Understanding Breast Cancer Diagnosis and Prognosis via Linear Programming

Palmy Klangsathorn and Angelina (Lenn) Kong

June 5, 2025

1 Introduction

Breast cancer is a common type of cancer that can affect anyone. The source of breast cancer starts in the breast tissue from an uncontrolled growth of abnormal cells and can result in the formation of tumors and the potentially spread throughout the human body. Many scenario's can go into play when cancer is involved such as a consequence in "false-positive screening test results can lead to fear, anxiety, and frustration with the health care system" (Winstead, 2024). So it is crucial to find ways to support a patient in a timely and accurate manner.

One approach that has presented helpful success in limiting threats from breast cancer is the use of linear programming. Linear programming (LP) is a mathematical method that can be used to support a diagnosis and prognosis of a patient. LP arose from its connection to analyzing breast cancer data from Fine Needle Aspirates (FNAs) to get early detection of breast cancer and avoid unnecessary surgery. By minimizing diagnostic errors and catching potential recurrence, LP can save time and financial burden on a patient. From this, LP can improves the quality and efficiency of patient care. This paper will dive deeper into how LP is being applied to support and improve the diagnosis and prognosis of breast cancer.

2 Approach and Methods

Our goal is to use three methodologies in order to make classifications with breast cancer to be malignant or benign, using L1-SVM and MSM/MSM-T. Additionally, we will categorize and make a prediction of the breast cancer: recurrence or non-recurrence. This will be done with RSA.

2.1 Dataset

We utilized the Wisconsin Breast Cancer Diagnostic and Prognostic dataset. These were read into Julia using a custom function, `read_to_dict`, which parsed the raw data into dictionaries for easy manipulation. We then converted the dictionaries into `DataFrame` objects for structured data analysis. In these datasets, 569 samples were taken, each representing 30 numerical features that described characteristics of cell nuclei, and since some features might have bigger values than others, we normalized the data. For each sample taken, one was classified and the diagnosis labels were converted to numerical values using the `ifelse` function to treat the classification task as a numerical optimization problem. We also carefully handled missing values and ensured consistency across all features. To see how well our models perform on new data, we split the dataset into two parts: one for training and one for testing. A common split like 70% training and 30% testing was used. This helps avoid overfitting and gives us a better idea of how the model will do in real situations.

2.2 Linear Support Vector Machine (SVM)

Outside of the paper, we will consider using another method, Linear Support Vector Machine (SVM) in which we will classify whether a tumor is malignant or benign and whether the tumor will recur or not. SVMs, especially when using the 1-norm or ∞-norm, can be formulated as linear programming problems. In these cases, finding the best decision boundary, a hyperplane, becomes a matter of solving a linear program.

2.3 Multi Surface Method (MSM)

We will consider the Multisurface Method (MSM) and the Multisurface Method Tree (MSM-T) to classify whether a tumor is malignant or benign. The MSM uses an LP model that constructs multiple series of separating planes in a space that divides the data into being either malignant or benign. To do this, let any two points be linearly separable; a plane will be placed between them. Each plane is found by solving a LP problem where the objective is to create the largest distance between the plane and the data points. If there is a case where the two points are not linearly separable, the MSM-T is introduced. MSM-T can construct a plane that will minimize the average distance of misclassified points to the plane, which reduces the number of misclassified points. Due to this, the MSM-T will classify breast cancer samples into being malignant or benign accurately from its use of a decision tree to classify the new points. Thus, MSM and MSM-T produces an interpretable and efficient classification of malignant and benign tumors.

2.4 Using Linear Programming for Tumor Classification (Malignant or Benign)

2.4.1 L1-regularized Support Vector Machine (L1-SVM)

The L1-SVM is a type of Support Vector Machine that we solved using linear programming. It tries to draw a line (or hyperplane) that best separates the two types of tumors. It also uses something called L1 regularization, which helps pick out only the most important features by setting some feature weights to zero. This makes the model simpler and often more accurate.

2.4.2 Multi-Surface Method (MSM)

We used the Julia package `JuMP` with the `HiGHS` solver to build our model. For each separating plane, the function `solve_msm_plane_lp` takes a subset of training data (`X_subset` for features and `y_subset` for labels) and solves a linear program.

