Machine Learning Analysis of Benign and Malignant Tumors

Charlie Condon
University of Tennessee Knoxville
COSC 425
ccondon@vols.utk.edu

Anderson Walsh
University of Tennessee Knoxville
COSC 425
awalsh15@vols.utk.edu

***Abstract*—In this paper, four different machine learning methods were explored for the binary classification of breast tumors as benign or malignant. The Breast Cancer Diagnostic dataset from UC Irvine was utilized, which contains 569 entries. These entries consist of 31 features and a binary label. This dataset was selected due to it being well-suited to classification tasks, in addition to the potential for feature ranking and analysis. The four machine learning methods applied were neural networks, decision trees, linear support vector machines, and logistic regression. Experiments were performed on the dataset, such as applying and evaluating methods with different permutations of oversampling and scaling. An in depth hyperparameter search was performed on the top performing model.**

The results showed that all four algorithms performed well, achieving testing accuracy above 90%. Standardizing or scaling the data generally improved the performance of the models. Feature ranking showed that before scaling, the decision making space was dominated by few features, while after scaling, all of the features had relatively equal importance. Logistic regression was also implemented to perform probabilistic interpretation of the data, showing that as the values of certain features increased, the probability of the tumor being malignant also increased.

Overall, the experimentation demonstrated the effectiveness of the chosen machine learning methods for the binary classification of breast tumors, as well as the importance of scaling and oversampling in improving model performance and generalization.

I. Introduction and Motivation

The Breast Cancer Diagnostic dataset from UC Irvine from Kaggle was utilized for this report. Each entry in the dataset is a different tumor identified as either benign or malignant. This dataset was chosen because binary classification problems naturally lend themselves to machine learning. Further, there is potential to do feature ranking and analysis at low computational cost. Binary classification is the among the most approachable machine learning problems. This has the benefit of allowing many possible approaches to the dataset. Further, model optimization and feature analysis can be the main area of focus, as opposed to potential complexities associated with other problems. Feature ranking and analysis can lead to important conclusions with breast cancer data. Understanding the features that contribute the most to a highly accurate model would suggest a strong correlation between those features and the cause of cancer.

The ML approaches being utilized are (1) Neural Networks (NN), (2) Decision Trees (DT), (3) Linear Support Vector Machines (linear SVM), and (4) Logistic Regression (LR).

1 (NN): Neural Networks can provide a high accuracy models at the cost of explainability. NNs will be optimized by performing a hyperparameter search on the neural architecture, the activation function, the initial learning rate, and the learning rate.

2 (DT): Decisions Trees provide more insight into the importance of features. This allows greater understanding of which factors have a greater impact on tumor type.

3 (Linear SVM): Linear Support Vector Machines are another high accuracy baseline method. This model will be useful for testing the efficacy of scaling and oversampling the data.

4 (LR): Logistic Regression is a probabilistic model that can be used to produce classifications with confidence between the input features and the output. This model will be used to test the relative performance of the other methods and to gain insight into the importance of different features.

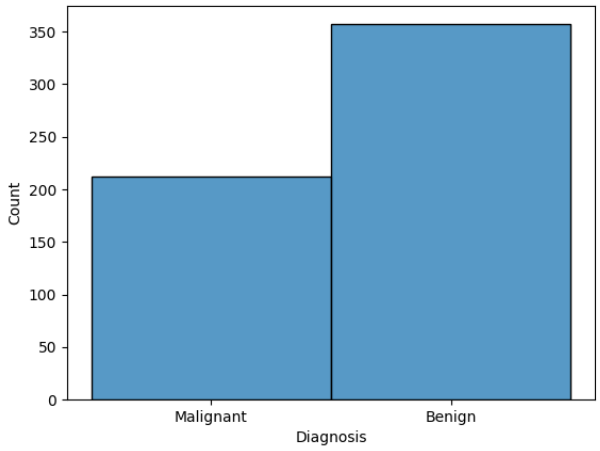
The overarching goal is to optimize various models using the above techniques in order to create the most accurate classifier. Having created a highly accurate classifier, the objective is then to gain a better understanding of which features contribute the most to tumor type. In summation of the technical approach: order to achieve these goals, the above 4 ML methods will be utilized to create different models. The performance of these models will be compared. A hyperparameter search will be conducted on the top performing model. Further, evaluation of the data with and without scaling and oversampling will be performed. Models will be tested under these conditions as well. Success in these goals will be based on creating multiple accurate models for comparison. Success will also include visualizing results and drawing conclusions about feature importance and model generalization.

