Cancer Classification Model - Genetic Detection

I. Introduction

For my CS 307 final project, I developed a machine learning model to classify a patient's cancer type based on genetic data.

Working with this dataset was a completely new challenge for me because each sample included ~4,000 gene-based feature variables, which is a much higher dimensional amount than any previous ML project I've worked on. Because of this, my goal wasn't just to build an accurate classifier, but to figure out how much of that genetic information was actually necessary.

To accomplish this, I used a logistic regression classification model and implemented a feature reduction technique to identify the smallest set of genes that could still achieve perfect prediction accuracy on new unseen test data.

II. Methods

Target Variable — *cancer*:

BRCA: Breast Invasive Carcinoma

PRAD: Prostate Adenocarcinoma

KIRC: Kidney Renal Clear Cell Carcinoma

LUAD: Lung Adenocarcinoma

COAD: Colon Adenocarcinoma

This dataset also contains 4,000 gene-based feature variables (all numeric) that represent genetic measurements for each sample. Each gene feature was used as a potential predictor for the cancer type.

To build the cancer classification model, I first created a pipeline to preprocess the data so it could be used effectively for machine learning. Within this pipeline, the numeric gene features were scaled to standardize their ranges, and imputed where necessary to handle missing values. Since the predictors were already numeric, no encoding was needed.

After preprocessing, I trained and evaluated multiple logistic regression models. Because the dataset had a very high number of features relative to the number of samples, I decided to use feature reduction to make the model faster and more efficient. Specifically, I applied SelectKBest with an ANOVA F-test (f_classif) to select the most important genes before fitting the model.

To tune the model, I tested multiple values of k (the number of selected genes) and compared performance on unseen test data. The final model used a pipeline of preprocessing → SelectKBest → logistic regression, and I selected the smallest k value that still achieved perfect test accuracy.

7	gene_8	...	gene_3990	gene_3991	gene_3992	gene_3993	gene_3994	gene_3995	gene_3996	g
371	0.0	...	0.000000	11.722884	11.831470	7.785282	11.877556	4.926711	0.591871	8
000	0.0	...	0.000000	10.406705	10.795447	8.007330	11.330637	5.041931	1.327170	6
595	0.0	...	0.000000	11.033850	10.292609	6.580491	11.152196	0.000000	0.000000	4
382	0.0	...	0.000000	11.137958	10.401871	6.259530	11.640385	3.068258	0.000000	5
982	0.0	...	0.360982	10.757932	9.944203	8.311430	11.275572	1.580097	0.000000	6

III. Results

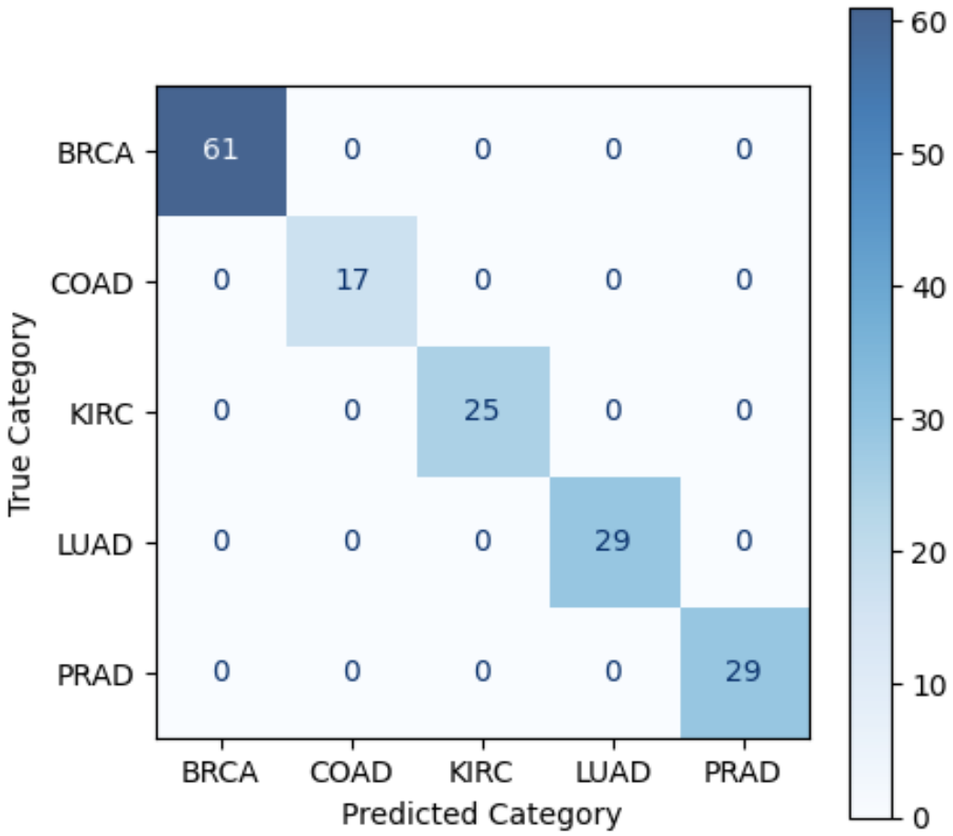
To evaluate model performance fairly, the dataset was split into separate training and test sets. All preprocessing and feature selection were fit on the training data only, and the final accuracy was reported on the unseen test set to estimate how well the model generalizes.

