Precision Oncology for breast cancer patients

Austin Ly, Joyce Yu, Sam McCarthy-Potter, Yanzhe Wang, Haobo Ling

Abstract

This paper introduces different combinations of machine learning methods used in efforts to find more personalized treatments for individual patients and grouped demographics. Much of the publicly available and disjoint datasets on a common topic do not share the same features. In many cases where they do, these features are implicitly assumed to be the same. This paper outlines the models used to more accurately combine datasets, visualize high-dimensional data, and impute missing information using different data science tools, such as UMAP, PCA, and naive bayes, specifically for breast cancer-related datasets. In addition, this paper aims to quantify the predictive power of translational research in the field of precision oncology using NearestNeighbor to map individual patients to a cell lines that proxy them, and a simple feed-forward neural network used to predict the patient response (proxied by the IC50 value) to 34 different chemotherapies tested in a traditional wet-lab setting. Finally, a logistic model was used to determine the survival rate of different individuals receiving chemotherapy versus radiotherapy.

Introduction

Background

According to the World Health Organization, breast cancer is the most commonly diagnosed cancer among women, with millions of new cases each year and hundreds of thousands of deaths. Accurate diagnosis and therapeutic prescriptions are critical to improving patient outcomes and survival rates, yet challenges remain in ensuring timely and precise identification of the optimal medication. Using a machine learning model, we could quickly and accurately match a patient's breast cancer to the most effective known therapy potentially saving lives by reducing the time the disease could spread before the right treatments are applied.

Challenges

Finding datasets on breast cancer that can accurately identify a patient's cancer is particularly challenging due to the limited availability of publicly accessible tumor DNA sequencing. For reasons of patient privacy, most datasets that contain human DNA are restricted. The most abundant breast cancer data sets are anonymized mammograms of the patients, which provide detailed information about tumor size, shape, and other visual characteristics. However, these features reflect only the external presentation of the disease. Without access to the underlying genetic information, it becomes difficult to distinguish between cancers that appear similar on imaging but differ significantly at the molecular level. This limitation reduces the accuracy of predictive models and hampers the development of truly personalized treatment strategies.

Methods

Data sources

We began by collecting comprehensive datasets from Kaggle and supplemented them with additional data from reputable sources. One of the primary datasets used was from Duke University, which included technical and procedural details from breast MRI scans of patients undergoing diagnostic imaging (Harowicz, 2024). We integrated this with the METABRIC dataset, which contains clinical and pathological information on breast cancer patients, including diagnostic, treatment, and outcome-related features (Evitan, 2020). Key variables from METABRIC include patient demographics, cancer subtypes, treatment history (e.g., surgery, chemotherapy, radiotherapy, hormone therapy), molecular classifications (e.g., PAM50, ER/PR/HER2 status), and prognostic indicators such as tumor size, stage, grade, and survival. To further enhance our dataset, we incorporated SEER registry data, which adds clinical, demographic, and pathological details, such as patient age, race, marital status, AJCC staging, hormone receptor status, lymph node involvement, and survival outcomes(Teng, 2019).

Exploratory data analysis - Merging disjoint datasets

We merged all datasets using 3 to 9 matching columns and conducted extensive data cleaning to ensure compatibility across sources. This comprehensive merged dataset served as a foundation for modeling the full range of clinical information we might obtain from a patient. Based on this dataset, two logistic and three regression models were developed. The first logistic model predicts whether radiation therapy or chemotherapy is more likely to yield the longest survival time, based on patient-specific features. The second logistic model, a random forest model, is designed to impute missing key clinical variables, which are later used to guide chemotherapy selection. Using the predicted features, each patient is matched to the most similar cell line in the Harvard dataset (Sorger, Mills, & Hafner, 2018). Finally, the therapy that was most effective in inhibiting that matched cell line is likely to be recommended as the optimal treatment based on the regression models predicting patient response trained on three different versions of the final processed dataset.

We selected these methods to make the most of the available datasets and to derive meaningful insights despite limitations in direct patient-to-treatment mappings. Specifically, we couldn’t link patients to individual medications, only to their response to radiation or chemotherapy. Separately, we had a dataset detailing how various medications affected immortalized cell lines. To bridge this disconnect, we integrated the datasets wherever possible and developed a chain of additional models to impute missing information and more accurately map a patient to a cell line that proxied them. Each model uses outputs from the previous, completed dataset to inform the next, enabling us to extract value from otherwise disconnected sources.

We chose logistic models to determine the survival outcomes because the prediction value is ordinal rather than continuous. In contrast, we chose regression models to determine patient response to specific chemotherapies because the prediction value is on a non-negative

and continuous scale. Using a trial-and-error approach, we identified which modeling strategies were most effective for predicting key features needed to pass information between datasets. This modular setup also makes our system flexible and scalable, allowing new data to be incorporated by retraining only the relevant segment of the model pipeline. This approach allows us to leverage experimental data not originally designed for our use case, making data acquisition feasible.

Exploratory data analysis - UMAP Analysis of Patient and Drug Datasets

To explore the structure of our datasets and visualize patient and drug-level patterns, we applied Uniform Manifold Approximation and Projection (UMAP), a nonlinear dimensionality reduction technique. Before running UMAP, all numeric features were preprocessed by imputing missing values (using mean imputation) and scaling them using StandardScaler from scikit-learn to ensure comparable variance. Categorical variables were either excluded or encoded, depending on context. For datasets containing both demographic and molecular features, we combined all numeric variables, allowing UMAP to capture a global structure based on both clinical and chemical attributes.

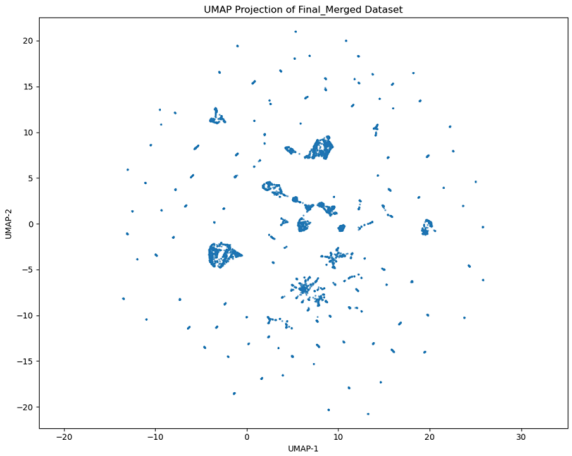
UMAP was configured with n_neighbors = 15 and min_dist = 0.1, and we used 2 output dimensions to facilitate visualization. In some cases, random state and sampling were used to reduce computation and ensure reproducibility. When we applied UMAP to each individual raw dataset (SEER, METABRIC, DUKE), the resulting visualizations showed clearly separated clusters:

Figure 1. The UMAP results of raw datasets

This behavior is expected, as each dataset comes from a distinct clinical or research source, with differences in variables such as patient age, cancer subtype, disease stage, and molecular profiling method. Since UMAP preserves local neighborhood structures, patients with similar characteristics were grouped together, and the preexisting subtype or institutional boundaries were amplified in the embedding. These results confirmed that even simple demographic and clinical variables are sufficient to distinguish patient subpopulations across datasets.

Next, we merged the patient demographic data from all three datasets. The resulting UMAP plot still displayed several dense and distinct clusters. This is largely because we were using a limited set of clinical and demographic variables, many of which were categorical (e.g., cancer stage, race, treatment status). Such features tend to cluster data strongly, especially when patient records come from institutions that follow consistent protocols. And the resulting clusters likely correspond to different data sources or dominant clinical subtypes, and mirrored what we observed in the individual datasets.