In this report we will describe the details of the dataset in the Dataset section. This will include diving into its shape and any preprocessing that we did on the data. In the Machine Learning Approaches and Methodology section, we will explore the 4 models we chose and are reasoning for choosing them. We will then run experiments and optimize the models, detailing the results in the Results section. Finally, we will discuss our conclusions based on the results.

II. Dataset

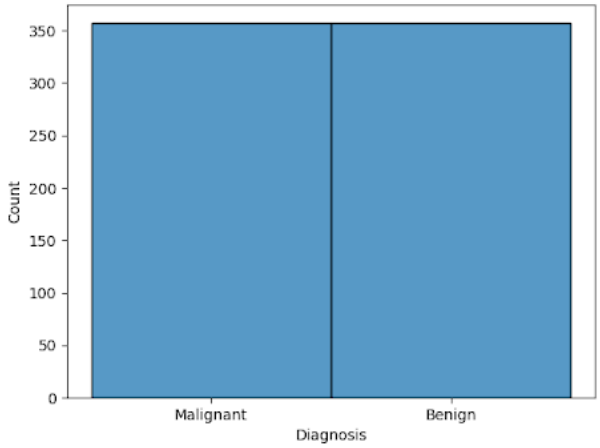
As stated previously, we used the Breast Cancer Diagnostic dataset from UC Irvine, and it can be found on kaggle. Within this dataset, every data entry (row) is a tumor. There are 569 total rows and 32 total columns, including 31 features and 1 label. This label is a binary classifier that determines the tumor type (benign or malignant). In total, about 63% of the entries are classified as benign with the remaining 37% classified as malignant.

Fig. 1. Original Dataset



This uneven distribution of labels could lead to a less generalizable or less accurate model. To address this potential issue, we implemented oversampling of the data in order to bring the split of malignant and benign to 50/50. Since the dataset only has 569 entries, we decided to use oversampling over subsampling to ensure that we're using all possible data. The computational inefficiency associated with oversampling is irrelevant on a dataset of this size.

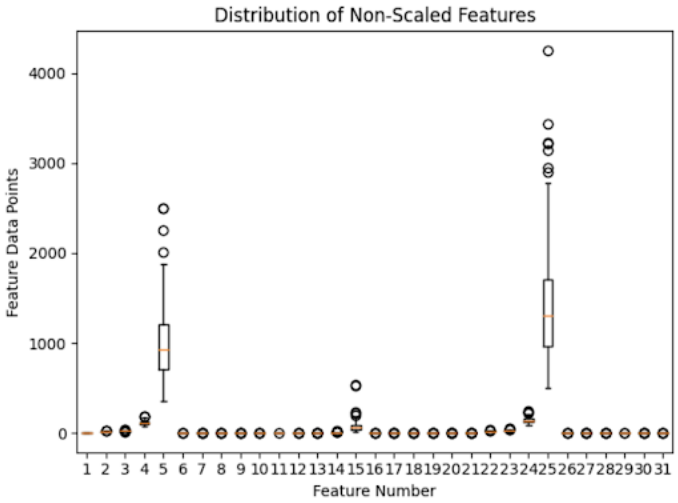
Fig. 2. Post Oversampling Dataset



There are now 357 entries for both malignant and benign tumors. Going forward, we will test both sets of data (pre and post oversampling) and compare the accuracy of the models.