The objective of this linear program is to:

- Maximize the margin ρ between the hyperplane and the closest points,
- While minimizing the total slack (classification error), controlled by a regularization constant C_{msm}.

We included L1-norm regularization in the optimization to reduce overfitting. Since L1-norm is not linear, we added auxiliary variables $\mathbf{t}$ to represent the absolute values of $\mathbf{w}$ and used linear constraints to enforce this. The constraint $\sum_j t_j \leq 1$ ensures that the solution remains sparse, focusing on the most important features.Each hyperplane is found iteratively by applying this method to subsets of the data. MSM-T was used to organize multiple planes in a tree-like format to handle more complex patterns, enabling us to construct a piecewise-linear boundary to separate malignant from benign cases more flexibly than a single plane would.

2.5 Recurrence Surface Approximation (RSA)

the Recurrence Surface Approximation (RSA) technique will also build on linear programming to classify and estimate a patient's risk of breast cancer occurring once more. Recurrence Surface Approximation (RSA) can be used to find a prognosis and predicting the likelihood and timing of recurrence. A patient can be classified as being recurrent if the disease has already been observed, but knowing if a patient is not recurrent can be a struggle, as it can not be observed in the future. To address this concern, RSA can be used to better understand what is known, Time to Recur (TTR) and the Disease Free Survival time (DFS).

RSA uses LP to determine a linear combination of the input features that will predict the Time to Recur where the Disease Free Survival time is the lower bound in the LP, and the aim is the minimize the difference between the time a recurrence is detectable and the time the recurrence was actually detected. That is we enforce the constraint:

$$s(\mathbf{x}_i) \geq \text{DFS}_i$$

for each patient i. This avoids underestimating the time a patient remained disease-free.

This linear program aims to determine a recurrence surface,

$$s(\mathbf{x}) = \mathbf{w}\mathbf{x} + \gamma$$

, where the linear program can learn more about the weight vector, $\mathbf{w}$, and the constant term, γ. Additionally, the function will have the vector of measured features, x, regarding the cell nucleus and size of the tumor, and the surface that fits the observed recurrence times, s. However, this is a linear function, so this function will be implemented in the format of a linear program. From here, a linear program can be constructed:

$$\min_{\mathbf{w},\gamma,\mathbf{v},\mathbf{y},\mathbf{z}} \quad \frac{1}{m}\mathbf{e}^T\mathbf{y} + \frac{1}{k}\mathbf{e}^T\mathbf{z} + \frac{\delta}{m}\mathbf{e}^T\mathbf{v}$$

$$\begin{aligned} &\textbf{subject to} \\ -\mathbf{v} &\leq -M\mathbf{w} + \gamma\mathbf{e} - \mathbf{t} \qquad\qquad \leq \mathbf{y} \\ -N\mathbf{w} - \gamma\mathbf{e} + \mathbf{r} &\leq \mathbf{z} \\ \mathbf{v} &\geq \mathbf{0} \\ \mathbf{y} &\geq \mathbf{0} \\ \mathbf{z} &\geq \mathbf{0} \end{aligned}$$

In the linear program,

$M \in \mathbf{R}^{m\times n}$: The m recurrent points with recurrence times t

$N \in \mathbf{R}^{k\times n}$: The k is non-recurrent points and their last known disease-free survival times in r

y is a vector representing the error for recurrent points

Z is a vector representing the non-recurrent points

V is a term that forces underestimated recurrent points to be closer to the surface

E is a vector of 1's of appropriate dimension

Additionally, it is important to note that when predicting the Time to Recur, if the Time to Recur is shorter than the Disease Free Survival, there will be an error alongside overestimating the Time to Recur of recurrences.

When working with RSA, the first step involves setting aside a tuning set that is one tenth of the training cases. Initially the RSA LP is solved using all the input features, and the resulting recurrence surface is evaluated using the tuning set. To get the best generalization, features are performed through an iterative process, features are removed one by one by setting the weight vector, w to zero. After each feature is removed, the model is re-evaluated on the tuning set until only one feature is left. Once RSA is done, the procedure is evaluated for accuracy and predicting future outcomes by fitting the RSA procedure into a cross-validation framework.