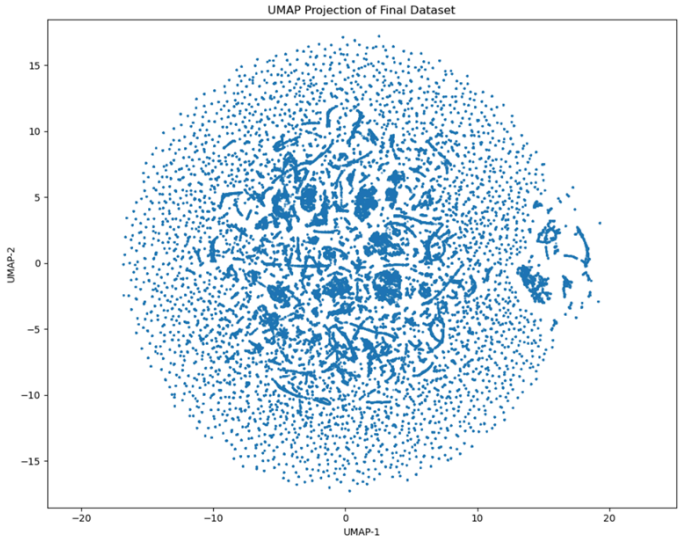
Figure 2. The UMAP result of merged patient dataset



After imputing missing racial information with Naïve Bayes, mapping cell lines to patients with Nearest Neighbor, followed by mapping up to 34 chemotherapies to each patient to incorporate drug molecular information (i.e. molecular fingerprints, hydrogen-bond counts, molecular weight, TPSA, and other continuous features) the structure of the UMAP projection changed significantly. Instead of forming multiple discrete clusters, the data points spread out more evenly in a circular or disk-like shape. This is because chemical fingerprints represent a high-dimensional, continuous space. Unlike categorical clinical data, molecular features vary smoothly across compounds.

Moreover, since molecular features are far more numerous and have higher variance, they tend to dominate the UMAP's distance calculations. As a result, clinical differences became less prominent, and the overall structure reflected similarity in drug chemical space rather than patient cohort characteristics.

Figure 3. The UMAP result of final merged dataset



Race prediction and Imputations

In addition to the logistic regression and random forest models, we developed a model aimed at identifying potential racial disparities in breast cancer patient data. The objective was to explore whether clinical, genomic, and imaging features could predict a patient's race, highlighting possible systemic biases or differential clinical patterns.

Equation 1: Distance Equation of NearestNeighbors

$$d(p,q) = \left(\sum_{i=1}^{d} |p_i - q_i|^m \right)^{1/m}$$

Race prediction and Imputations - Data and Preprocessing

The race prediction model was developed using a comprehensive dataset comprising 140 features and 9,222 samples, which integrated clinical, genomic, and imaging data to capture a wide range of potentially relevant variables. To ensure data quality, features with more than 80% missing values were removed, and the remaining missing entries were imputed using column means, preserving as much of the original dataset as possible without introducing significant bias. Once the dataset was cleaned, StandardScaler was applied to normalize the feature values, a critical step to prepare the data for algorithms like Gaussian Naive Bayes and Principal Component Analysis (PCA), which are sensitive to feature scale. This preprocessing pipeline ensured that the model received a consistent and well-structured input, enhancing its performance and reliability during training and evaluation.

Race prediction and Imputations - Model implementation using Gaussian Naive Bayes

Gaussian Naive Bayes (GNB) was chosen as the baseline model for its simplicity, ease of interpretation, and computational efficiency, making it well-suited for initial experimentation on high-dimensional clinical datasets. GNB operates under the assumption that all features are independent and follow a normal distribution—assumptions that become more valid after applying feature scaling through standardization. Despite its simplicity, the model served as a useful benchmark for identifying broad racial classification patterns and understanding the relative importance of different features. Its transparent structure allowed for straightforward interpretation of results, providing a foundational reference point for evaluating more complex models in future iterations.

Race prediction and Imputations - PCA Implementation:

Principal Component Analysis (PCA) was applied to reduce the dataset to 36 principal components while retaining 90% of the original variance, effectively balancing dimensionality reduction with information preservation. This technique was particularly important given the dataset's high dimensionality, which included a large number of clinical and genomic features. By transforming the data into a lower-dimensional space, PCA helped minimize noise and redundancy, improving model efficiency without significantly compromising the underlying structure of the data.

Implementation Details for racial feature prediction / imputation:

The Gaussian Naive Bayes model was implemented using the scikit-learn library, with the primary objective of predicting the *Race* feature. To evaluate the impact of dimensionality on model performance, the classifier was trained on both the original full feature set and a dimensionally reduced version obtained through Principal Component Analysis (PCA). highlighting the model's behavior under different feature space conditions and providing insights into the trade-offs between computational efficiency and predictive accuracy.

Technical Challenges of racial feature prediction / imputation:

A major technical challenge encountered during model development was class imbalance, with the Caucasian class significantly outnumbering other racial groups such as African American, Asian, and Hispanic. This imbalance resulted in a model that achieved high overall accuracy by favoring the majority class, but at the cost of poor recall for the minority classes. To mitigate this issue, Principal Component Analysis (PCA) was applied to reduce the dimensionality of the feature space. This transformation helped to partially alleviate the imbalance by amplifying patterns relevant to minority classes, thereby improving recall. However, this came with a trade-off, as precision slightly decreased due to increased misclassification rates.

Translational mapping of Cell lines to Individual Patients

In this part, we chose the one-hot encoder and standardscaler to deal with categorical and numerical features. And the ColumnTransformer was applied as the preprocessor for the

cell line dataset. With the ‘Race’, ‘T stage’, ‘Age’ columns, the NearestNeibor was able to predict the distances and indices between the patient ID and the cell line ID. Therefore, after getting the outcomes flattened, a csv file containing patient ID, cell line ID, similarity distance was outputted.

IC50 Determinations using non-linear regression - Data generation and Preprocessing

In part of the EDA process used to identify features for training, IC50 values were calculated for 34 potential anti-cancer drugs that were each tested against 35 different cell lines excised from breast cancer patients of different ages and ethnic groups using Python SciPy, NumPy, and Math libraries. This dataset originates from the Sorger lab at Harvard University (Sorger, P.K. et al 2019). The “IC50” is known as the inhibitory concentration of a drug causing 50 percent inhibitory response from the measured activity (Motulsky, H. 2024). The IC50 (sometimes referred to as the EC50 or “effective concentration”) is a widely used measure for drug efficacy. The lower the IC50 value, the more potent the therapeutic candidate is (Srinivasan and Lloyd 2024).

