 Logistic Regression

On average, the Logistic Regression model returned an accuracy of 0.9578, a F-Score of 0.9470, and a MSE of 0.0366. The SSE was 3.66 and the training SSE was 4.22. The new variance is -3.290 and the training is variance -3.798. This model was also overfitting.

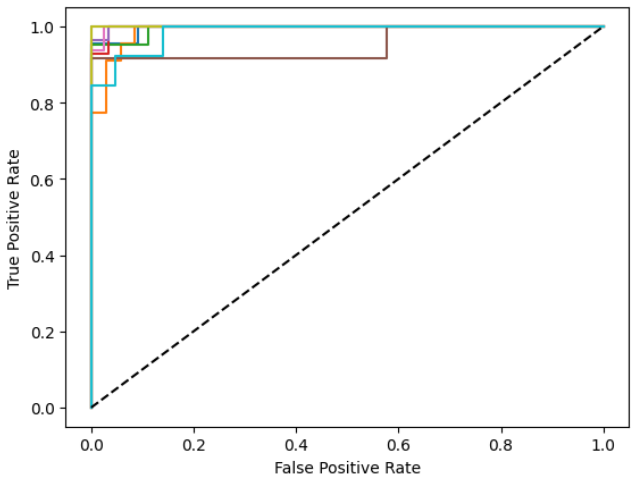


Figure 3: ROC Curve for Logistic Regression

Figure 3 refers to the accuracy of 10 folds of Logistic Regression.

### 7.3 Support Vector Machine

On average, the support vector machine with oversampling returned an accuracy of 0.9720, a F-Score of 0.9600, and a MSE of 0.0280.

Without oversampling, we saw an average accuracy of 0.9648, a F-Score of 0.9490, and an MSE of 0.0328. The training SSE was 3.524 and the testing SSE was 3.275. The training variance was -3.172 and the testing variance was -2.949.

We first tried oversampling without normalization, and this yielded a 0.09 hit to our accuracy score. Notably, this was the only case where we did not observe a 100% accuracy rate for at least one of the 10 folds. After normalizing, we began to see folds with 100% accuracy again.

By normalizing alone, we saw a 1% increase in accuracy and another 0.8% after oversampling with normalization. Overall our support vector machine performed 1.8% better with both these optimizations than without.

We would like to note that using K-Fold validation with oversampling in this case may be problematic. When using oversampling, some of the folds become homogeneous - with only malignant samples, there is no true negative rate and the AUC is undefined. These homogeneous folds may have produced a misleading increase in model accuracy.

Our model's bias and variance statistics indicated overfitting, which prompted us to perform hyperparameter tuning. Based on our implementation of hyperparameter tuning for our SVM model, we found the optimal parameters to use to prevent overfitting were - a linear kernel, with a gamma value of 0.1, and a regularization parameter strength of 10. Unfortunately, due to time constraints we were unable to include the results of our tuning in the final model.

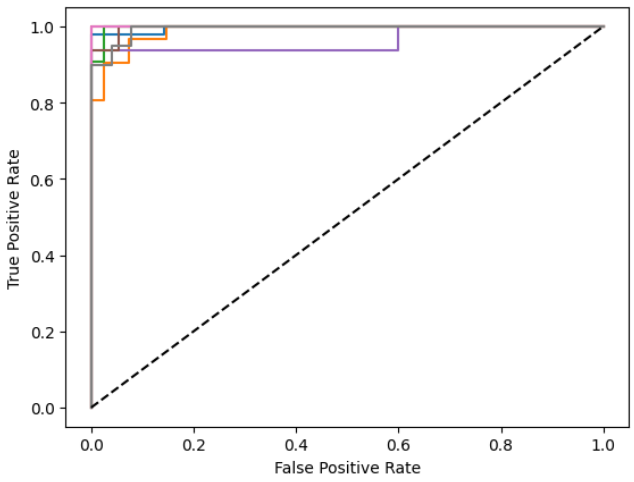


Figure 4: ROC Curve for Support Vector Machine (oversampling)

Figure 4 refers to the accuracy of 10 folds of SVM (oversampling).

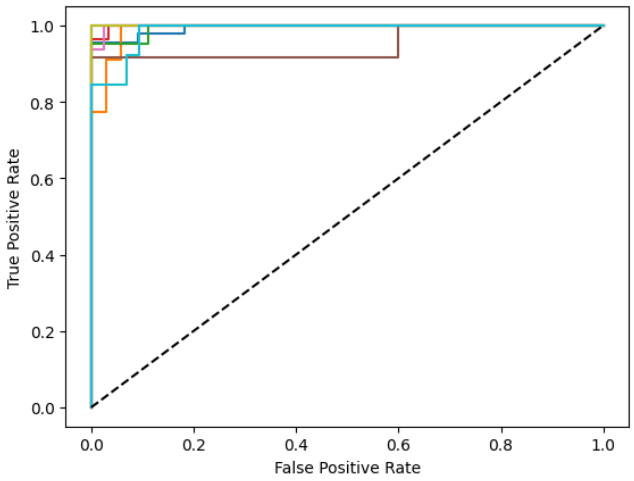


Figure 5: ROC Curve for Support Vector Machine (no oversampling)

Figure 5 refers to the accuracy of 10 folds of SVM (no oversampling).

# 8 Experimental Results

In conclusion, we found the SVM to be the best choice in classifying tumors based on the average accuracy and F-score. Compared to the other models we tested, it also had the lowest difference in bias and variance between the training and testing data.

Although we found that our SVM was slightly overfitting, with a relatively low bias (3.28 vs 3.52) and high variance (-2.95 vs -3.17), we could reduce this by performing hyperparameter tuning using the optimal parameters as discussed in the experimental results section.

We also chose to not oversample our SVM because it appeared to artificially increase the SVM's accuracy. In the future, a more reliable dataset with more samples may be required. Additionally, in the future, we can instead create a model that classifies tumors as malignant or benign based on imagery.

# References

[1] Chao, CM., Yu, YW., Cheng, BW. et al. Construction the Model on the Breast Cancer Survival Analysis Use Support Vector Machine, Logistic Regression and Decision Tree. J Med Syst 38, 106 (2014). https://doi.org/10.1007/s10916-014-0106-1

[2] Mathew, Tina & S, Anil. (2020). A LOGISTIC REGRESSION BASED HYBRID MODEL FOR BREAST CANCER CLASSIFICATION. Indian Journal of Computer Science and Engineering. 11. 899-906. 10.21817/indjcse/2020/v11i6/201106201

[3] Homework & lecture in ECS 171 - Spring 2023