2.6 Using Linear Programming for Predicting Cancer Recurrence

2.6.1 Recurrence Surface Approximation (RSA)

We started by using the same custom data-reading function to parse the file into a dictionary, and then constructed a `DataFrame` from that dictionary. Each record includes a recurrence label ("N" for non-recurrence, "R" for recurrence), time to recurrence (in months), and 30 numerical features describing the tumor.

We used the `JuMP` modeling language in Julia with the `HiGHS` solver to build and solve this linear program. Like our other models, RSA also benefits from L1 regularization to encourage sparsity. We used auxiliary variables and constraints to linearize the L1-norm, which limits the sum of the absolute values of the weights $\mathbf{w}$.

3 Results and Conclusion

3.1 Tumor Classification (Malignant or Benign)

3.1.1 Confusion Matrices

The following matrics were computed for the performance of the L1-SVM (Linear Programming Support Vector Machine) and the Multi-Sphere Model (MSM) model.

Metric	**L1-SVM**	**MSM**
True Positives (TP)	40	23
True Negatives (TN)	73	48
False Positives (FP)	0	19
False Negatives (FN)	1	24
Accuracy	99.12%	62.28%
Precision	100.0%	48.94%
Recall (Sensitivity)	97.56%	54.76%
F1-Score	98.77%	51.69%
Specificity	100.0%	66.67%

Table 1: Confusion Metrics of L1-SVM and MSM on Test Data

Table 1 shows a big difference in how well the L1-SVM and Multi-Sphere Model (MSM) performed on the breast cancer diagnosis task.

The **L1-SVM model** reached a test accuracy of 99.12%, with perfect precision (100.0%) and specificity (100.0%). This means that every time it said a case was cancerous, it was right, and didn't mistake any healthy cases for cancer. Its recall was 97.56%, so it caught almost all cancer cases, missing just one. The F1-Score of 98.77% shows that it was both accurate and consistent. The L1-SVM is a reliable model when avoiding wrong cancer predictions and catching almost all real ones.

The **Multi-Sphere Model (MSM)**'s accuracy was only 62.28%, and its precision was 48.94%, meaning that over half of its cancer predictions were wrong (19 false positives). Its recall was 54.76%, meaning it missed many real cancer cases (24 false negatives). The F1-Score was 51.69%, showing weaker overall performance. It correctly identified about two-thirds of the healthy cases, but still called 19 of them cancer by mistake.

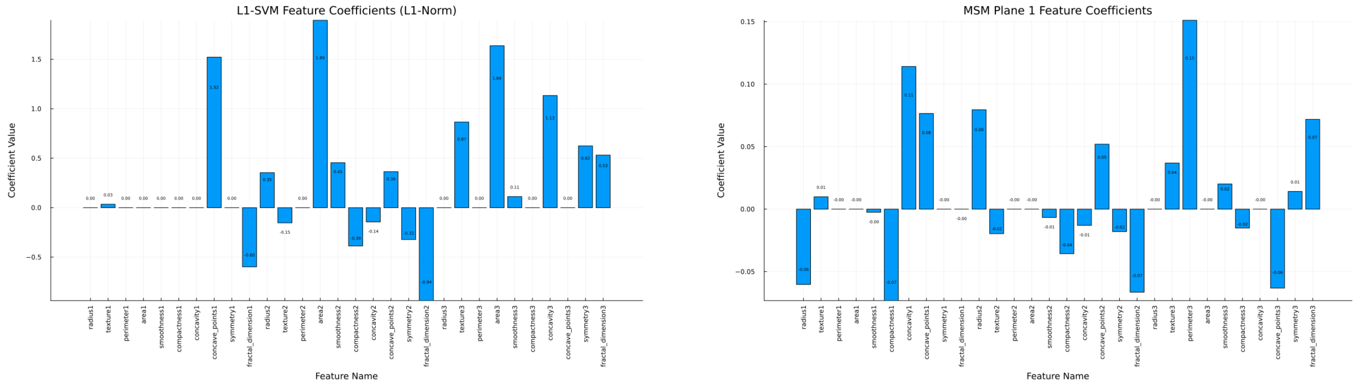
The big gap in performance likely comes from how the models work. The L1-SVM uses a method that picks out the most useful features and ignores the rest. Thus making better decisions and avoid being confused. Meanwhile, the MSM tries to separate data using spheres, which may not work well with complex or overlapping patterns like in this dataset. Because of that, it struggled to tell cancerous and healthy cases apart.