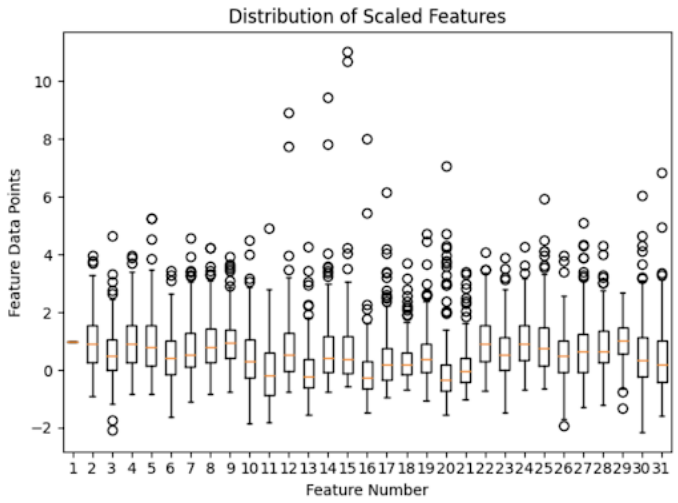
The features consist of 10 physical characteristics with 3 metrics given for each. The features are radius, texture, perimeter, area, smoothness, compactness, concavity, concave points, symmetry, and fractal dimension. The dataset includes the mean, standard error, and the average of largest outliers in means for each of these physical characteristics. Below is a plot of the distribution of the features:

Fig. 3.



In Figure 2, there are a few features that have much higher values than the rest. When applying machine learning methods to this dataset, features with much higher values may be weighted differently. To prevent assigning unnecessary weight, we decided to scale the data using the StandardScaler library. The below plot shows the distribution of the features after scaling:

Fig. 4.



The dataset on kaggle also had one column that contained

only null values. We removed this and did not count it as a feature.

III. Machine Learning Approaches and Methodology

As described in the introduction, we will be using 4 ML approaches on the dataset: Neural Networks (NN), Decision Trees (DT), Linear Support Vector Machines (linear SVM), and Logistic Regression (LR). We chose NNs because they are highly accurate and offer various optimization methods through hyperparameter search. DTs will help with feature ranking. Linear SVMs can help with determining how effective the scaling and oversampling of our dataset was. LR helps with understanding the relationships between the input features and the output. LR also offers probabilistic interpretation of the features.

Part of our experiment will include testing these approaches on the scaled, non-scaled, oversampled, and non-oversampled dataset. Through visualizing the accuracy of the model associated with each of these sets of data, we will be able to determine which combination yields the best result.

We will also compare the accuracy of all 4 models and visualize that comparison. In order to improve the accuracy of each individual model, we will perform a hyperparameter search using K-Fold Cross Validation. K-Fold CV allows us to evaluate many combinations of parameters even with our relatively small dataset size. For NNs, we will search for the best neural architecture, activation function, initial learning rate, and learning rate. For DTs, we will search for the best depth of the tree. For linear SVMs, we will search for the best max number of iterations and C value. We will use the default parameters for LR.

We will be reporting the accuracies associated with these searches and compare the best results across methods. We will also list which combinations of hyperparameters yielded the highest accuracy, and which features were the most important.

We will achieve success when we have performed a parameter search on each model, visualized the accuracies of these models, and visualized the importance of each feature. Then we can perform analysis on the results and reveal our findings.

IV. Results

We first evaluated the models with default parameters on the dataset with no oversampling. We did this both pre and post scaling the data.

All algorithms performed well, achieving above 90% testing accuracy. This performance was evaluated using K-Fold Cross Validation. Standardization or scaling of data generally improved the model's performance. We see an accuracy improvement in every model except DTs. Neural Networks experienced the greatest improvement after scaling the data, but were the second best performing approach against linear SVMs with an accuracy of 97.74%. NNs observed an increase of over 6% accuracy from the pre-scaled data.

We then ran the same evaluation with the oversampled dataset.

Fig. 5.

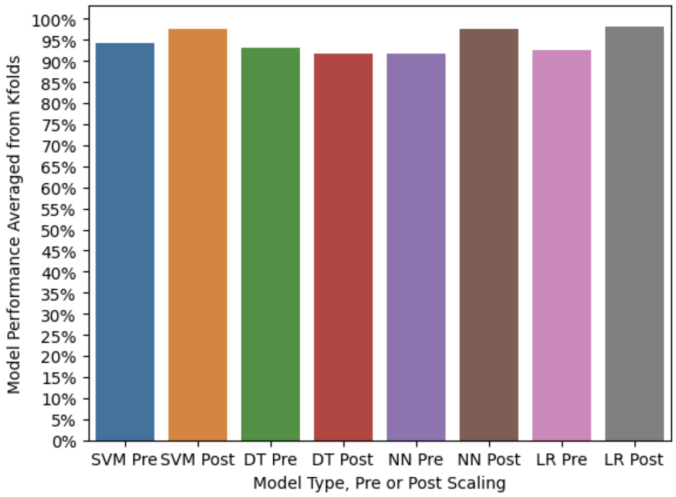


Fig. 6.