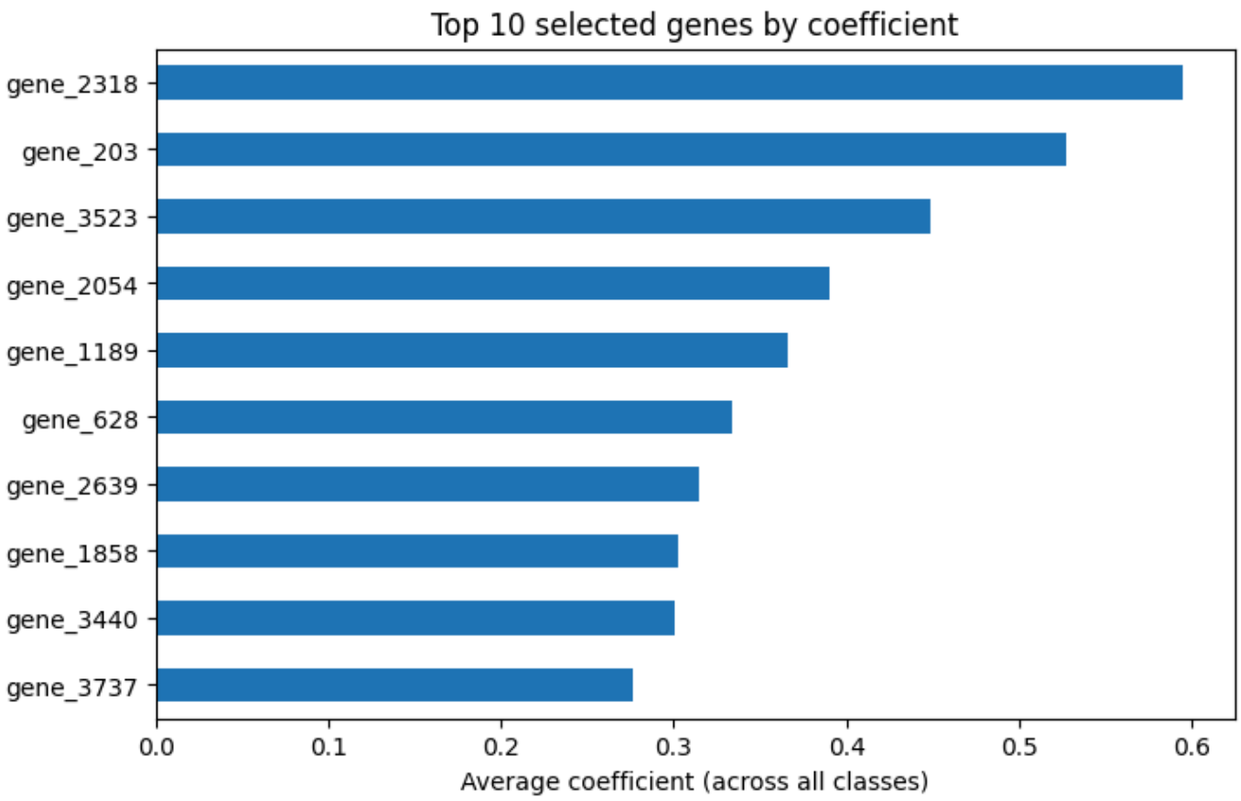
Because the goal of this project was to correctly classify cancer type, I used accuracy as the main evaluation metric (along with a confusion matrix to confirm performance across all five classes).

I tested multiple values of k in SelectKBest to see how many genes were actually needed to achieve strong performance. The results are shown below:

	num features	accuracy
0	1	0.527950
1	5	0.844720
2	10	0.944099
3	20	0.968944
4	50	1.000000