3.1.2 Coefficients Plots

Coefficient plots display coefficients assigned to each feature by their respective models and indicate the importance and direction of influence of each feature in the model's decision-making process.

For MSM Plane 1, it seems to give significant positive weight to `radius3`, `concavity1`, and `concave_points1`. It also places some negative weight on `radius1` and `compactness1`. The coefficients are generally smaller in magnitude compared to L1-SVM.

For L1-SVM, it exhibits much larger coefficients for certain features, particularly `smoothness2`, `concave_points1`, and `area3`. This indicates a more aggressive weighting, which is characteristic of L1-SVM (Lasso) which drives less important feature coefficients to zero. We can see many features have coefficients of 0.00, meaning L1-SVM has performed feature selection. The L1-SVM model, due to its L1 regularization (Lasso), has effectively performed feature selection. Many coefficients are exactly zero, meaning those features are considered irrelevant by the SVM for its decision boundary. The MSM Plane 1, on the other hand, seems to use a broader set of features, with smaller but non-zero contributions from almost all of them.



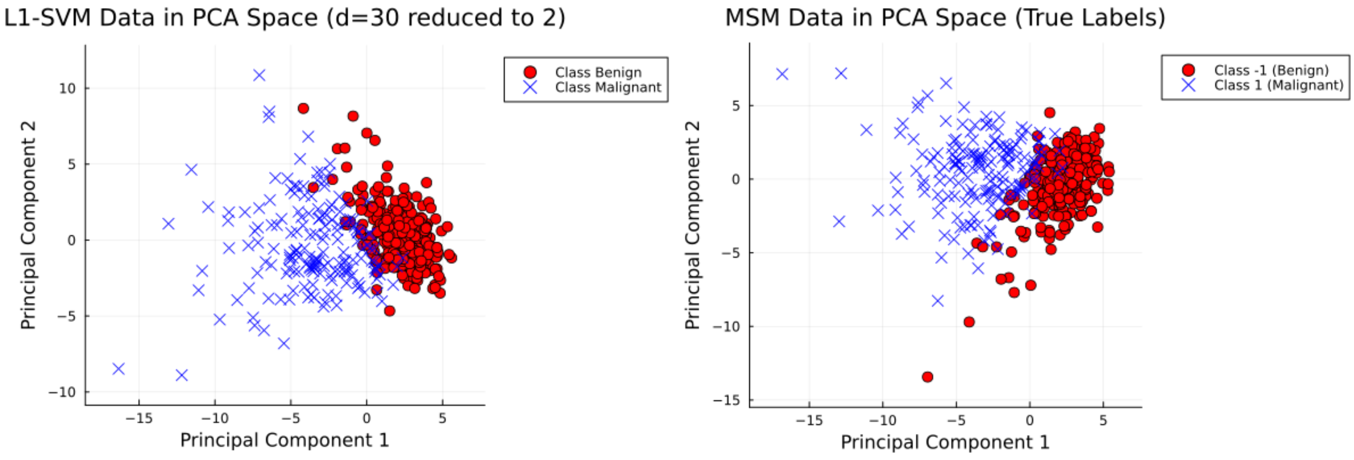
(a) L1-SVM Feature Coefficients (L1-Norm)

(b) MSM Plane 1 Feature Coefficients

Figure 1: Comparison of Feature Coefficients for L1-SVM (Left) and MSM Plane 1 (Right).

Both models assign high importance to features related to `concave_points1`, `smoothness2`, `area3`, and `fractal_dimension3`. This suggests these features are generally strong predictors, regardless of the specific modeling approach. `radius3` is highly influential for MSM, while `smoothness2` and `area3` are dominant for L1-SVM.