Model Type	Pre-Scaling	Post-Scaling
Linear Support Vector Machine	94.21153846153845	97.74358974358975
Decision Tree	93.21153846153847	91.69871794871796
Neural Network	91.69871794871796	97.73076923076923
Logistic Regression	92.70512820512822	98.24358974358974

Fig. 7.

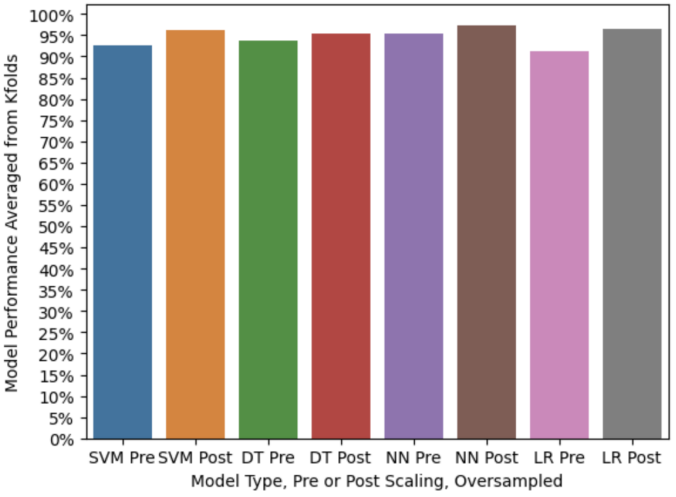


Fig. 8.

Model Type	Pre-Scaling	Post-Scaling
Linear Support Vector Machine	92.58367346938776	96.19591836734693
Decision Tree	93.58775510204082	95.39183673469388
Neural Network	95.39183673469388	97.39999999999999
Logistic Regression	91.19999999999999	96.59183673469387

After oversampling, every model performed better. The NN's accuracy jumped by 2% from 95.39% to 97.39%. This is a large increase considering the already high accuracy of the model, and it ultimately is the highest performing model on the oversampled dataset.

Due to neural networks' high accuracy on this dataset, we decided to try to optimize it further. We searched over various hyperparameters. All subsequent experimentation takes place utilizing the oversampled dataset, as it saw a universal performance improvement.

Fig. 9.

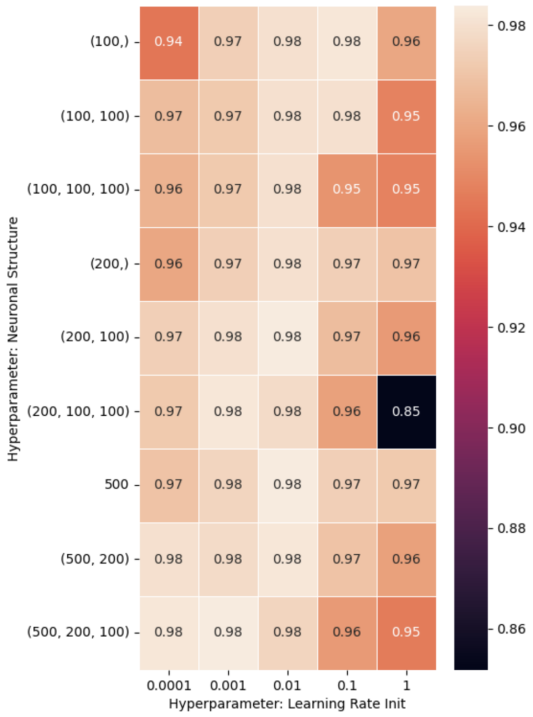


Figure 9 shows the heatmap plot of the neural structure and the initial learning rate. Here, we saw the highest accuracy with an initial learning rate of 0.01 and a neural architecture of (200,100).