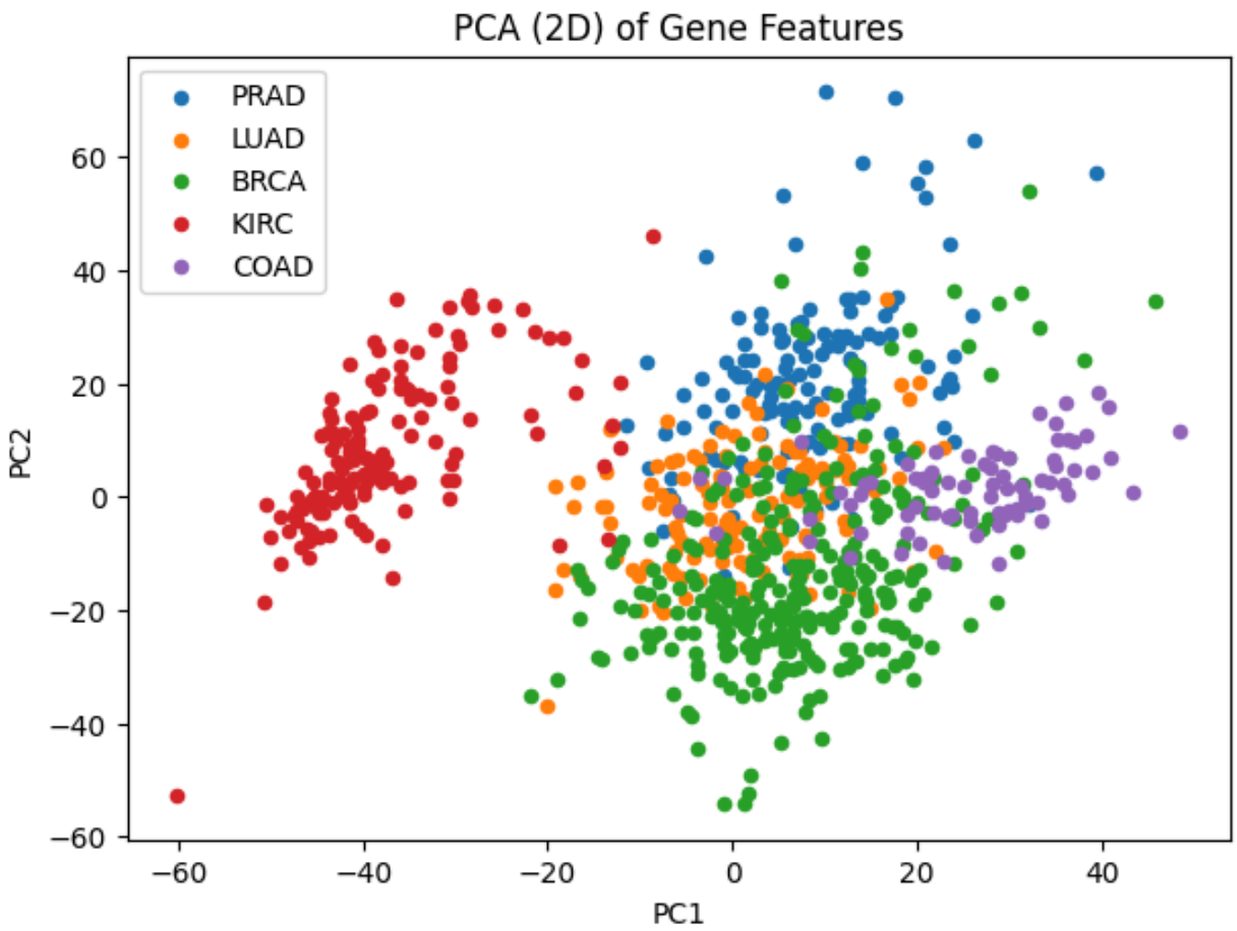
These results show that performance improved quickly as more gene features were included, and the model reached perfect test accuracy with only 50 selected genes. The final selected model was the smallest feature set that still achieved 100% accuracy, and the confusion matrix confirmed that every test sample was classified correctly.





IV. Principal Component Analysis

The PCA visual is a 2D projection of the full gene-feature space, where each point represents a sample and the colors represent the true cancer type. Even after reducing ~4,000 gene features down to just two components, we can see several visibly seperate clusters. Overall, this visual illustrates that there is a strong structure in the genetic features that align with cancer type, which supports how my simple linear model achieved a very high accuracy.



V. Discussion

Conclusion

Overall, the model that I developed did extremely well at accurately classifying cancer type with genetic feature data. With feature selection, the model achieved perfect test accuracy with only a tiny fraction of the original number of features.

Would this model be used in practice?

No, even though the performance was perfect with the data I used, I would not deploy this model without stronger testing. In the world of healthcare, the cost of being wrong is way too high and I'm sure there are things that could have been done to improve my model even further.

Limitations + what I'd do next

The model was tested on just one dataset split, with only ~800 samples, which means it could have been sensitive to a lot of underlying factors like population differences, collection methods, etc. If I were to improve this model going forward, here are a few of the techniques that I would use:

- Cross-validation
- Choose certain genes for interpretability
- Have an actual doctor verify the model's decisions frequently