Equation 2 and 3 (Top equation) half-way response and (Bottom equation) Non-linear regression sigmoidal curve fitting for the determination of a drug’s IC50 value.

$$Fifty = \frac{[\text{Top - Bottom}]}{2}$$

$$Y = \text{Bottom} + \frac{\text{Top} - \text{Bottom}}{1 + 10^{\left((\text{IC50} - X)\cdot \text{HillSlope} + \log\left(\frac{\text{Top}-\text{Bottom}}{\text{Fifty}-\text{Bottom}} - 1\right)\right)}}$$

The measured activity is denoted by the author as “The normalized mean growth inhibition rate” and is indicative of cell proliferation and inhibition activity. For the author to quantify this activity, their data collection method was done by the following assay setup: Each cell line had their own set of 4 x 384-well plates with cells seeded near 40% confluency per well. One plate was used as a control (i.e. cell count before or at chemical treatment), and 3 plates were used as triplicate conditions to test each cell line-to-small molecule pair with a 9 step, 1-to-3 fold dilution series starting at 10uM of the drug concentration. An additional small molecule (dimethyl sulfoxide or “DMSO”) was included as a control for the 34 anti-cancer drug candidates in the study. Please note that the authors who produced the data states they used a 1-to-2 fold dilution series, however, looking at the raw .csv file, it is actually a 1-to-3 fold series. Fixation and staining of cells were done to aid the cell count of the triplicate conditions used to measure cell activity where x(c) is the mean of the measured ‘Live’ cell counts after a given treatment, x0 is the mean of the ‘Live’ cell counts from day 0 or the untreated plate grown in parallel, and xctrl is the mean of the ‘Live’ cell counts of the DMSO-treated control wells for all technical replicates. The values are then normalized by the DMSO-treated control wells to account for the size differences of each singular cell across different cell types that have an impact on confluency and initial cell plating. The following figure composes the final measure of activity denoted by the Sorger as Mean Normalized Growth Rate Inhibition Value, and with 34 drugs, 35 cell lines, and 9 points per cell-line-to-drug-pair; the dataset used to calculate the IC50

was 10,710 rows long (Sorger, P.K., et al 2019).

Equation 4:
Cell activity defined by Sorger et al (2019) as "Normalized growth rate inhibition"

$$\text{Normalized GR Inhibition} = 2^{\frac{\log_2\left(\frac{x(c)}{x_0}\right)}{\log_2\left(\frac{x_{\text{ctrl}}}{x_0}\right)}} - 1$$

To fit the IC50 curve, each cell-line-to-drug pair had their cell activity plotted against the y-axis and their test concentrations on the log-scaled x-axis. Isolating for the IC50 in the equations and fitting the curves helped to determine "invalid IC50 values" which we define as an ineffective drug to the particular cell line. IC50 values exceeding the maximum dosing concentration were excluded from training as these drugs were deemed ineffective in relation to other drug candidates for the same cell line. This process helped to narrow down more effective anti-cancer therapeutic candidates for particular patients using the cell line as the patient's proxy.

Exploratory data analysis and feature extraction of chemical descriptors

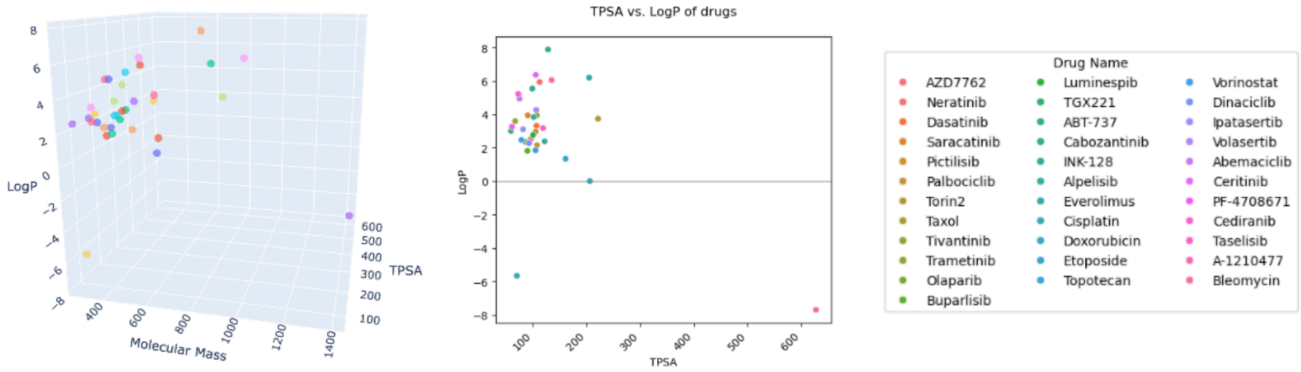
In addition to characterizing drug's effectiveness against their respective targets by calculating their IC50, additional chemical features were extracted from 34 SMILES strings using the RDKit library for each drug candidate. Two most important chemical descriptors used for training were the drug's LogP and topological polar surface area (TPSA) values.

LogP or the octanol-water partition coefficient for a drug helps to measure how easily absorbable or hydrophilic a compound is in a living organism (Green 2024).

TPSA Topological Polar Surface Area was also determined which is related to a molecule's ability to form hydrogen bonds and pass through membranes (Romanick and Holt 2023). Generally it is more favorable to have lower TPSA values, indicating drugs that are more extensively metabolized with slow clearance.

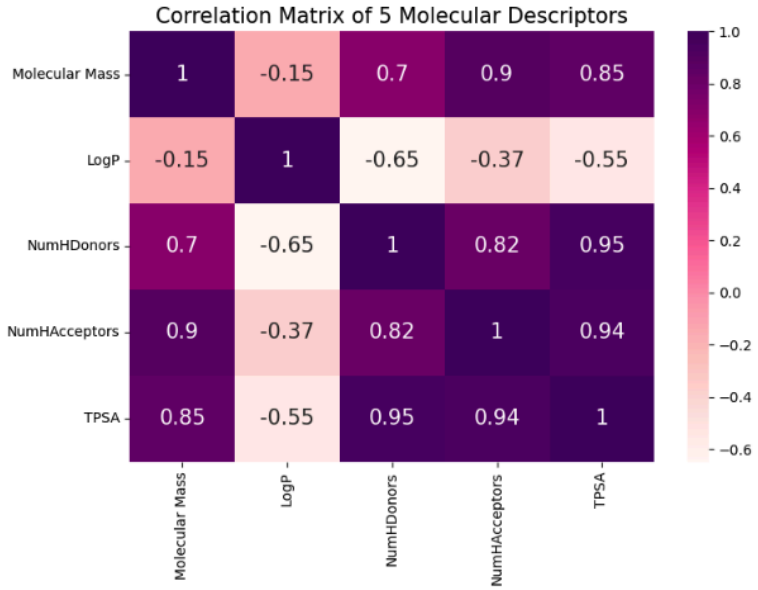
Although there are more descriptors to a drug that goes well beyond LogP and TPSA, these two features are good measures for how a drug may be absorbed, metabolized, and cleared by a living organism; and are both helpful measures in pharmacodynamics (i.e. what the drug does to the living organism) and pharmacokinetics (i.e. what the living organism does to the drug) studies used to determine ADME properties in preclinical research. In other words, they are both good measures of in-vivo drug response and may help us determine how safe the drug candidate is in living organisms (Daina et al. 2017).