Our highest accuracy yielding combination of hyperparameters included a neural architecture of (200,100), an initial learning rate of 0.01, a constant learning rate, and using the RELU activation function. The training accuracy of a NN with these parameters was 100%, and the testing accuracy was 97.902%.

We then used this optimized model to do feature ranking. The SKLearn Python library includes a permutation_importance() function which allows for ranking of the features in a neural network model. We used this function to determine the importance of each feature pre and post scaling.

Figure 10 shows the heatmap plot of learning rate and the activation function accuracies. The learning rate did not have an impact on the accuracy of the model. The RELU activation function yielded the highest accuracy.

Fig. 10.

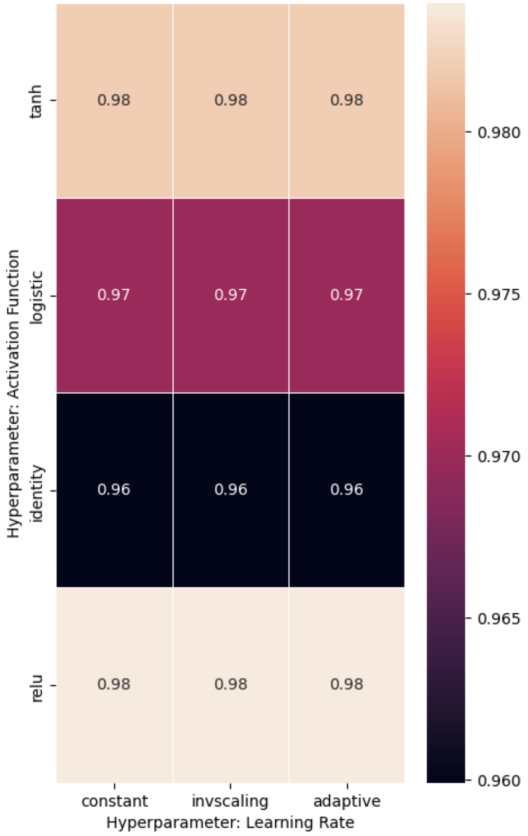


Fig. 11.

Feature	Importance
area_worst	0.3870052539404554
area_mean	0.38287215411558667
perimeter_worst	0.08577933450087569
area_se	0.039124343257443 11
perimeter_mean	0.036672504378283746
texture_worst	0.015586690017513167
concavity_worst	0.010507880910683031
compactness_worst	0.009001751313485 14
radius_mean	0.007075306479859922
concavity_mean	0.005288966725043822
radius_worst	0.004378283712784623
concave points_worst	0.00343257443082315
concave points_mean	0.002977232924693 5446
perimeter_se	0.0023467600700525625
texture_se	0.0014010507880910805
symmetry_worst	0.0010507880910682955
compactness_mean	0.0008406304728546 31
texture_mean	0.00010507880910684664
symmetry_mean	0.0
smoothness_mean	0.0
fractal_dimension_worst	0.0
fractal_dimension_mean	0.0
radius_se	0.0
compactness_se	0.0
concavity_se	0.0
concave points_se	0.0
symmetry_se	0.0
fractal_dimension_se	0.0
smoothness_worst	0.0
smoothness_se	0.0

Before scaling, decision making is dominated by the area_worst feature with 0.387 importance. The next most significant feature is area_mean at 0.3828 performance. Many of the features visualized had no impact on model learning at all, as denoted by an importance of 0.0.

Fig. 12.

Feature	Importance
radius_se	0.031792717086834744
texture_worst	0.029075630252100845
concave points_worst	0.025154061624649886
concavity_worst	0.018711484593837574
concave points_mean	0.018011204481792764
radius_worst	0.014705882352941204
compactness_se	0.012997198879551844
concavity_mean	0.012801120448179288
area_worst	0.012717086834733899
perimeter_worst	0.011624649859943985
fractal_dimension_se	0.010644257703081236
fractal_dimension_worst	0.010196078431372544
area_se	0.009663865546218476
smoothness_worst	0.007310924369747878
texture_mean	0.006666666666666657
texture_se	0.006078431372548998
symmetry_worst	0.005546218487394938
compactness_mean	0.0049299719887955
compactness_worst	0.0044537815126050265
perimeter_se	0.0033333333333333214
concave points_se	0.0030532212885153952
symmetry_se	0.002633053221288506
smoothness_se	0.001764705882352935
fractal_dimension_mean	0.0010084033613445343
symmetry_mean	0.0002521008403361336
area_mean	0.00019607843137254833
radius_mean	2.8011204481792616e-05
perimeter_mean	-5.602240896358523e-05
concavity_se	-0.00011204481792717046
smoothness_mean	-0.0002801120448179262