3.1.3 PCA Plots



(a) SVM PCA Plot

(b) MSM PCA True Labels Plot

Figure 2: Comparison of PCA Plots showing True Labels for L1-SVM and MSM Data

The Principal Component Analysis (PCA) plots in Figures 3 and 4 show a simplified view of the breast cancer dataset by reducing the 30 original features to just two. This helps us see how well we can separate malignant (blue 'x') and benign (red circle) cases in a two-dimensional space.

In Figure 2 (Left), which shows the L1-SVM data after PCA, we notice that the two classes overlap quite a bit. Benign cases tend to group more on the right, while malignant ones are more on the left, but there's a large area in the middle where both mix. This means the data isn't perfectly separable, even after simplification. That's why a soft-margin SVM—which allows for some mistakes—is useful here. If we added a decision boundary to this plot, it would try to draw the best line to split the two groups, while allowing for some overlap to make the model more flexible and accurate.

Figure 2 (Right) shows a similar picture using MSM. Again, we see that benign and malignant cases are not clearly separated, especially in the center. This overlap explains why the models can't reach perfect accuracy—it's simply a tough dataset to separate cleanly. However, the L1-SVM still performed well, which suggests it handled the messy data better by using its optimization and regularization techniques to focus on the most important features, even if we can't see a perfect split in just two dimensions.

Also, we see that malignant cases (blue 'x') are spread more widely along the first principal component (horizontal axis), especially towards the left. In contrast, benign cases (red circles) are more tightly clustered. This suggests that the features making up the first principal component are especially helpful for telling the two groups apart.

3.2 Predicting Cancer Recurrence

3.2.1 Confusion Matrices

Metric	RSA
True Positives (TP)	2
True Negatives (TN)	26
False Positives (FP)	1
False Negatives (FN)	11
Accuracy	70.0%
Precision	66.67%
Recall (Sensitivity)	15.38%
F1-Score	25.0%
Specificity	96.3%

Table 2: Confusion Metrics of RSA on Test Data

The RSA model demonstrates a moderate overall accuracy of 70.0% in predicting cancer recurrence. Its high specificity (96.3%) indicates strong performance in correctly identifying non-recurrent cases. However, the model struggles with sensitivity, achieving only 15.38% recall, which means it fails to detect most actual recurrence cases. While the precision is relatively high at 66.67%, the low F1-score (25.0%) reflects a significant imbalance between precision and recall. These results suggest that while the RSA model is reliable at ruling out recurrence, it is not effective at identifying patients who are at risk of recurrence.