Figure 4:
LogP and TPSA of 34 potential anti-cancer therapeutics for breast cancer



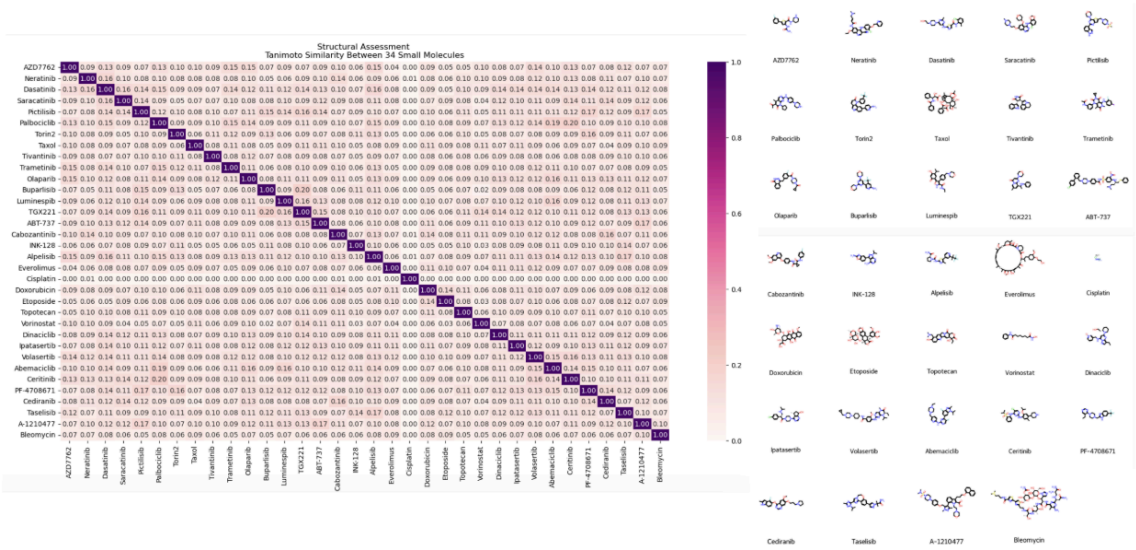
In addition to LogP and TPSA, 3 additional chemical descriptors were extracted to use as features for training based on the Lipinski rule of 5 that is used to characterize drug candidates for therapeutic use (OMx Personal Health Analytics Inc. 2025) .

Figure 5:
Heatmap visualizing the correlation of 5 chemical descriptors that were feature engineered from 34 potential anti-cancer drugs for downstream applications and model training



Additional information was explored from the 34 small molecules to assess drug diversity, but not used for training. In a practical setting, it is important to have a more diverse set of compounds for the same ailment in the case one dissimilar compound is not effective. Pairwise similarity scores were calculated using tanimoto distances between fingerprinted SMILES strings characterizing the structural diversity of the anti-cancer drug candidates (Bajusz et al. 2015). Given the IC50 values, chemical descriptors, and chemical structures; the chemical dataset used in this report is representative of a relatively good set of effective, safe, and diverse chemotherapies to recommend to patients with breast cancer.

Figure 6:
Heatmap and RDkit mol objects visualizing structural diversity of 34 potential anti-cancer drugs



Patient drug-response or IC50 predictions - Simple Feed Forward Neural Network

After cell lines were matched to patient records using our second model, each patient was mapped up to 34 small molecules based on the proxied cell-line that the drug was shown to be effective against (i.e. IC50 value < 10nM). A simple feed forward neural network was constructed to predict the patient response to the set of 34 potential and effective anti-cancer drugs with valid IC50 values. The purpose of this simple feed forward neural network is to predict the IC50 (target) to a set of drugs based on the following selected patient demographic and chemical features: Age, Race, T-stage, Cell Line, Small Molecule, Molecular Mass, LogP, TPSA, NumHDonors, and NumHAcceptors. All categorical features were one-hot-encoded. The architecture of this neural network consisted of 6 layers. This separate model architecture was meant to condense the important features separately and maintain the importance of the cell line features.The network takes two sets of features as input, processed through parallel pathways. The first pathway processes 48 features. The second pathway processes 38 features. Each of these input feature sets is passed to a Linear Layer, which transforms the 48 features into 32 latent features and the 38 features into 32 latent features, respectively. After each of these initial Linear Layers, a ReLU Activation Function is applied in parallel to the respective 32 latent features. The outputs of these two parallel pathways are then combined and passed into a subsequent Linear Layer with 64 neurons, which is followed by another ReLU Activation Function. Finally, the output of the second ReLU activation function is fed into a final Linear Layer which outputs 32 features.

Equation 5: Linear regression used in simple feed forward neural network.

$$y = w_1x_1 + w_2x_2 + w_3x_3 + \cdots + w_nx_n + b$$

Equation 6: ReLu activation functions used in simple feed forward neural network.

$$f(x) = max(0, x)$$

The neural network was paired with an Adam optimizer for optimization with a learning rate of 0.001. The Adam optimizer (or “Adaptive momentum optimizer”) is a popular optimization algorithm for neural networks (Kingma and Lei Ba 2015). It is similar to stochastic gradient descent, however it combines the idea of momentum and root mean square propagation (RMSprop). The next few paragraphs explain our understanding of the Adam optimizer and why we chose to use it.

In momentum, the vanilla/stochastic gradient descent is optimized by subtracting a summed proportion of previous gradients (i.e. “Velocity” or vt) instead of the most recent loss gradient itself (wt). This summed proportion of previous gradients is normalized by the number of iterations the algorithm has gone through. It is a weighted average called “velocity” in the ML world. Velocity is calculated using a new hyperparameter (beta) that scales the amount that previous velocity and previous gradients contribute to the next iteration of weight values. Beta is typically less than 1, therefore over multiple iterations, previous velocities are compounded by beta making velocity smaller for previous iterations and larger for more recent iterations.

Equation 7: Vanilla gradient descent

$$W_{t+1} = W_t - \alpha \nabla W_t$$

Equation 8: Velocity in the idea of momentum

$$V_{t+1} = \beta W_t + (1 - \beta) \nabla W_t$$

Although small, previous gradients are taken into consideration when updating weights. Rather than subtracting by only the most recent gradient like in Vanilla SGD, momentum subtracts by the most recent gradient AND previous gradients (scaled down to a small amount by beta). The addition of momentum allows weights to change at a relatively larger magnitude compared to SGD as both algorithms converge towards the minimum (SOURCE). Due to a relatively larger change in weight’s magnitude per iteration, momentum helps SGD escape local minimas. However, it may also miss global minimas if the learning rate is too large.

Equation 9: Momentum applied to gradient descent

$$W_{t+1} = W_t - \alpha \nabla V_t + 1$$

In addition to momentum, the Adam optimizer incorporates concepts from root mean squared prop (“RMSprop”). RMSprop is one of many adaptive learning algorithms that is similar to momentum, but created to speed up training time and address the concern of parameters having drastically different weights which slows down momentum.Similar to momentum, a velocity term is calculated with hyperparameter beta. However, the gradient is squared, and the step scales the gradients averaged by the root of squared gradients.

Equation 10:
RMSprop's modification of velocity with mean squared error of the loss gradient.

$$V_{t+1} = \beta W_t + (1 - \beta)\nabla W_t^2$$

Equation 11: RMSprop's modification of gradient descent

$$W_{t+1} = W_t - \alpha \frac{\nabla W_t + 1}{\sqrt{V_{t+1} + \epsilon}}$$

Adam optimizer combines the idea of momentum and RMSprop and accounts for bias with estimated moments (M and V) due to the initial random assignments to parameter weights in our architecture.

Equations 12 to 16: Adam optimizer

Moments:

$$\text{moment1}_t = \beta W_t + (1 - \beta)\nabla W_t$$

$$\text{moment2}_t = \beta W_t + (1 - \beta)\nabla W_t^2$$

Estimated moments:

$$\hat{moment1}_t = \frac{V_t}{(1 - \beta_1^t)}$$

$$\hat{moment2}_t = \frac{V_t}{(1 - \beta_2^t)}$$

modified gradient descent:

$$W_{t+1} = W_t - \alpha \frac{\hat{moment1}_t}{\sqrt{\hat{moment2}_t + \epsilon}}$$

The model was initially trained with 500 epochs that was narrowed down to 300 based on visual inspection of where the model is getting diminishing returns on the space and time resources used to run each epoch. The evaluation metric used per epoch was the mean squared error (MSE) between predicted vs actual IC50 values because the IC50 values run on a continuous non-negative scale. K-fold cross-validation (k=10) was conducted to ensure the models generalized well to unseen data and that final evaluation metrics were not due skewed

due to randomization during the initial splits of our training data Please note that the non-log IC50 values are in units of nM and cannot be negative due to how IC50 is defined.