We then ran the permutation_importance() function on the scaled dataset, which was the one with the higher initial accuracy. After scaling, every feature's importance is much closer in value. There is a relatively even distribution of the weights. However, some features actually began negatively affecting learning. This naturally led us to feature pruning.

We were indeed able to use those feature rankings to inform feature pruning, through a wrapper to sklearn's permutation_importance tool for feature ranking, provided by the feature selection toolset. 15 features were pruned in this way. Those that remained are visualized in Figure 13.

Fig. 13.

Feature	Importance
area_se	0.07271453590192645
area_mean	0.054640980735551674
concave points_mean	0.03786339754816112
symmetry_mean	0.02837127845884414
perimeter_se	0.02672504378283712
fractal_dimension_mean	0.026234676007005263
compactness_mean	0.022802101576182144
radius_mean	0.020805604203152354
perimeter_mean	0.020455341506129588
texture_se	0.017618213660245182
concavity_mean	0.017057793345008745
texture_mean	0.010963222416812603
smoothness_mean	0.010507880910683007
smoothness_se	0.010017513134851132
radius_se	0.009106830122591942

While the feature selection algorithm is a bit of a black box, we can infer that these features were pruned for several reasons. Likely due to the previously discovered direct negative impact on learning, indirect negative impact on learning due to feature correlation, and in general the feature selection algorithm's bias towards creating more generalized models. Half of all the data's features were pruned. This can be observed in Figures 14 and 15. Figure 14 is a correlation matrix of our initial scaled dataset, and Figure 15 is that of the post-pruning dataset. These visuals illustrate that not only is the model much simpler, but in general has fewer hotspots of strong feature correlation.

Fig. 14.

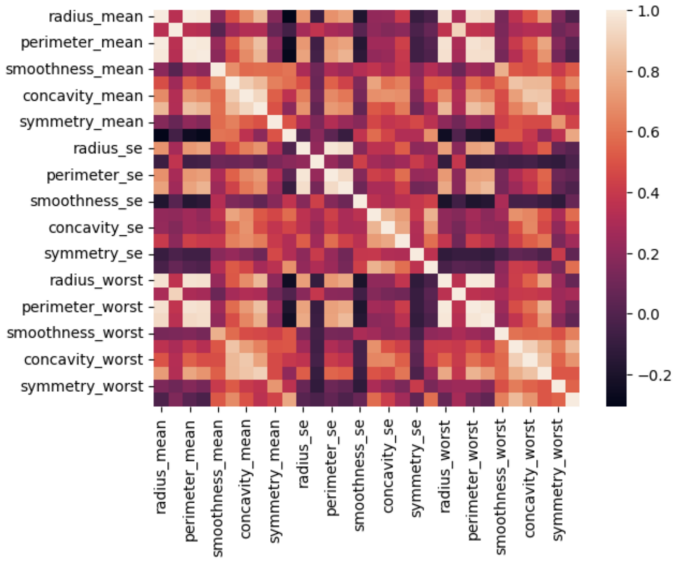
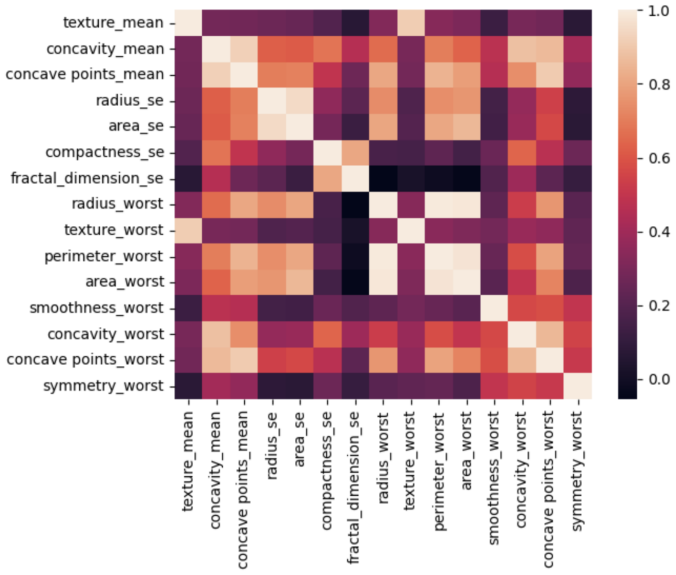