3.2.2 Coefficients Plots

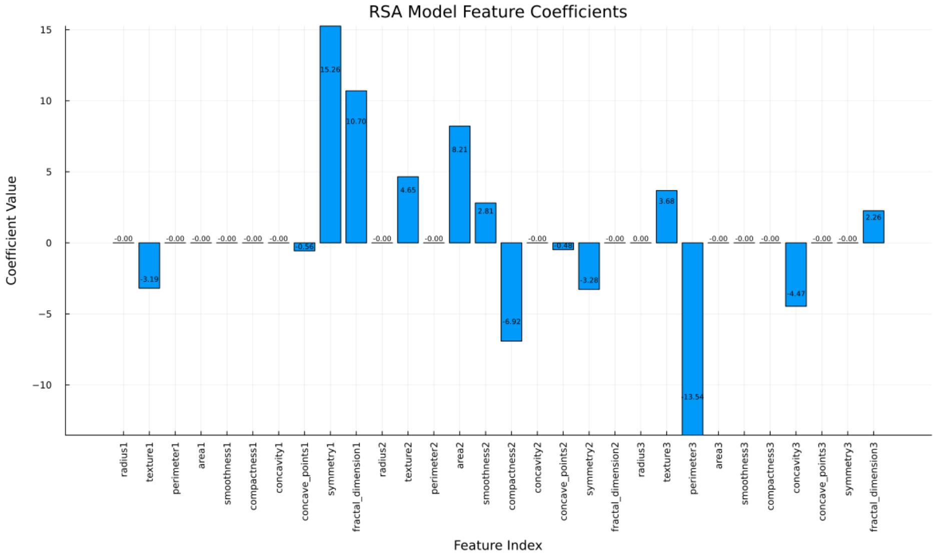
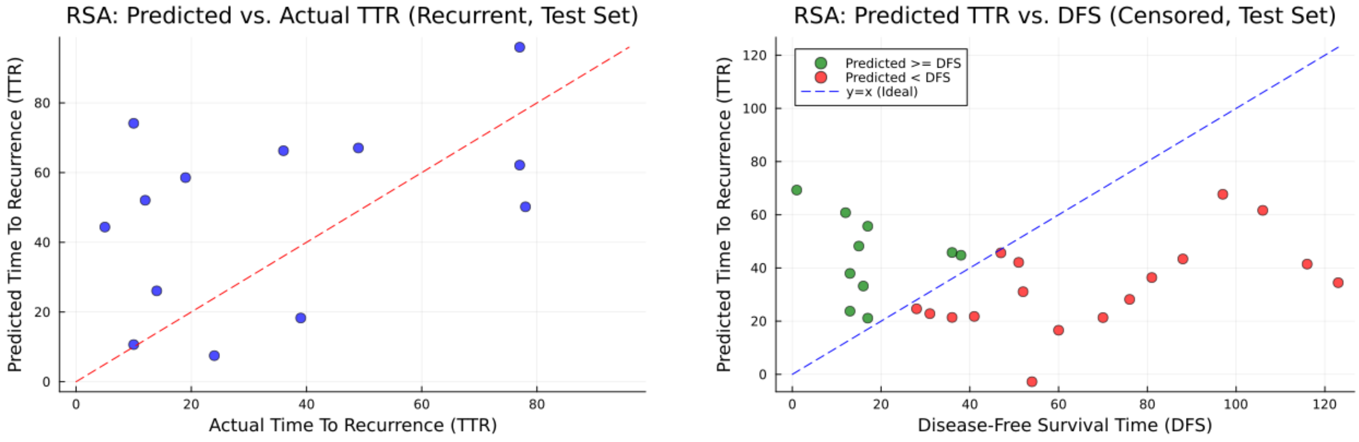


Figure 3: Feature Coefficients for RSA

Figure 3 presents the feature coefficients learned by the RSA model. These coefficients indicate the relative importance and direction of influence of each input feature on the predicted Time To Recurrence (TTR). A larger absolute coefficient value signifies greater importance.

The plot clearly shows that certain features exert a much stronger influence on the RSA model's prediction of TTR. Notably, `concave_points1` (with a coefficient of 15.26) and `texture3` (with a coefficient of -13.54) have the largest absolute magnitudes, indicating they are the most significant predictors in this RSA model. A positive coefficient for `concave_points1` suggests that higher values of this feature are associated with a longer predicted TTR, while the highly negative coefficient for `texture3` implies that higher values for this feature are associated with a shorter predicted TTR. Other features like `radius2` (10.70), `smoothness2` (8.21), `area2` (3.81), and `perimeter3` (3.66) also show substantial positive coefficients, suggesting they contribute to a longer predicted TTR. Conversely, `radius1` (-2.19), `compactness2` (-6.92), `concavity2` (-1.28), and `compactness3` (-4.47) have notable negative coefficients, correlating with a shorter predicted TTR. The presence of numerous coefficients exactly at 0.00 indicates that the L1 regularization in the RSA formulation effectively performed feature selection, assigning zero importance to features deemed irrelevant for predicting TTR.

3.2.3 Scatter Plots



(a) Predicted vs. Actual TTR

(b) Predicted TTR vs. DFS (Censored)

Figure 4: Scatter Plots for RSA Model Performance: Predicted vs. Actual Time To Recurrence (Left) and Predicted Time To Recurrence vs. Disease-Free Survival (Right)

Figure 4 (Left) shows the predicted Time To Recurrence (TTR) against the actual TTR for recurrent patients in the test set. The dashed red line represents the ideal scenario where 'Predicted TTR = Actual TTR'. Points lying close to this line indicate accurate predictions. While some points are near the ideal line, there is a noticeable scatter, suggesting that the model's predictions are not perfectly aligned with the actual recurrence times. Some recurrent cases are underestimated (points above the line, where predicted TTR is higher than actual), while others are overestimated (points below the line, where predicted TTR is lower than actual). The overall spread indicates that accurately predicting the exact time to recurrence for every patient remains a challenging task.