Three different models were trained with the same architecture: (a) trained with the Cell Line feature after mapping patients to their cell lines and drug set, (b) trained without the Cell Line feature after mapping patients to their cell lines and drug set, (c) trained without the cell line feature with some randomization used to impute missing racial information in the initial dataset based on existing racial proportions in the same dataset (i.e. no Naïve Bayes model used) and randomized mapping of cell lines to patient records. Running the first and second model provided insight on the predictive power of general translational research in the field of precision medicine. Running the third model provided a baseline for how well the naïve bayes and NearestNeighbor model imputed missing information and mapping cell lines to patients, as well as an indirect indicator for how much these two features matter in the field of precision medicine. Overall, these 3 datasets trained on the same architecture assume that the cell lines are a proxy for the patient, and take on a more translational medicine approach when predicting patient response to potential drug candidates in the field of precision medicine. The final evaluation metric used was mean squared error and the R^2 value to help determine the amount of variability from the dataset that each model captured.

Life Expectancy prediction

In order to test the success rate of our predicted chemo treatment over radiation treatment we first attempted to use a linear regression model to predict the survival weeks. Utilizing the same data set as the race prediction model we used all 76 of the numeric variables of the merged data set to predict survival rate in months. Unfortunately this model proved to be highly unreliable and only predicted a narrow range of survival rates around the average survival.

To address the limitations of our initial approach, we developed a more focused neural network. We narrowed the feature set to 7 variables to maintain human interpretability with the addition of if chemo or radiation therapy was used. Using TensorFlow's Keras API, we implemented a binary classification neural network aimed at predicting whether a patient would survive for five years following treatment rather than estimating an exact survival duration, which proved unreliable. The model architecture begins with a dense layer of 64 neurons, using ReLU activation to capture non-linear relationships. This is followed by batch normalization and a dropout layer with a rate of 0.2 to mitigate overfitting. Two subsequent dense layers with 32 and 16 neurons, both activated by ReLU, with an additional batch normalization layer. The output layer consists of a single neuron with a sigmoid activation function, which returns a probability representing the likelihood of five-year survival. The model was then trained for 300 epochs.

Results and Evaluation

Racial Model Evaluation

The model's performance was assessed using a comprehensive set of evaluation metrics, including accuracy, precision, recall, F1-score, and a confusion matrix to capture both

overall and class-specific predictive behavior. To understand the impact of feature selection and dimensionality reduction, the model was tested across three distinct datasets: randomly selected feature subsets consisting of 5, 7, and 10 features; the full feature set comprising 140 features; and a PCA-reduced set containing 36 principal components. This comparative approach allowed for a nuanced analysis of how different feature configurations influenced model effectiveness, particularly in handling class imbalance and maintaining predictive accuracy.

Figure 7: Racial Model confusion matrix

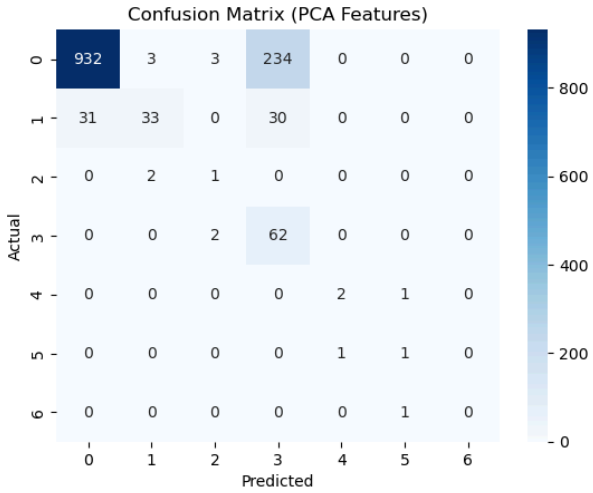
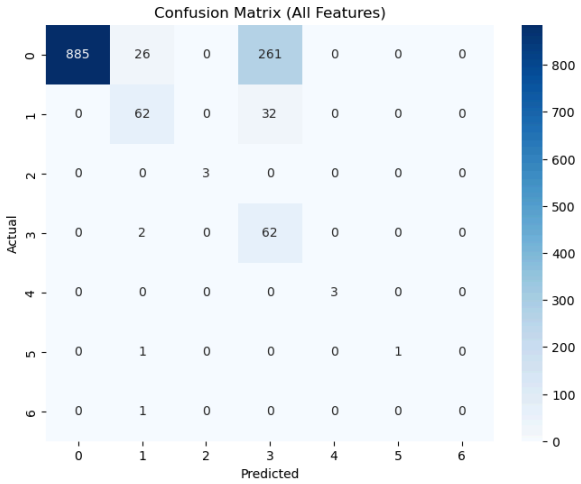


Figure 8: Racial Model PCA confusion matrix



Racial Model Evaluation - Confusion Matrix Analysis:

The confusion matrix for the PCA-transformed Gaussian Naive Bayes model revealed a pronounced disparity in predictive performance across racial groups. While the majority class, Caucasian, was identified with high precision (0.97), this came at the cost of significantly lower recall for minority classes such as African American and Asian, underscoring the model's

susceptibility to racial bias stemming from underlying class imbalance. However, the application of PCA led to notable improvements in recall for some underrepresented groups, including Hispanic and Native American individuals. This enhancement is likely due to PCA's ability to compress overlapping clinical and genomic features, making minority class patterns more distinguishable within the reduced feature space.

Racial Model Evaluation - Interpretation of PCA Effectiveness:

Principal Component Analysis (PCA) proved effective in retaining essential variance while significantly reducing the dimensionality of the dataset, enabling the model to concentrate on the most relevant and informative features. The 36 principal components preserved during the transformation revealed distinct racial clustering patterns that were less apparent in the original high-dimensional feature space. This enhanced separability supported the utility of PCA not only in improving computational efficiency but also in uncovering latent structures within the data, justifying its application in subsequent modeling efforts.