Fig. 15.



We also used logistic regression to perform probabilistic interpretation of the data. We hoped to visualize trends in features as a probabilistic model progressed towards 100% certainty of a malignant classification. This could identify early warning signs in tumors, or possibly even indicate the stage of cancer. The algorithm used to accomplish this was hand-written. It groups entries in the data by their malignant classification probability to the nearest percentile in multiples of 5, then for each feature makes a plot of the average feature

value at each level of confidence. To visualize this, we made a plot for each feature that showed the average value of that feature and the percentage chance of the entries at that value being malignant. The chance of malignancy was calculated using logistic regression. We did this with the oversampled data due to its increased performance. We also ran the test on both the scaled and unscaled data. Many of the visualizations produced for each feature did not have a clear trend. The non-scaled data is separated into 3 plots so the value change of the individual features are visible. Out of all the features, area_se, radius_worst, area_worst, and area_mean showed the most discernible trends. However, they along with a few others in Figure 19 showed similar trends. So much so that, the first 3 of these features were actually pruned by feature selection, indicating they're likely positively correlated. Nonetheless, on average, increases in these values seem to be a good indicator for risk factor in tumor malignancy, and may have some relationship with the stage of a tumor.

Fig. 16.

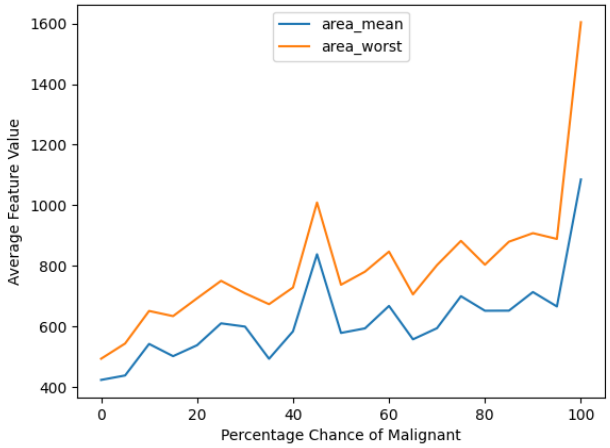


Fig. 17.

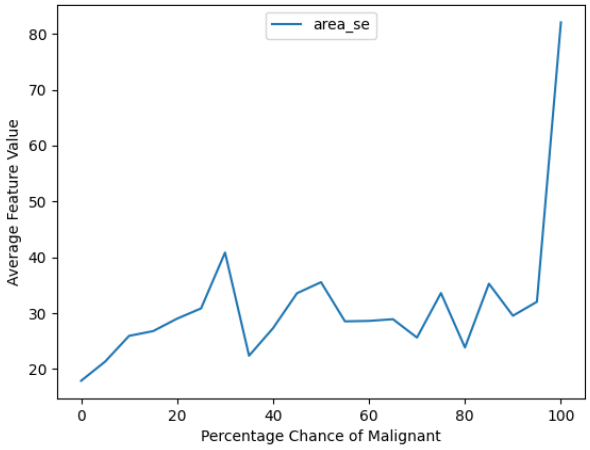


Fig. 18.