Figure 4 (Right) displays the predicted TTR against the Disease-Free Survival (DFS) time for censored (non-recurrent) patients in the test set. The blue dashed line represents 'y=x', which in this context indicates if the predicted TTR matches the DFS time. For censored patients, the RSA model is designed to predict a TTR that is ideally greater than or equal to their DFS time, reflecting that they have not yet recurred by their last known follow-up. The green circles represent cases where 'Predicted TTR $\geq$ DFS', which is the desired outcome, as it correctly implies the patient has not recurred by their last known follow-up time. The red circles indicate cases where 'Predicted TTR $<$ DFS', which are undesirable outcomes because they suggest a patient who has not recurred (censored) is predicted to recur earlier than their observed disease-free period. The plot shows a mix of both. While a significant portion of censored patients have predicted TTRs above or close to their DFS times (green circles), there are also several red circles, particularly among patients with higher DFS times. This indicates that the model struggles to accurately predict a TTR greater than DFS for all censored individuals, leading to some inconsistencies with the observed non-recurrence. This is crucial for prognosis, as predicting recurrence for a patient who remains disease-free can lead to unnecessary anxiety and interventions. The presence of these red circles suggests that the 'v' slack variable in the RSA formulation is being utilized to allow some violation of the DFS constraint to achieve the overall optimization objective.

4 Conclusion

This project explored how three linear programming models,L1-SVM, MSM, and RSA, can help diagnose breast cancer and predict if it might return.

The **L1-SVM model** was the most accurate for diagnosis, reaching 99.12% accuracy. It reliably identified cancerous cases with few false positives and focused on a small number of important features. In contrast, the **MSM model** struggled, with lower accuracy and more errors in classification.

For predicting cancer recurrence, the **RSA model** had decent accuracy (70.0%) and was good at identifying non-recurrence cases. However, it missed many actual recurrences, which is a key area for improvement.

Overall, L1-SVM is a strong tool for diagnosis, while RSA shows some promise for prognosis but needs better sensitivity. This project also gave us valuable experience working with real medical data and applying optimization methods in Julia. Future work could focus on improving RSA and combining models for better results.

If we had more time, we would like to learn more about LP's work in other medical diseases. One limitation of this project was that we were not as experienced when using Julia Code, and so our work had a lot of trial and error. Overall, the project was an enjoyable discovery in finding how LP models can be used to support the healthcare system.

References

[1] Dedieu, Antoine, et al. "Solving L1-Regularized SVMs and Related Linear Programs: Revisiting the Effectiveness of Column and Constraint Generation." Journal of Machine Learning Research, vol. 23, 2022, pp. 1–41, www.jmlr.org/papers/volume23/19-104/19-104.pdf. Accessed 9 June 2025.

[2] Mangasarian, Olvi L., et al. "Breast Cancer Diagnosis and Prognosis via Linear Programming." Operations Research, vol. 43, no. 4, Aug. 1995, pp. 570–577, www.aaai.org/Papers/Symposia/Spring/1994/SS-94-01/SS94-01-019.pdf,https://doi.org/10.1287/opre.43.4.570.

[3] Winstead, Edward. "Mammogram False Positives Affect Future Screening Behavior." Cancer.gov, 4 Oct. 2024,www.cancer.gov/news-events/cancer-currents-blog/2024/mammogram-false-positives-affect-future-screening.

Dataset:

[4] Wolberg, William, et al. "Breast Cancer Wisconsin (Diagnostic)." UCI Machine Learning Repository, 1993, https://doi.org/10.24432/C5DW2B.

[5] Wolberg, William, W. Street, and Olvi Mangasarian. "Breast Cancer Wisconsin (Prognostic)." UCI Machine Learning Repository, 1995, https://doi.org/10.24432/C5GK50.

Used the code from this website:

[6] scikit-learn. "1.4. Support Vector Machines." Scikit-Learn.org, 2018, scikit-learn.org/stable/modules/svm.html .