Racial Model Evaluation - Model Prediction Scores for Race feature

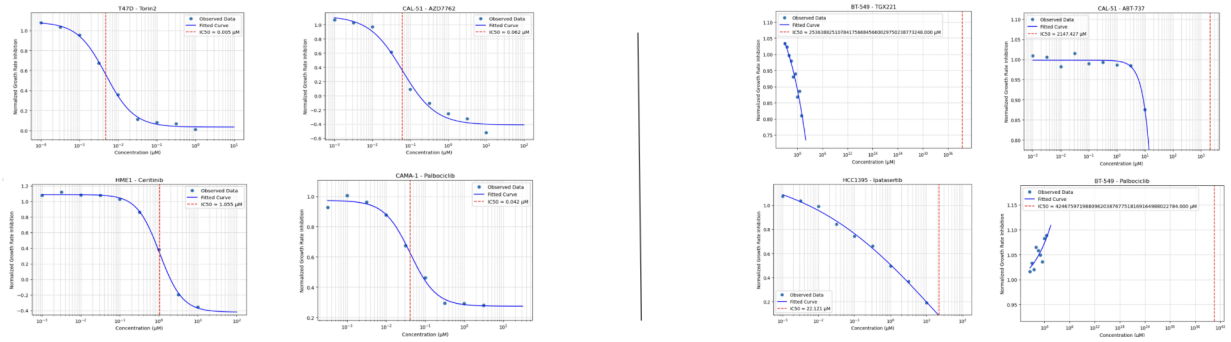
Dataset	Accuracy	Macro F1	Minority Recall
5 Features	79%	0.52	0.31
All Features	98%	0.63	0.39
PCA Features	77%	0.43	0.52

Patient-drug response and IC50 predictions

Our IC50 calculations yielded 1,150 IC50 values, of which 324 came from small molecules that were determined to be relatively ineffective against the treated cell line. These drug-to-cell line pairs producing IC50 values estimated well beyond the physical concentrations (>10uM) tested in the lab, and would require additional testing at higher doses to see any inhibitory effects of the cell proliferation activity inhibited by the drug. Thus, these 324 cell-to-drug line pairs were excluded from downstream applications when training machine learning models.

Figure 9:

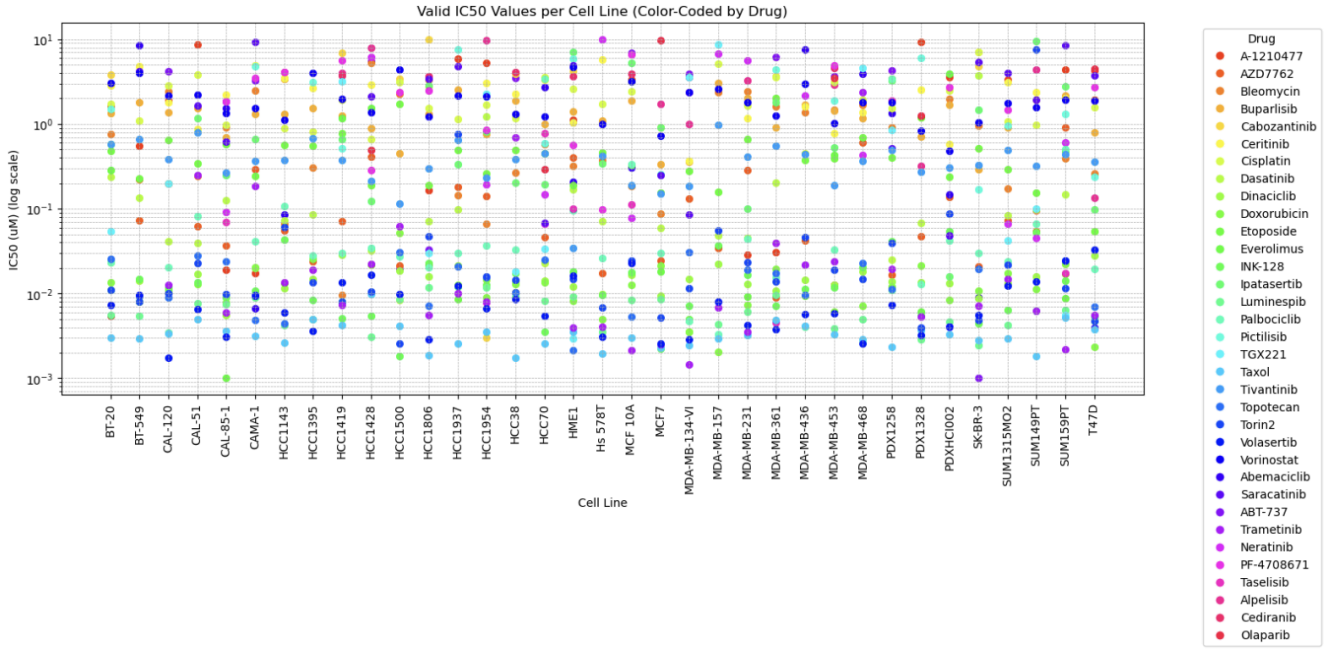
(Left) 4 samples of dose-response curves exhibiting valid IC50 values. (Right) 4 samples of dose-response curves exhibiting invalid or overestimated IC50 values. An upper constraint can be applied to better estimate the IC50 values or have tested in the lab.



Nevertheless, each cell-line still received at least one effective drug causing a reduction of 50% activity. In total, 826 of 1,150 drug candidates were determined to be effective (IC50 <=10 uM) treatments against 34 cell lines, and 50 cell-line-to-drug pairs were determined to have the drug be extremely potent against that particular cell-line (IC50 <0.001 uM).

Figure 10:

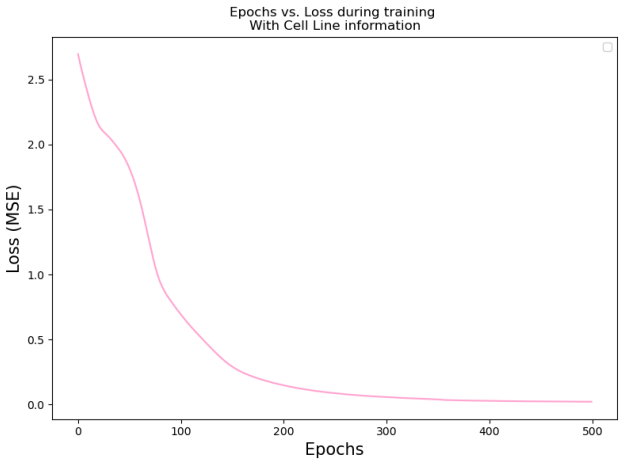
826 valid IC50 values plotted via dot-plot with Cell Lines on the x-axis, log(uM) concentration on the y-axis, and dots landed on IC50 values color-coded by drug name.



The loss during training had a steep decline between 0 to 300 epochs. After 300 epochs, there was a gradual decline in mean square error. This helped determine the number of epochs (i.e. 300) that was sufficient to train the neural network without spending more space and time that provided diminishing returns.

Figure 11:

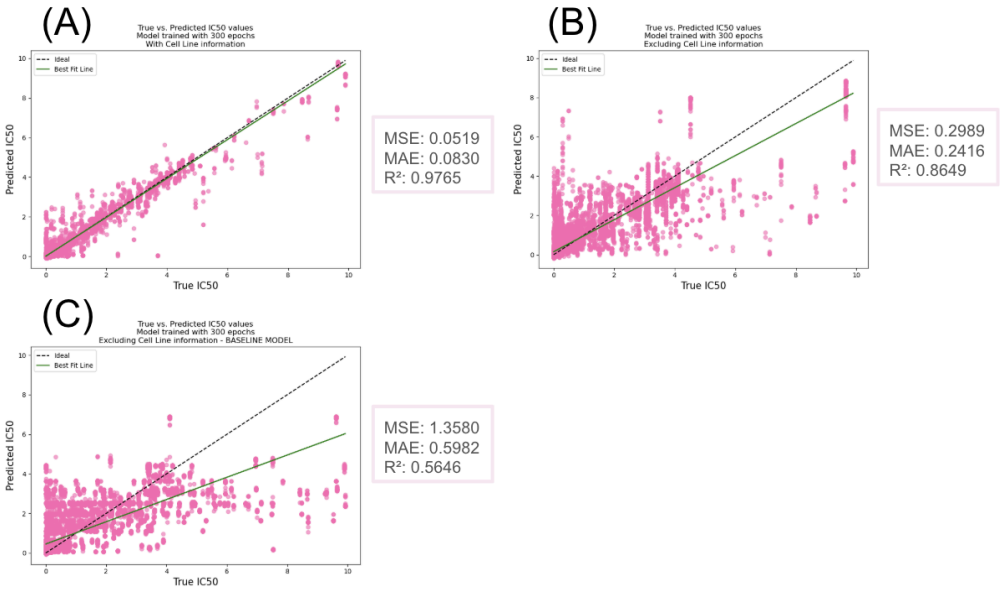
Epochs vs. Mean squared error (MSE) loss of neural network trained with all selected features.