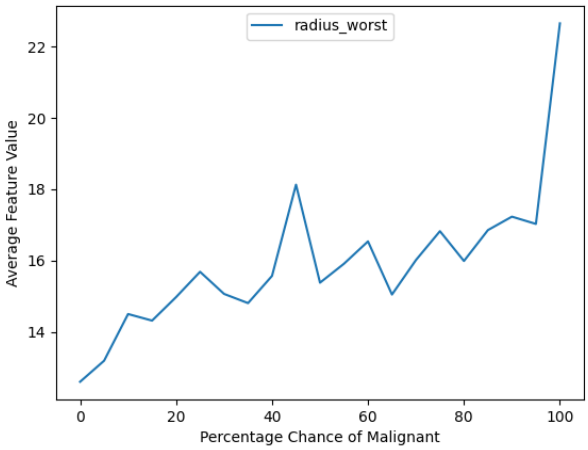
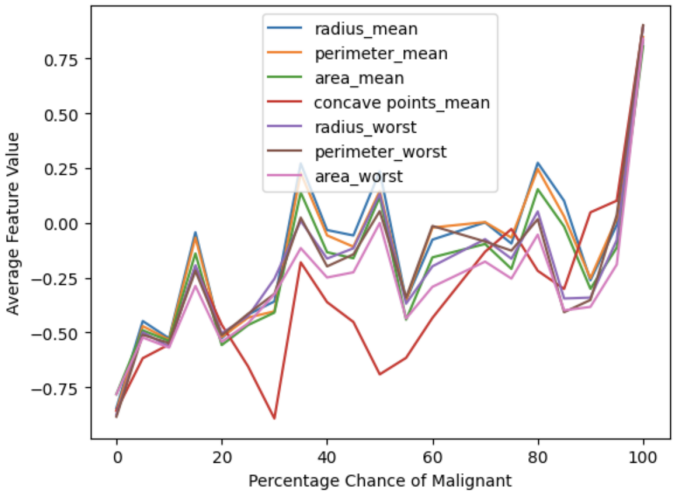


Fig. 19.



Figures 16-19 show the trends of area_se, radius_worst, area_worst, and area_mean among others. Generally, as their values increase, so does the probability of the tumor being malignant.

V. Conclusions

The major conclusions involve the performance of the different algorithms, the impact of scaling on the models, and the impact of oversampling on the models.

In terms of the algorithms, all of them performed well, achieving testing accuracy above 90%. This suggests that the dataset was well-suited for the methods used and that the algorithms were able to effectively learn from the data.

Scaling the data had a large impact on the model's performance. Standardizing or scaling the data generally improved the accuracy of the models, with the exception of decision trees. DTs are not impacted by variance in the data, so they don't need the features to be scaled. The improvement in the other models is likely because scaling the data can help to

reduce the effects of having vastly different feature values. Neural networks saw the largest improvement from scaling the data.

Oversampling led to an increase in accuracy for DT, NN, and LR. By balancing the dataset, the model was less likely to overfit to the benign class. This also increased the size of the dataset, and, importantly, did not throw out any data as we would have seen with subsampling.

Another key finding was the high performance of neural networks, which achieved the highest accuracy of all models with default parameters. The optimization of the neural network showed that the best combination of hyperparameters included a neural architecture of (200,100), an initial learning rate of 0.01, a constant learning rate, and using the RELU activation function. Careful tuning of the hyperparameters can further improve the performance of neural networks.

Logistic regression was used to perform probabilistic interpretation of the data, showing that as the values of certain features increased, the probability of the tumor being malignant also increased. This has potential implications for indicating the severity of tumors, or the likelihood of their malignancy.

With more time, we would dive deeper into feature pruning and feature ranking. This would allow us to further tune the dataset. We could also perform a more exhaustive hyperparameter search for each model. This would allow us to gain a better understanding as to what causes a tumor to be malignant, and would allow us to create more accurate models for predicting tumor type. We would also like to obtain more data to evaluate relationships between tumor characteristics and disease. Specifically we're interested in data with more labels regarding type and severity of disease. We're considering the possibility of applying an unsupervised method to generate our own labels, such as clustering, given the fact that feature pruning cut the dimensionality of our data in half. But intelligently creating labels from cluster groupings would be difficult.

VI. Contributions of Team Members

Charlie and Andy collaborated on the entire project. We both contributed to the presentation slides, writing the code, and writing the report. Andy's focus went more into the probabilistic interpretation of data, in addition to feature pruning. Charlie's focus was on dataset manipulation and model evaluation.