After 300 epochs, each of the 3 models yielded a mean square error less than 2. The model trained with the cell-line names as categorical features captured about 97.65% of variance in our data; and had the lowest mean squared error of 0.0519 amongst the 3 models, indicating that the predictions did not stray not far from the true IC50 values. The model trained without the cell line names as categorical features captured about 86.49% of variance in our data with a mean square error of 0.2989. Finally, the baseline model trained without the cell line names as categorical features, with some randomization in imputing missing racial information using existing proportions of racial data, and randomized mapping of cell lines to patients had only captured 56.46% of variability within the data and had the highest mean square error amongst the three models at 1.3580.

Figure 12

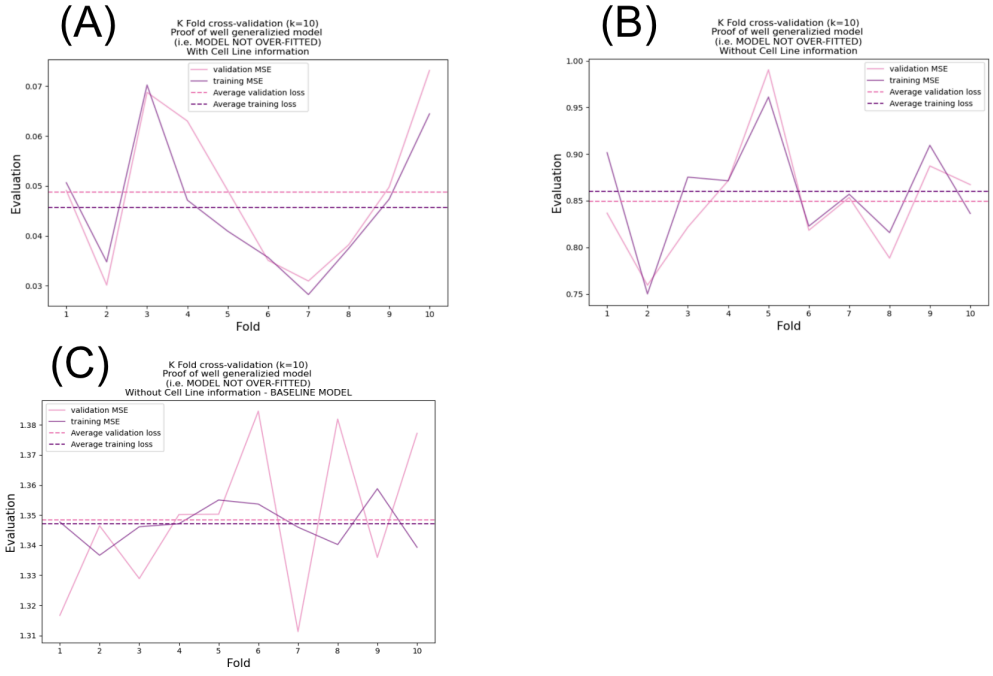
Results of 3 Models trained on the same Simple Feed Forward Neural Network.



(A) Most complex model trained with all selected features with missing data imputed by Naïve Bayes model and cell-lines mapped to patients with Nearest Neighbor model. (B) Second most complex model trained with selected features that excluded one-hot-encoded cell line name, with missing data imputed by Naïve Bayes model and cell-lines mapped to patients with Nearest Neighbor model. (C) Baseline model trained with selected features that excluded one-hot-encoded cell line name, with imputation of missing racial information using weighted randomization of existing racial proportions in the dataset and random assignments of cell-lines to patients.

K-fold cross validation (k=10) indicates that our initial results seen above are consistent with multiple random splits of our data, indicating that each of the three models generalizes well to unseen data without overfitting. Through visual inspection of our baseline model, the validation error varies widely across different splits in our training data. In contrast, the other 2 models that contain racial imputations using Naïve Bayes and cell-line-to-patient mapping using nearest neighbors generalized much better to unseen data, with validation and training errors following closely to one another across different k-folds. This indicates that the more complex models incorporating Naïve bayes, nearest neighbors, and a simple feed-forward neural network not only generalizes well to unseen data; but also makes more accurate predictions in how a patient may respond to a particular set of anti-cancer drugs when compared to the baseline model.

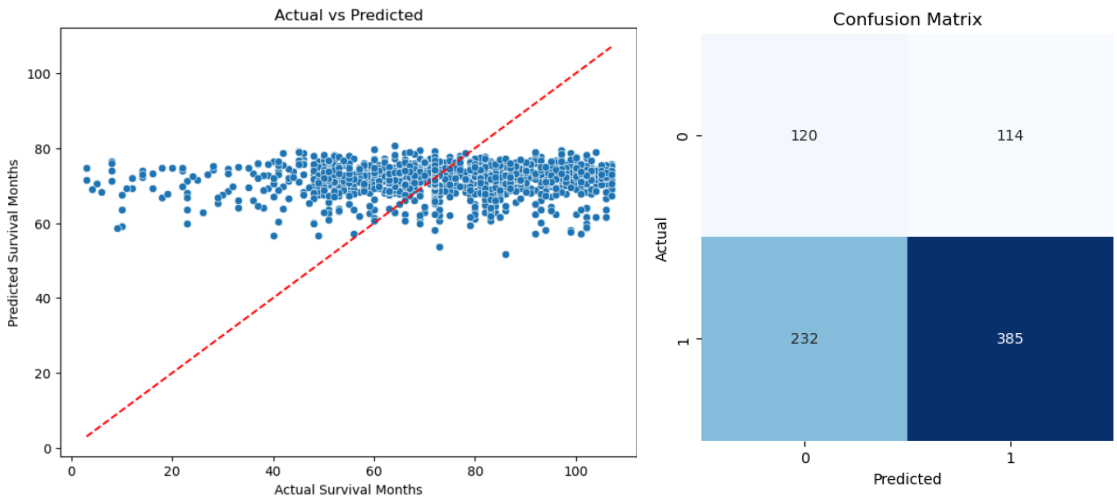
Figure 13:
Results of the 3 Models trained on 10 different splits of their selected features and design matrix to show generalizability to unseen data and no overfitting.



Life expectancy

To evaluate the effectiveness of the life expectancy linear regression model we utilized mean squared error and a line plot that displaced the predicted life expectancy vs the actual. The evaluation showed that the linear regression model was highly inaccurate so a new model was required. The second binary classification neural network was evaluated off of a confusion matrix (where 1 was positive for 5-year survival and 0 was negative) as well as the accuracy, recall and precision for each group.

Figure 14: Life expectancy linear regression line plot



Discussion

UMAP as a visualization tool for high-dimensional data

UMAP proved to be a valuable tool for exploring both patient-level and drug-level patterns across our datasets. The projection of raw and merged demographic data clearly revealed existing cohort structures, driven by source-specific or subtype-related differences. And we can see, once molecular features were introduced, the embeddings reflected a continuous chemical space instead, reducing the visual impact of clinical subgrouping. Overall, UMAP provided clear visual cues about the nature of our data—whether categorical, clustered, or continuous—which helped us better understand the relationships among patients and treatments.

Racial prediction and imputation

The Gaussian Naive Bayes model effectively identified race-correlated patterns within the dataset, though it struggled with significant class imbalance that skewed predictions toward the majority group. Applying PCA transformation improved recall for minority classes, suggesting that dimensionality reduction can help surface subtle racial patterns embedded in

clinical and genomic data. However, overall performance remained biased, emphasizing how data imbalance limits the predictive power of such models. These findings highlight the potential for bias in clinical datasets, where the dominance of majority race groups can influence model outcomes and obscure the needs of underrepresented populations.

Limitations with racial prediction and imputation

The model has several notable limitations that impact its reliability and generalizability. One key limitation is its assumption that all features are independent, which oversimplifies the complex, interrelated nature of clinical data. In reality, medical variables often interact in non-linear and dependent ways, making this assumption unrealistic and potentially misleading. Additionally, the dataset suffers from significant class imbalance, which skews predictions toward the majority class and reduces the model's confidence and effectiveness in identifying minority group patterns.

Future improvements with racial prediction and imputation

Future improvements to the modeling approach should focus on addressing its current limitations and enhancing both performance and ethical robustness. Exploring non-linear models such as Random Forests or Neural Networks could better capture the complex interactions between clinical features that linear models like Naive Bayes may miss. These models are more capable of representing the intricate, non-independent relationships inherent in medical data. Additionally, implementing feature importance analysis would help identify which clinical or genomic factors are most influential in racial classification, improving the model’s transparency and interpretability.

Patient-drug response and IC50 predictions

As translational medicine approaches aim to speed the development of drug discovery to clinical practice by determining effective drug candidates targeting different diseases, the amount of time it takes to bring a drug to market still takes about 10-15 years with high failure and attrition rates using existing lab practices alone. With new approaches such as the implementation of AI models in translational research, the speed at which the drug reaches market, and ultimately the patient, is greatly increased with higher precision towards drug alternatives best suited for individualized treatments.

It is important to acknowledge the limitations due to future use and improvements of the model used to predict patient-drug response. It is worth noting that the model is trained on a small subset of cell lines and chemotherapies, which are not representative of the diverse human population and treatment plans that exist. It is also important to note that the target value, the IC50, is a measure of drug potency and estimate of drug-response against particular cell lines. It does not fully capture all factors related to the side effects of a drug used directly on more complex organisms such as human patients rather than immortalized cell lines. Therefore, further studies acquiring data from human clinical trials, family history, and genomic information if accessible would help to improve the inferences that could be made from our model.

With these limitations in mind, it is still encouraging to see the results coming from the neural network used to predict breast cancer patient response to a particular set of anti-cancer drug candidates. It is reasonable to see that the most complicated model trained with cell line information, nearest neighbor mapping of patients to cell lines, and strategically imputed racial information captures the most variability in our dataset and results in the smallest mean square errors between predicted vs actual IC50 values. IC50 values are better predicted with what the drug is directly in contact with. Overall, the models outlined in this report collectively and successfully reduce high-dimensional information to quantify the predictive power of translational research used in precision oncology. As much as translational research is not 100 percent indicative of patient outcomes, this report shows that companion tools such as in-vitro testing paired with AI research has good predictive power in patient-drug response to help narrow down effective drug candidates and speed delivery of cancer therapeutics onto market, and eventually to those whose lives can be saved in a timely manner.

References

Evitan, G. (2020, December 23). *Breast cancer (METABRIC)*. Kaggle. https://www.kaggle.com/datasets/gunesevitan/breast-cancer-metabric/data

Saha, A., Harowicz, M. R., Grimm, L. J., Weng, J., Cain, E. H., Kim, C. E., Ghate, S. V., Walsh, R., & Mazurowski, M. A. (2021). Dynamic contrast-enhanced magnetic resonance images of breast cancer patients with tumor locations [Data set]. The Cancer Imaging Archive. https://doi.org/10.7937/TCIA.e3sv-re93

Sorger, P.K. Mills, C., & Hafner, M. (2018, March 30). *Breast Cancer Profiling Project, Drug Sensitivity phase I: Fixed-cell GR measures of 35 breast cell lines to 34 small molecule perturbagens from library plate I. Dataset 1 of 2: Normalized growth rate inhibition values*. HMS Lincs Project. https://lincs.hms.harvard.edu/db/datasets/20343/results

Teng, J. (2019, January 18). *Seer breast cancer data*. IEEE DataPort. https://ieee-dataport.org/open-access/seer-breast-cancer-data

Srinivasan, B., Lloyd, M.D. (2024). Dose–Response Curves and the Determination of IC50 and EC50 Values. Journal of Medicinal Chemistry 2024 67 (20), 17931-17934. DOI: 10.1021/acs.jmedchem.4c02052

Motulsky H. (2024). GraphPad PRISM (version 10.4). Computer Software. https://www.graphpad.com/guides/prism/latest/curve-fitting/reg_absolute_ic50.htm

Green, M. (2024). *LogP, LogD, pKa and LogS: A Physicists guide to basic chemical properties.* GitLab. https://doktormike.gitlab.io/posts/navigating-logp-logd-pka-and-logs-a-physicists-guide/

Romanick, M., Holt, A. (2023). An ABC of PK/PD Core Concepts in Pharmacokinetics and Pharmacodynamics for Students of Medicine, Dentistry, and Pharmacy https://pressbooks.openeducationalberta.ca/abcofpkpd/chapter/tpsa/#:~:text=Topological%20polar%20surface%20area%20(TPSA,indication%20of%20its%20lipid%20solubility

Daina A., Michielin O., Zoete V. (2017). SwissADME: a free web tool to evaluate pharmacokinetics, drug-likeness and medicinal chemistry friendliness of small molecules. Nature Scientific Reports. DOI: 10.1038/srep42717

OMx Personal Health Analytics Inc. (2025). *Lipinski's Rule of Five*. DrugBank. https://dev.drugbank.com/guides/terms/lipinski-s-rule-of-five

Bajusz, D., Racz, A., Heberger, K. (2015). *Why is Tanimoto index an appropriate choice for fingerprint-based similarity calculations?* https://jcheminf.biomedcentral.com/articles/10.1186/s13321-015-0069-3

Tabosa, M. A. M., Hoppel, M., Bunge, A. L., Guy, R. H., Delgado-Charro, M. B. (2020). Predicting topical drug clearance from the skin. Springer. doi: 10.1007/s13346-020-00864-8

Landrum, G. (2013). *RDKit: Open-source cheminformatics*. https://www.rdkit.org

Cournapeau, D. (2007). *Scikit-learn: Machine learning in Python*. Journal of Machine Learning Research, 12, 2825–2830. https://www.jmlr.org/papers/v12/pedregosa11a.html

Paszke, A., Gross, S., Chintala, S., Chanan, G. (2016). *PyTorch: An imperative style, high-performance deep learning library. Advances in Neural Information Processing Systems*, 32, 8026–8037. https://papers.nips.cc/paper_files/paper/2019/hash/bdbca288fee7f92f2bfa9f7012727740-Abstract.html

Kingma D.P., Lei Ba J. (2015). Adam: A method for stochastic optimization. arXiv. https://arxiv.org/pdf/1412.6980